



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**ANÁLISIS COMPARATIVO DE CLASIFICADORES
PARA LA DETECCIÓN DE SUBTIPOS DE CÁNCER**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
DE SISTEMAS**

Autor(a) (es):

Bach. Díaz Bernilla Nataly Marlene

ORCID:

<https://orcid.org/0000-0001-5124-8303>

Asesor(a):

Mg. Tuesta Monteza Víctor Alexci

ORCID:

<https://orcid.org/0000-0002-5913-990X>

Línea de Investigación:

Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú 2021

APROBACIÓN DEL JURADO

**ANÁLISIS COMPARATIVO DE CLASIFICADORES PARA LA DETECCIÓN DE
SUBTIPOS DE CÁNCER**

Bach. Díaz Bernilla Nataly Marlene

Autor

Mg. Tuesta Monteza Víctor Alexci

Asesor

Mg. Herber Ivan Mejía Cabrera

Presidente de Jurado

Mg. Maria Noelia Sialer Rivera

Secretario de Jurado

Mg. Victor Alexci Tuesta Monteza

Vocal de Jurado

Dedicatorias

La presente investigación se la dedico a **Dios**, mi creador, único merecedor de toda GLORIA, a mi madre **Adby J. Bernilla**, a mi asesor **Mg. Victor A. Tuesta** por la contribución para la investigación.

Agradecimientos

A **Dios**, por la vida y por ser el soporte para enfrentar con sabiduría el proceso del desarrollo de la investigación. A mi madre **Maria M. Bernilla** y a mi padre **Luis F. Yuptón** por el apoyo moral, la paciencia y la comprensión en las madrugadas. Y por último a mi asesor por la guía y paciencia brindada.

Resumen

En la actualidad el cáncer es una de las primeras causas de muerte a nivel mundial, en la ingeniería el aprendizaje automático se está utilizando para analizar datos y aprender de ellos, consecuentemente son capaces de predecir o sugerir, y está teniendo un alto impacto en los avances tecnológicos médicos. El objetivo de este trabajo es realizar un análisis comparativo para la detección de los subtipos de un cáncer, la investigación inicia con la selección del tipo de cáncer, el cual se seleccionó el cáncer de mama, posteriormente se caracterizó los subtipos del cáncer obteniendo 4 subtipos los cuales son, Luminal A, Luminal B, Basal o triple negativo y el tipo de cáncer enriquecido con Her2. Posteriormente se realizó la clasificación siendo los clasificadores Support Vector Machines, K-Nearest Neighbor y Naive Bayes los seleccionados, además se utilizaron los datos obtenidos del bioproyecto GSE10886 que contiene 200 muestras de tejido tumorosos generados en GEO2R (Herramienta que analiza datos genómicos). Los resultados obtenidos de los indicadores precisión, error, sensibilidad y especificidad de los clasificadores son SVM (97%, 3%, 95%, 99%) , siendo el que obtuvo mejor performance en comparación al clasificador KNN(88%, 12%, 89%, 96%) y del clasificador NB (90%, 10%, 89% y 98%) respectivamente, demás se obtuvo el tiempo de respuesta de la ejecución de los clasificadores siendo del clasificador SVM 0.36 segundos, 2.79 segundos del clasificador KNN y 0.33 segundos del clasificador Naive Bayes. Finalmente se concluyó que el clasificador que obtuvo mejor performance en los resultados evaluados es el clasificador SVM con un 97% de precisión, 3% de error , 95% de sensibilidad y un 99% de especificidad y por último el clasificador con menos tiempo de respuesta fue el clasificador Naive Bayes con 0.33 segundos.

Palabras Clave: Clasificación automática, Subtipos de Cáncer, Support Vector Machines, K-Nearest Neighbor, Naive Bayes

Abstract

Currently cancer is one of the leading causes of death worldwide, in engineering machine learning is being used to analyze data and learn from them, consequently they are able to predict or suggest, and is having a high impact on medical technological advances. The objective of this work is to perform a comparative analysis for the detection of the subtypes of a cancer, the research begins with the selection of the type of cancer, which was selected breast cancer, then the subtypes of cancer were characterized obtaining 4 subtypes which are, Luminal A, Luminal B, Basal or triple negative and the type of cancer enriched with Her2. Subsequently, the Support Vector Machines, K-Nearest Neighbor and Naive Bayes classifiers were selected for classification, and the data obtained from bioproject GSE10886 containing 200 tumor tissue samples generated in GEO2R (tool that analyzes genomic data) were also used. The results obtained for the indicators accuracy, error, sensitivity and specificity of the classifiers are SVM (97%, 3%, 95%, 99%), which obtained the best performance compared to the KNN classifier (88%, 12%, 89%, 96%) and the NB classifier (90%, 10%, 89% and 98%) respectively. The response time for the execution of the classifiers was also obtained, being 0.36 seconds for the SVM classifier, 2.79 seconds for the KNN classifier and 0.33 seconds for the Naive Bayes classifier. Finally, it was concluded that the classifier that obtained the best performance in the evaluated results is the SVM classifier with 97% accuracy, 3% error, 95% sensitivity and 99% specificity and finally the classifier with the shortest response time was the Naive Bayes classifier with 0.33 seconds.

Keywords: Automatic classification, Cancer Subtypes, Support Vector Machines, K-Nearest Neighbor, Naive Bayes.

Índice

I. INTRODUCCIÓN	11
1.1. Realidad Problemática	11
1.2. Trabajos previos	12
1.3. Teorías relacionadas al tema	18
1.4. Formulación del Problema	39
1.5. Justificación e importancia del estudio	39
1.6. Hipótesis	39
1.7. Objetivos	39
1.7.1. Objetivo general	39
1.7.2. Objetivos específicos	40
II. MATERIAL Y MÉTODO	40
2.1. Tipo y Diseño de Investigación	40
2.2. Población y muestra	40
2.2.1. Población	41
2.3. Variables, Operacionalización	41
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad	42
2.5. Procedimiento de análisis de datos	42
2.6. Criterios éticos	42
2.7. Criterios de Rigor Científico	42
III. RESULTADOS	44
3.1. Resultados en Tablas y Figuras	44
3.1.1. Resultados de Clasificador SVM	44
3.1.2. Resultados del Clasificador KNN	45
3.1.3. Resultados del Clasificador Naive Bayes	47
3.1.4. Resumen de los Resultados	48

3.2. Discusión de resultados.....	49
3.3. Aporte práctico.....	51
IV. CONCLUSIONES Y RECOMENDACIONES	74
4.1. Conclusiones.....	74
4.2. Recomendaciones.....	75
REFERENCIAS.	76
ANEXOS.....	80

Anexos

Anexo 1 Resolución de aprobación del proyecto de investigación.....	80
Anexo 2 Población de Algoritmos.....	81
Anexo 3 Resumen de Resultados SVM.....	82
Anexo 4 Resumen de Resultados KNN.....	83
Anexo 5 Resumen de Resultados NB	84
Anexo 6 Interfaz SVM	90
Anexo 7 Manual de la implementación SVM	91
Anexo 8 Interfaz KNN.....	93
Anexo 9 Manual de la implementación KNN	94
Anexo 10 Interfaz NB.....	95
Anexo 11 Manual de la implementación Naive Bayes	96

Figuras

<i>Figura 1 Preparación de Muestras</i>	25
<i>Figura 2 Imagen Microarray, Fuente: (Prada, 2017)</i>	26
<i>Figura 3. Proceso del experimento de un microarray, Fuente: (Instituto Nacional del Genoma Humano, 2020)</i>	29
<i>Figura 4 Grafico de Resultados de Sensibilidad y Especificidad con la tecnica SVM, Fuente: Elaboración Propia</i>	45
<i>Figura 5 Gráfico de Resultados de Sensibilidad y Especificidad del Clasificador KNN, Fuente: Elaboración propia</i>	46
<i>Figura 6 Gráfico de Resultados de Sensibilidad y Especificidad del Clasificador NB, Fuente: Elaboración Propia</i>	48
<i>Figura 7. Grafico Del Resumen de los Resultados (Tiempo de respuesta), Fuente: Elaboración Propia</i>	49
<i>Figura 8. Gráfico del Resumen de los resultados de los indicadores (Precisión, error, Sensibilidad y Especificidad), Fuente: Elaboración Propia</i>	49
<i>Figura 9 Método Propuesto, Fuente: Elaboración Propia</i>	51
<i>Figura 10 Muertes por cáncer de mama, Fuente: (MINSA, s.f.)</i>	53
<i>Figura 11. Comparación gráfica de algoritmos</i>	60
<i>Figura 12. Secuencias de experimentos de 232 muestras</i>	62
<i>Figura 13 Diseño de dispersión de datos</i>	64
<i>Figura 14. Resumen</i>	66
<i>Figura 15. Resultado de la selección de Vectores de Soporte</i>	67
<i>Figura 16 Ejemplo KNN</i>	68
<i>Figura 17 Ejecución de str() y head()</i>	69
<i>Figura 18. Eliminación de columna 1</i>	69
<i>Figura 19. Datos en consola normalizados</i>	70
<i>Figura 20. Resultado del Modelo KNN</i>	71
<i>Figura 21. Resultado de modelo NB</i>	73
Figura 22 <i>Resumen de Resultados SVM</i>	82
Figura 23 <i>Resumen de Resultados KNN</i>	83
Figura 24 <i>Resumen de los Resultados NB</i>	84
Figura 25 <i>Interfaz SVM - RSTUDIO</i>	90
Figura 26 <i>Interfaz KNN- RSTUDIO</i>	93
Figura 27 <i>Interfaz NB – RSTUDIO</i>	95

I. INTRODUCCIÓN

1.1. Realidad Problemática.

En la actualidad, existen compañías a nivel mundial que toman mucha importancia en el tema del aprendizaje automático y los avances relacionados, como el aprendizaje profundo, los cuales han permitido que las máquinas adquieran conocimientos tácticos. (Buhigas, 2018)

El aprendizaje automático en el uso básico de sus algoritmos para analizar datos, aprender de ellos para luego ser capaces de predecir o sugerir, el aprendizaje está teniendo un impacto en los avances tecnológicos, en la industria del software como en la vida diaria (Rodríguez, 2018), además encontramos soluciones a los problemas de salud, estos avances permiten desarrollar nuevas herramientas importantes para el diagnóstico como para el tratamiento de diferentes enfermedades como el cáncer. (HIRALES, 2015)

El cáncer es una de las primeras causantes de muerte a nivel mundial; en 2012 se le sumaron 8,2 millones de muertes (Organización Mundial de la Salud, 2019). El cáncer de mama es el primer tipo de cáncer entre las mujeres y existe el riesgo de que una de cada 8 mujeres se queden atrapadas en ésta (World Cancer Research Fund International, 2012).

El diagnóstico precoz de esta enfermedad mortal es muy importante, el examen de detección del cáncer es la búsqueda de cáncer antes de que una persona tenga algún síntoma, el atlas del genoma humano nos describe que existen diferentes clases de exámenes de detección, como las pruebas de laboratorio, estudios de imágenes y Biopsias que ayudan al diagnóstico, pero los médicos no pueden confiar solo en ellos para diagnosticar el cáncer (Instituto Nacional del Cáncer, 06), porque éstos presentan riesgos como resultados positivos falsos o resultados negativos falsos. (National Cancer Institute, 2015).

Encontramos ciencias que aportan a la investigación tecnológica aplicada a la biología; como por ejemplo la bioinformática que es la combinación de las

ciencias computacionales y las biológicas; ciencia se involucra en el uso de computadoras, bases de datos, matemática y estadísticas para almacenar, organizar y analizar grandes volúmenes de información biológica. (Instituto Nacional del Cáncer, 2020).

La bioinformática aporta a la recolección de datos biológicos con la tecnología de expresión génica, los microarrays de datos han comenzado a utilizarse con frecuencia en diferentes ambientes, pero su desventaja al uso en tecnologías es que es muy costoso. (MAULI, 2014)

Hoy en día existen tecnologías que nos facilitarían a la clasificación de datos biológicos como es la clasificación de procesos agrupados, y éstos aportarán a un diagnóstico eficaz y preciso.

1.2. Trabajos previos.

Ramírez Pérez, Natalia Andrea; Gómez Vargas , Ernesto; Forero Cuellar, Oscar Mauricio (2019), realizaron una investigación de Clasificadores supervisados del cáncer de próstata a partir de imágenes de resonancia magnética en secuencia T2, donde resaltan la importancia de la inserción de modelos computacionales que afecten al diagnóstico de cáncer de próstata, proponiendo una solución para los investigadores en la oncología, se enfocan en el diagnóstico del cáncer de próstata y ayudando oportunamente a cancerólogos, hospitales, oncólogos expertos en cáncer y a todos los interesados en mejorar el diagnóstico del cáncer de próstata. Realizaron un estudio de resonancias magnéticas que posee con 110 pacientes y todas los estudios incluyeron imágenes ponderadas de T2, iniciando por un filtro de las imágenes para lograr realzar y detectar contornos, donde a partir de éstos se determinan las diferentes características, las cuales se encuentran en pixeles y se realizan considerando el sistema de datos e informe de imágenes prostáticas. Utilizando modelos de clasificación como; regresión logística, red neuronal, bosques aleatorios y árbol de decisión. Finalmente obtuvieron como resultados

al modelo de regresión con una precisión 0,831, siendo el modelo de clasificación con más alto grado de precisión.

Ghongade, R.D.y Wakde, D.G. (2017) Realizaron Computer aided Diagnosis System for breast cancer using, RF classifier, proponiendo el método de aprendizaje automático basado en un clasificador random forest. Utilizó una base de datos llamada MIAS que integra imágenes de mamografías digitales. El pre-procesamiento generalmente es necesario para mejorar la baja calidad de la imagen. El ROI se determina de acuerdo con el tamaño del área sospechosa. Una vez segmentada la región sospechosa, las características se extraen mediante análisis de textura. La técnica de selección de características se utiliza para la detección de características de alta dimensión. Un método estadístico, la matriz de coincidencia de nivel de gris (GLCM) se utiliza como un atributo de textura para extraer el área sospechosa. Utilizando imágenes de mamografía con 1024×1024 píxeles que se importaron de la base de datos MIAS. Para mejorar el contraste y suavizar la imagen, se realiza un pre procesamiento que será útil en etapas posteriores. Luego, se segmenta la región del seno para encontrar el área sospechosa de los segmentos del seno. Posteriormente se realiza la extracción de las características de textura y el cálculo de estadísticas de textura. Algunas características relevantes se seleccionan mediante el método de Selección de características basadas en correlación rápida (FCBF) y estas características se utilizan para la clasificación para determinar las masas o no masas. Los resultados obtenidos en la investigación logró una precisión de 97.32%, sensibilidad de hasta 97.45%, especificidad de aproximadamente 98.13% y ROC con AUC es 97.28%.

Pise (2016), realizaron Lung Cancer Detection Using Bayesein Classifier and FCM Segmentation, en la cual propusieron una detección de cáncer de pulmón en la etapa primaria utilizando el clasificadores bayesiano confirmando la salud de un paciente. Utilizaron un pre procesamiento de imagen para mejorar el contraste, para posteriormente realizar la extracción de características como la textura que reducen la dimensionalidad y transformando datos de entrada en un conjunto de características, por último se realiza la segmentación en la que

se utiliza una técnica aplicada, ésta técnica permite que una pieza de datos pertenezca a un grupo, los valores de membresía asignados están entre 0 y 1. Al implementar este trabajo de investigación, el algoritmo bayesiano se usa para clasificar la entrada de imágenes pulmonares CT para decidir si es normal o anormal. El método que han implementado resulta útil y proporciona resultados más precisos para los métodos mejores que los mencionados en la literatura. Lograr una alta precisión es posible gracias al uso de 12 características diferentes de contraste, correlación y varianza. Se utilizan diferentes momentos inversos y prominencia de agrupación y sombra de agrupación con 6 características diferentes y más eficientes para mejorar la precisión y para la extracción de las 12 características.

Alam, Janee; Alam, Sabrina y Hossan, Alamgir en (2018), realizaron Multi-Stage Lung Cancer Detection and Prediction Using Multi-class SVM Classifier, proponiendo en su método una mejora de la imagen utilizando un filtro de mediana selectiva reparando una identificación más confiable y más competente, luego se realizó la segmentación y detección de imágenes de nódulos pulmonares en la imagen TC que la utilizaron para mejorar la resolución, después realizaron una extracción de características utilizando la técnica GLCM(Método de conciencia de nivel de Gris) posteriormente realizaron la clasificación del nódulo canceroso utilizando el clasificador SVM, concluyendo con el método de predicción con una técnica de binarización volviendo diferente sobre la imagen en escala de grises.

Yu Lin, Chun; Li, Ruiming; Akutsu, Tatsuya; Ruan, Peiying y See, Simon en (2018), realizaron Deep Learning with Evolutionary and Genomic Profiles for Identifying Cancer Subtypes, proponiendo una estrategia basada característica (FES) y una estrategia usa el clasificador PAM50, utilizó un método contiene dos estrategias, la primera en FES identifican genes y proteínas conservados evolutivamente llamados genes y proteínas centrales, mediante el uso de 310060 familias de interacción, posteriormente se realizó la selección de genes y proteínas para recoger modificando el puntaje de evolución para que se pueda realizar la identificación de proteínas centrales y PPI del proteína humano, luego

se realizó una coevaluación y distanciamiento de expresiones, evaluando las correlaciones para cada par de gen central, calculando la distancia como entrada del análisis MDS utilizando la herramienta KEGG Orthology, finalmente se realizó la arquitectura de red neuronal convolucional para perfiles de expresión y diagramas MDS, para que se pueda utilizar la información combinada de valores de expresión de RNA-seq y las alteraciones del número de copias, concluyendo un conjunto de datos de n muestras de tamaño $2 \times q$, realizando una red neuronal convolucional (CNN), cuyo tamaño de filtro de la primera capa es $2 \times w$, donde q es el número de genes y w es el ancho del filtro. En los resultados obtenidos se pudo examinar la capacidad de identificación del subtipo de cáncer, realizando estrategias basadas en características en los datos genómicos de BRCA usando el conjunto de genes centrales (6045 genes), el conjunto completo del genoma (16321 genes) y el conjunto aleatorio (6045 genes). Después de realizar la construcción aleatorio de genes centrales el procedimiento se repitió diez veces obteniendo como resultados la precisión más alta de 77% de los genes FES que se cruzaron 5 veces, en comparación al uso de conjunto de genes PAM50 que alcanzaron una precisión del 65%. Además se descubrió que utilizando IMS con aumento de datos llega a una precisión de 87% siendo la más alta. Y tuvo como conclusiones relevantes un conjunto de nuestro FES e IMS mediante el uso del conjunto de genes centrales, que se selecciona por conservación evolutiva, proporcionando una nueva ruta de acceso posible para la identificación del tipo / subtipo de cáncer.

R. Catchpoole; Eoberts, Aedan y Kennedy, Paul (2018), realizaron Variance-based Feature Selection for Classification of Cancer Subtypes Using Gene Expression Data, Investigación realizada en Australia, que se enfoca en la clasificación de subtipos de cáncer utilizando datos de expresión genética, propusieron una solución con la selección de características por varianza diferencial al problema general de clasificación de subtipos de cáncer, proponiendo además el método que inicia con la extracción de datos, posteriormente realizaron la selección de características y transformación utilizando métodos como el Dinalankara y Corrada Bravo, calculado la relación de varianza entre muestras de cáncer y la varianza entre muestras normales,

pasando a la clasificación realizando un análisis con Rv 3.4.0 y utilizando el paquete randomForest. Obteniendo como resultados propusieron un método denominado DCB para predecir el tumor progresión o pronóstico basado en sumas atípicas, además se realizó la clasificación con términos como alto o bajo riesgo según la función de la recaída y la información de supervivencia.

Turgut, Dagtekin, y Ensari (2018), realizaron *Microarray Breast Cancer Data Classification Using Machine Learning Methods*, abordando la clasificación de datos utilizando métodos de aprendizaje automático enfrentando el cáncer de mama, propusieron una solución utilizando los algoritmos de SVM, KNN, MLP, árboles de decisión, bosque aleatorio, regresión logística, Adaboost y máquinas de refuerzo de gradiente, para la selección de características, el método que propusieron inició con la selección de características, realizando el procesamiento de datos en ciencia de datos, y por último se utiliza los algoritmos de clasificación como el SVM (Support Vector Machines), KNN(K-Nearest Neighbours), MLP (Multi Layer Perceptron), DT (Decisión Trees), RF (Random Forests), LR (Registicistic Logression), Ada(Adaboost) y GFM (Gradient Boosting Machines). Después de la validación cruzada en K-Fold que la utilizaron para garantizar la exactitud del resultado de clasificación y la matriz de confusión se obtienen resultados del primer conjunto de datos con 133 muestras con 1919 características obteniendo al SVM como el más alto después de los métodos de selección de características, los arboles de decisión se clasificaron con la precisión más baja, el método MLP dio resultados cercanos a otros métodos.

Polat y Sentuk en (2018) , realizaron *A Novel ML Approach to Prediction of Breast Cancer: Combining of mad normalization, KMC based feature weighting and AdaBoostM1 classifier*, en Turquía se realizó esta investigación abordando el tema con enfoque de ML de la combinación de normalización loca, ponderación de características basadas en KMC y clasificador AdaBoostM1, enfrentando la predicción del cáncer de mama, proponiendo una estructura híbrida de tres pasos para detectar la presencia de cáncer de seno, utilizando un método que tiene como primer paso la normalización MAD (desviación

absoluta media), en la que midieron un conjunto de datos, después se realizó la ponderación de características propuesta por k. Polat en la que utiliza los conjuntos de datos de cáncer de mama por último se realizó la clasificación mediante AdaBoost que resolvió problemas de clasificación binaria y clasificando las muestras, en sus resultados demostró mediante siete medidas de rendimiento que el método propuesto es el mejor modelo que solo un modelo clasificador.

Setiawan, y otros en (2018) , realizaron Classification of Cell Types In Acute Myeloid Leukemia (AML) of M4, M5 and M7 Subtypes With Support Vector Machine Classifier, Esta investigación aborda el uso del clasificador de SVM, enfrentando los tipos de células en la leucemia mieloide aguda (ALM) de los subtipos M4, M5, M7. Ellos propusieron una solución en la que intenta ayudar a superar el problema haciendo una clasificación automática del tipo de célula a partir de imágenes de células. Según su método propuesto el primer paso fue la adquisición de imágenes AML, M4, M5 y M7 que se obtuvieron de RSUP Sardjito Yogyakarta a través del procesamiento de aprobación ética, posteriormente se procedió a la mejora de la imagen y a la segmentación de ella a través de la combinación de colores con canales RGB, luego se realizó el proceso de extracción de características obteniendo seis datos numéricos que representan las características de las celdas que incluye área, perímetro, circularidad, relación de núcleos, media y desviación estándar que serán utilizadas como entradas, luego se procedió a la normalización y combinación de características y se finalizó con el proceso de selección de parámetros y las pruebas necesarias y obtuvieron como resultados de segmentación de un total de 105 imágenes que consisten en 35 imágenes de cada subtipo, 1500 células segmentadas correctamente y 210 células están segmentadas incorrectamente.

J, Y, y Q (2012), realizaron Classification Network of Gastric Cancer Construction based on Genetic Algorithms and Bayesian Network, abordando el tema de algoritmos genéticos y red bayesiana enfrentando el cáncer gástrico, proponiendo una construcción de una red de clasificación, que se realizó mediante un método que inicia seleccionando un gen esencial, el que cambió

significativamente el patrón de expresión y luego se utilizaron estos gens como un conjunto de características para que reduzca el número de variables, luego se usaron algoritmos genéticos y modelo de red bayesiana para construir el clasificador, el procesos de construcción usa estos tres datos de expresión génica para que el clasificador aprenda. Los hallazgos obtenidos fueron que la precisión de la clasificación se calculó mediante la validación cruzada de licencia única (LOOCV) y alcanzó el 99.8%.

1.3. Teorías relacionadas al tema.

Para realizar la investigación se presentan las siguientes teorías que se relacionan al tema.

1.3.1. Microarrays en estudios de expresión génica

En la actualidad se está hablando mucho acerca de los experimentos de biología molecular, en la presente investigación trataremos, los microarrays están permitiendo a los investigadores caracterizar enfermedades genéticas así como el cáncer a nivel molecular para que sean conducidas a diagnósticos más efectivos, ahora les detallaré puntos importantes de los microarrays y métodos de caracterización existentes (McLachlan, Do, & Ambrise, 2019).

1.3.1.1. Genoma, genotipo y expresión génica

La representación de todo nuestro complemento genético es el genoma humano, el genoma humano fue completado en el 2003, realizándose una representación de identificación y predicción de las secuencias de pares de bases, en los 23 para de los cromosomas presentes en el núcleo de las células, ésta representación es un mapeo de complementos genéticos individuales, estos llamados genotipos, el genotipo de cada persona es único ya que existen innumerables de variaciones de secuencia genética en forma de mutaciones y polimorfos. (McLachlan, Do, & Ambrise, 2019)

El mapeo realizado del genoma humano proporciona bases para los investigadores de genética y además en ramas como la biología

molecular, la bioquímica y la biofísica, bioestadística, fármaco genética, bioinformática, informática y muchos otros.

1.3.1.2. Aspectos de la biología y fisicoquímica subyacentes

El ácido desoxirribonucleico también conocido como ADN está contenido dentro de los cromosomas en el núcleo de cada célula. La molécula de ADN consta de dos cadenas antiparalelas de enlaces de azúcar y fosfato que están unidos en una doble hélice derecha por el enlace de hidrógeno no covalente entre pares de bases amino unidas, que se encuentran en un plano aproximadamente perpendicular al eje largo de la molécula. La disposición antiparalela de las cadenas de nucleótidos requiere la transcripción de una nueva cadena de ARN o ADN para ejecutarse en la dirección opuesta de of la plantilla. Las interacciones hidrófobas entre las bases apiladas en el interior de la estructura de la molécula de ADN también estabilizan la doble hélice al apretarla fuertemente para excluir el agua y otras moléculas no polares. Adenina, timina, guanina. y la citosina son las bases de amina, cuyo orden secuencial contribuye al funcionamiento de un segmento particular de la cadena de ADN (un gen). Las bases exhiben un enlace característico y específico conocido como base purina.

El emparejamiento de bases (también conocido como ensamblaje de bases Watson-Crick) es un proceso de enlace químico que permite que se produzca la hibridación molecular. (McLachlan, Do, & Ambrise, 2019) Entre dos cadenas de ADN, la base conocida como adenina (A) se une con timina ('T) a través de dos enlaces de hidrógeno, y la guanina (G) se une específicamente a la citosina (C) a través de dos enlaces de hidrógeno, de una manera que crea el doble hélice. Entre una cadena de ADN y una cadena de ácido ribonucleico o ARN (durante la transcripción), la adenina de la cadena de ADN se unirá específicamente al uracilo base (U) de la cadena de ARN, y la guanina se unirá nuevamente específicamente a la citosina. La base de amina que formará un par de unión con otra base de amina (A con T o A con U, y G con C)

se considera su base complementaria, y una sola cadena de ADN o ARN que contiene el mismo orden secuencial de Las bases complementarias para la unión como una cadena dada se consideran su cadena complementaria. Las cadenas simples de ADN o ARN formarán enlaces estables solo con una cadena complementaria. Esta especificidad permite que el "mensaje" de la secuencia de emparejamiento de bases en ese segmento de ADN se comuniquen a través del proceso de transcripción. (McLachlan, Do, & Ambrise, 2019)

1.3.2. CDNA

El ARN mensajero también conocido como ARNm es la forma de ácido ribonucleico que dirige la producción de proteínas celulares, por lo que es importante en los experimentos de expresión génica. Los investigadores quieren observar qué proteínas celulares se producen y la función de esas proteínas en tipos particulares de células (como las células tumorales) o en respuesta a estímulos externos específicos, por lo que están interesados en probar los patrones de expresión del ARNm. Aunque la síntesis y activación de proteínas no están reguladas únicamente a niveles de ARNm en una célula, la medición de ARNm se usa para estimar los cambios celulares en respuesta a señales externas y cambios ambientales. (McLachlan, Do, & Ambrise, 2019)

El ARNm en una muestra biológica de ensayo que se une primero químicamente a una molécula de ADN para posteriormente eliminarlo de los otros componentes celulares. Sin embargo, la molécula de ARNm es muy frágil y puede descomponerse fácilmente por la acción que realizan las enzimas que prevalecen en soluciones biológicas, por lo que los investigadores comúnmente manipulan una forma de ADN que posee las bases complementarias del ARNm mientras existe en un estado más estable. Esta forma de ADN, conocida como ADN complementario (ADNc), es la que se crea directamente a partir de la muestra de ARNm a través de un procedimiento conocido como transcripción inversa (transcripción de secuencias de bases genéticas complementarias de ARN a ADN). El ADNc

también se llama ADN sintético, ya que se forma a través de la transcripción inversa del ARN en lugar de a través de la autorreplicación durante la división celular. El ADNc generalmente se prepara en longitudes de cadena de 500 a 5.000 bases en secuencia conocida.

1.3.2.1. Etiqueta de secuencia expresada

Los genes humanos están formados por secuencias de emparejamiento que continuamente se replican, durante la traducción de ARNm para formar cadenas de polipéptidos específicos en la síntesis de proteínas. Las secuencias que se traducen en síntesis de proteínas son secuencias codificantes, conocidas como e, contras, por otro lado las secuencias no se conocen como intrones. Las enzimas activadas durante la transcripción de ARNm reconocen las uniones no codificantes en la secuencia de nucleótidos y empalman los exones para la producción de proteínas después de eliminar los intrones. Etiqueta de secuencia expresada (EST) es el nombre dado a un segmento secuencial corto de un gen, el cual se genera para representar la porción de codificación de un gen; así. Un EST se utiliza con frecuencia como un sustituto genético para la amplificación por PCR, la producción de microarrays y los experimentos. La sustitución de secuencias de nucleótidos más cortas por ADN genómico se propuso en la década de 1980 y se realizó por primera vez en experimentos con clones de ADNc derivados de tejido cerebral humano por un grupo de investigación del Instituto Nacional de Trastornos Neurológicos y Accidentes Cerebrovasculares, Institutos Nacionales de Salud de los Estados Unidos (McLachlan, Do, & Ambrise, 2019)

1.3.3. Tecnología y aplicación de Microarray

La tecnología de microarrays de ADN de alta densidad permite a los investigadores monitorear las interacciones entre miles de transcripciones de genes en un organismo en un solo medio experimental, que a menudo es un portaobjetos de vidrio o una membrana de nylon. Antes de la informatización y la miniaturización de esta tecnología, los investigadores se

limitaron a los exámenes de un número mucho menor de unidades genéticas por experimento y pudieron evaluar las interacciones entre genes en condiciones cambiantes a una escala mucho menor.

La tecnología de microarrays es principalmente útil en la evaluación de series de expresión génica en trastornos complejos debido a su capacidad para observar la expresión de los mismos genes en diferentes muestras al mismo tiempo y en respuesta a los mismos estímulos. En la investigación biomédica el uso de los microarray se asemeja a algunos avances tecnológicos encontrados en la industria de la informática, como el de la distribución paralela. Distribuir el "trabajo" de un experimento de manera paralela facilita la resolución de problemas computacionalmente complejos y se convierte en algo más que el equivalente a ejecutar miles de pasos experimentales al mismo tiempo. Los microarrays están generalmente diseñados para proporcionar una distribución paralela del trabajo de un experimento, cada microarray puede representar miles de ensayos bioquímicos separados realizados en un período de tiempo mucho más corto (MS, TM, & CM , 1993)

1.3.3.1. Herramientas de tecnología de microarrays

La siguiente es una simplificación de los complicados procesos bioquímicos y protocolos detallados involucrados en la preparación de materiales de ácido nucleico y microarrays y en la realización de estudios de expresión génica en el laboratorio de biología, principalmente se introduce a la tecnología, y se alienta a los lectores a consultar publicaciones de referencia y especialistas en el campo para mejorar su comprensión de la tecnología y de los procesos experimentales que crean los datos que posteriormente analizarán (McLachlan, Do, & Ambrise, 2019)

a) Array

La matriz es una base sólida sobre la cual se organiza sistemáticamente una cuadrícula de "puntos" o gotitas de material genético de secuencia

conocida. La matriz es una pequeña pieza de vidrio o nylon (que se asemeja a un portaobjetos de microscopio), con miles de puntos o pozos que pueden contener una gota que representa una secuencia de ADNc diferente. Los tamaños de los chips de silicio cuadrados de 0.5cm x 0.5cm. Cada punto en la cuadrícula de la matriz puede representar un ensayo experimental independiente para la presencia y abundancia de una secuencia específica de bases en la cadena de polinucleótidos de la muestra (McLachlan, Do, & Ambrise, 2019)

b) Observador

La máquina robótica que aplica las gotitas de diferentes cadenas de ADNc de secuencia conocida a un pozo o punto en la matriz se llama spotter o matriz. El producto se aplica a cada punto en una cuadrícula para alinear una gran cantidad de pruebas dentro de cada experimento. El observador utiliza métodos de contacto o sin contacto para aplicar el material de la sonda a la matriz. El manchado por contacto se realiza con un instrumento similar a una pluma estilográfica bajo presión constante. Métodos relativamente nuevos de manchado sin contacto aplican la tecnología de chorro de tinta o el efecto capilar piezoeléctrico para completar la cuadrícula de gotitas de la sonda. La tecnología de detección original, que utilizaba un alfiler o una aguja como observador, todavía se usa en muchos laboratorios. Los métodos de detección sin contacto generalmente aumentan la velocidad de producción de microarrays. (McLachlan, Do, & Ambrise, 2019)

c) Etiquetado de una muestra para detección

Para el etiquetado, identificamos y procedemos a medir la presencia de un polinucleótido de secuencia desconocida (en la muestra objetivo) después de que se une al material en el microarray, se marca con un tinte fluorescente, se puede utilizar fluoresceína, rodamina o cumarina. El colorante se incorpora inmediatamente con la molécula durante la transcripción inversa. Se utiliza un color de tinte diferente para cada

muestra y, en general, solo se usan dos colores de tinte, debido a los requisitos de escaneo e imagen para detectar la luz fluorescente de longitudes de onda específicas. Los protocolos experimentales comúnmente requieren la aplicación de un tinte fluorescente a los polinucleótidos de la muestra objetivo experimental o desconocida, y un tinte fluorescente diferente a los que son los polinucleótidos de una muestra objetivo conocida o control e ilustra la preparación de muestras objetivo. (McLachlan, Do, & Ambrise, 2019)

d) Hibridación

La hibridación molecular es la asociación de cadenas simples de polinucleótidos a través de sus propiedades específicas de apareamiento de bases para formar una molécula complementaria de cadena doble. Este proceso químico ocurre entre las cadenas de polinucleótidos marcadas de los tejidos diana (incluidas las de secuencias desconocidas) y sus cadenas complementarias de ADNc de secuencia conocida entre los puntos en la matriz. Idealmente, si un polinucleótido de la muestra objetivo contiene una secuencia de bases que es complementaria a la de un polinucleótido en un punto de la matriz, se hibridará con la molécula en ese punto. La ubicación de ese punto en la cuadrícula de la matriz será entonces detectable por la luz fluorescente que se emite durante los procesos de escaneo e imagen. Cuando muchos polinucleótidos diana se hibridan con hebras de sonda de ADNc complementarias en un punto de la matriz, entonces las señales fluorescentes emitidas y detectadas en ese punto tendrán mayor intensidad. (McLachlan, Do, & Ambrise, 2019)

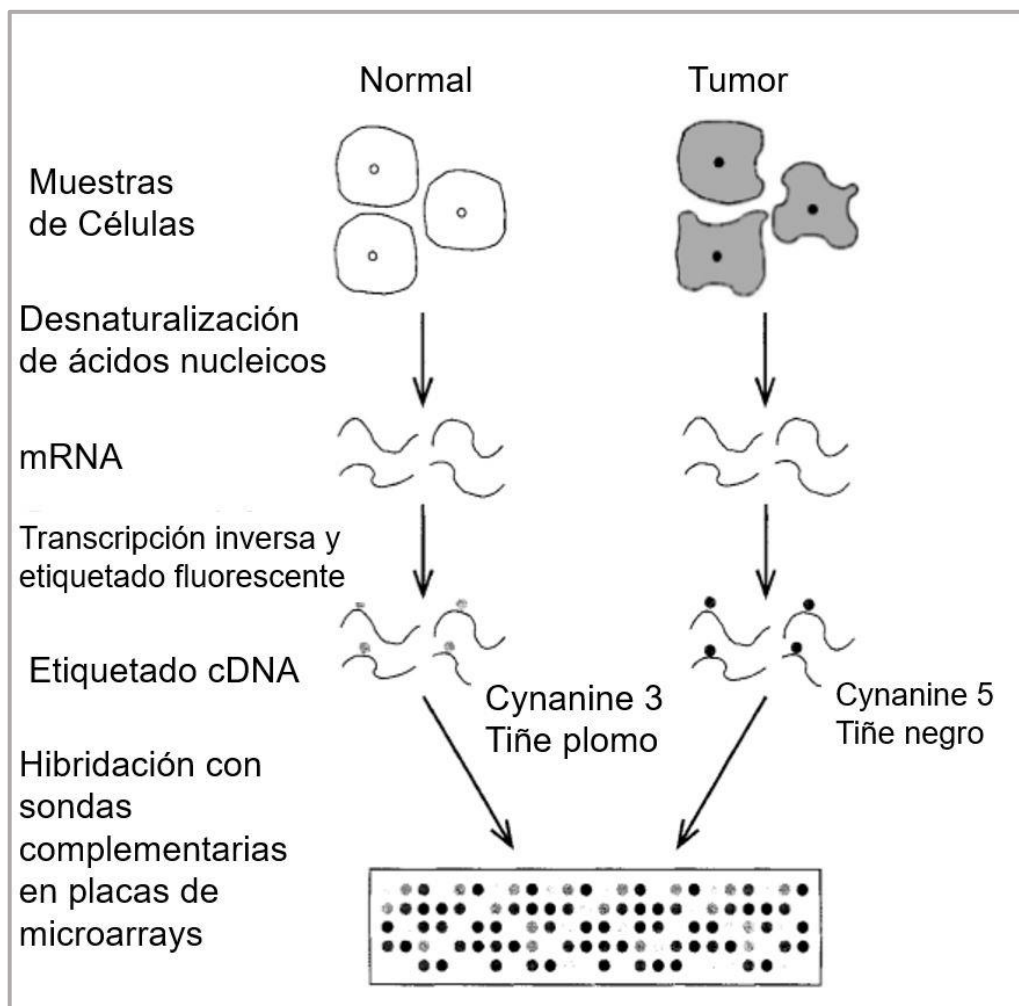


Figura 1 Preparación de Muestras

Fuente: (McLachlan, Do, & Ambrise, 2019)

Preparación de muestras grandes; el proceso desde las muestras celulares hasta el microarray.

e) Escanear el array

Existen muchos escaneres y la gran mayoría utiliza una frecuencia específica de luz (por ejemplo, un láser de argón) en la región ultravioleta para excitar el tinte fluorescente unido a las muestras objetivo que se han hibridado con sus secuencias de sonda complementarias en la matriz. Los fotones emitidos por el tinte excitado se recogen en un detector, que mide y registra sus niveles, convirtiendo las mediciones en señales eléctricas. Se usa un microscopio confocal o dispositivo acoplado a carga que registra la intensidad de los fotones como un detector de exploración.

Dado que generalmente se usan dos marcadores fluorescentes diferentes en los estudios de expresión génica, cada portaobjetos se escanea a dos longitudes de onda, y el generador de imágenes debe ser capaz de detectar los polinucleótidos hibridados y medir sus cantidades al menos en las dos longitudes de onda diferentes de la luz, y debe poseer una alta -resoluciones de escaneo de resolución. La cianina 3 (Cy3) y la cianina 5 (Cy5) son colorantes fluorescentes que se usan comúnmente en esta tecnología porque sus espectros de emisión están bien separados, lo que proporciona una baja probabilidad de diafonía.

Los lectores de imágenes suelen utilizar una cuadrícula sobre la matriz para asociar la señal de cada punto con su ubicación en la matriz y, por lo tanto, con su identificación de secuencia de emparejamiento de bases. Con una buena filtración de la luz dispersada, el detector registrará solo la luz de los pares hibridados marcados con fluorescencia, y se detectará una mayor intensidad de fluorescencia en los puntos donde se han hibridado más polinucleótidos a la matriz.

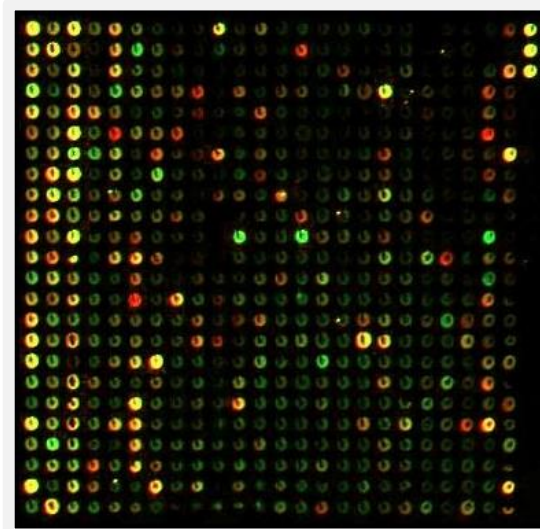


Figura 2 Imagen Microarray, Fuente: (Prada, 2017)

Imagen de microarray que muestra genes expresados diferencialmente.

f) Procesamiento de imagen final

Se obtiene una imagen digitalizada de la matriz escaneada del escáner de microarrays y se muestra en un monitor. La falsa coloración de las intensidades fluorescentes, traducida en el monitor de la computadora como intensidades de píxeles, se aplica a la imagen para producir una imagen en color para que el analista la lea. Si el bioquímico marcó el polinucleótido de la muestra experimental desconocida con un tinte rojo y la muestra de polinucleótido de control con un tinte verde, y las coloraciones falsas imitan el marcado fluorescente, entonces la visualización de una mancha roja en la cuadrícula de la matriz final indica que polinucleótido desconocido hibridó abundantemente al ADNc fijado en esa ubicación en el portaobjetos de microarrays. Una mancha verde final indica que el polinucleótido de control hibridó abundantemente con el ADNc fijado en esa ubicación, una mancha amarilla indica que lo desconocido y los polinucleótidos de control hibridaron en cantidades relativamente iguales en esa ubicación en el microarray, y una mancha negra indica que ninguna muestra de polinucleótidos hibridados en esa ubicación.

g) Verificación de datos de microarrays

Los fabricantes de matrices comerciales e investigadores utilizarán otras técnicas para confirmar los hallazgos en un experimento de microarrays, el análisis de transferencia Northern y la reacción en cadena de la polimerasa con transcriptasa inversa (RT-PCR) son técnicas comúnmente utilizadas para realizar tales verificaciones, la técnica de transferencia Northern analiza muestras de ARN en una membrana de nylon, pero es análoga a la técnica de transferencia Southern. (McLachlan, Do, & Ambrise, 2019)

h) Análisis de datos del experimento del array

El análisis de los datos de microarrays de ADNc es un nuevo desafío para el bio estadístico, este paso probablemente será realizado por el personal

del laboratorio de biología durante los procesos de escaneo y visualización, pero su efecto sobre los datos en bruto debe comunicarse al bioestadístico. El análisis de datos por parte del bioestadístico puede requerir la aplicación de un procedimiento de normalización, el método más simple es una transformación lineal, a los datos de cada experimento para corregir las variables dentro de los procesos experimentales.

La normalización no lineal requerirá la aplicación de métodos estadísticos más sofisticados. Hay muchos factores de variabilidad experimental que deben tenerse en cuenta, como la cantidad y la pureza de los polinucleótidos, los cambios en la temperatura y la humedad relativa del entorno experimental, la eficiencia del marcado fluorescente, los resultados de hibridación y los efectos de saturación, y el aumento de la intensidad de fluorescencia de fondo niveles, la normalización con frecuencia requiere el uso de genes de limpieza o hebras de ARNm de referencia (agregadas a una muestra a un nivel específico y medible) durante el experimento. Los investigadores que realizan los experimentos pueden proporcionar información sobre los genes particulares de limpieza que se usan.

Algunos investigadores proponen repetir un experimento de microarrays en muestras replicadas para ayudar al bioestadístico a corregir la variabilidad entre muestras experimentales. Después de la normalización de los datos, se realiza un análisis de visualización por computadora para identificar similitudes o patrones en el perfil de expresión génica. (McLachlan, Do, & Ambrise, 2019)

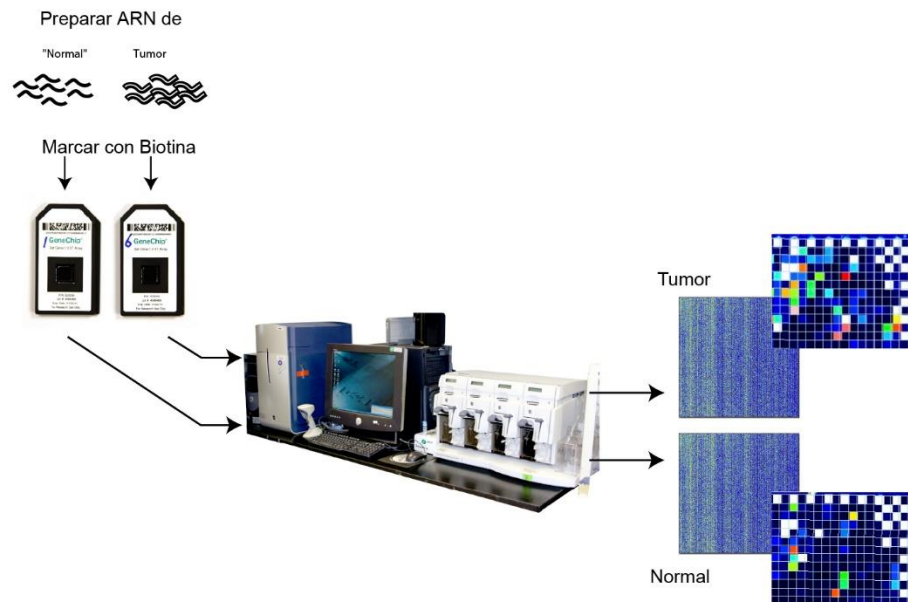


Figura 3. Proceso del experimento de un microarray, Fuente: (*Instituto Nacional del Genoma Humano, 2020*)

1.3.3.2. Limitaciones de la tecnología de microarrays

Existen muchas limitaciones para la precisión y la aplicación de la tecnología de microarrays de ADN, el más básico de los cuales es que la tecnología no mide los niveles de expresión génica o la abundancia de ARNm directamente, el bioestadístico debe comprender otras limitaciones para que las correcciones y los métodos de normalización se puedan realizar adecuadamente y para que los datos obtenidos de los experimentos puedan interpretarse adecuadamente.

Limitaciones sobre muestras de ADN y ARN

A continuación se listan algunas limitaciones de los experimentos de microarrays que los investigadores también debe tener en cuenta. (McLachlan, Do, & Ambrise, 2019)

1. Iniciamos con la limitación más común que es la disponibilidad de clones y / o muestras de tejido en cantidad suficiente.

2. La Limitación de los datos resultantes on respecto a la calidad de las muestras de ARN y ADNc, dependiendo de la purezaa y las concentraciones utilizadas.
3. Las diferentes moléculas de ARNm se someten a transcripción inversa a diversos grados de eficiencia, lo que resulta en lo que se conoce como sesgo de transcripción inversa.
4. Los tintes fluorescentes suelen tener una mayor afinidad de unión a un tipo de nucleótido, como la guanina (G); por lo tanto, las cadenas de ADNc que contienen más guanina en su secuencia aparecerán más brillantes al detectar la fluorescencia de los microarrays. Esto se conoce como sesgo de secuencia.
5. Otra limitación es el rango limitado con respecto a la fluorescencia si el estudio es un fenómeno no lineal.
6. Las mediciones de la expresión génica usando microarrays de ADNc actualmente solo proporcionan niveles de expresión relativos cuyas transcripciones de genes se expresan más abundantemente en una muestra en relación con las mismas transcripciones de genes en otra muestra de tejido o en un experimento en comparación con otro experimento.
7. La expresión génica solo puede proporcionar información parcial sobre las actividades en una celda. Hay muchas variaciones en la expresión, y el producto de un gen (es decir, la proteína) puede volverse más o menos activo porque se produce a un ritmo más rápido, otras proteínas lo degradan o se modifica químicamente..

8. Otra limitación es la hibridación de DNPJRNA por ser muy sensible a la temperatura y la fuerza iónica en solución, estas características dependen de las secuencias de bases en la cadena de ADN o ARN, ningún conjunto de condiciones experimentales es óptimo para todos los genes. Por lo tanto, algunos genes pueden no ser detectables porque la hibridación prevista simplemente no ocurre en las condiciones experimentales elegidas.

9. La medición de los microarrays son promedios de expresión genética, por lo tanto la tecnología tiene una resolución limitada con respecto a espacio y tiempo para detectar eventos moleculares transitorios en ciertos tipos de células.

1.3.4. Lenguajes de programación

En la informática se atribuye un englobe a las matemáticas, los matemáticos que son programadores han escrito sus programas y realizan sus pruebas con palabras matemáticas, es decir usan notaciones matemáticas y es de allí que nace los lenguajes de programación y cada uno es muy útil en algunos contextos. La presente investigación tiene como propósito extenderse sobre 3 idiomas diferentes con estilos de programación no trivial, destacando sus ventajas y características (Lee, 2018)

a) Implementación del lenguaje

Existen 3 formas que se pueden implementar como son: la interpretación del idioma, un idioma se compila en un lenguaje de máquina y un lenguaje puede implementarse mediante una combinación de los primeros dos métodos (Lee, 2018).

b) Compilación

Para que pueda realizarse la traducción de un programa a lenguaje de máquina se requiere de un programa de compilación que consta de varias

partes, el cual consta de un analizador que es el que lee un programa fuente y traduce de una forma intermedia llamada árbol de sintaxis abstracta que representa internamente el programa fuente. Después de que lee los datos el generador de código atraviesa nuevamente el AST y posteriormente se produce otro programa de ensamblaje, el cual no es un lenguaje de máquina pero está mucho más cerca, finalmente un ensamblador y un enlazador traducen de lenguaje ensamblador a un lenguaje a máquina. Haciendo que el programa esté listo para su ejecución (Lee, 2018).

En este proceso se ha completado la encapsulación mediante un compilador de datos de la plataforma, cuando nosotros pensamos que el compilador ejecuta nuestros programas no tenemos presentes las fases que ocurrieron, ahora presento las tres fases que ocurren: Primero se escribe el programa fuentes, luego se compila produciendo un programa ejecutable y por último se ejecuta el programa ejecutable.

Finalmente cuando haya terminado de ejecutarse se obtendrá el idioma fuente y el lenguaje de máquina, si el desarrollador realiza una modificación se tendrá que volver a compilar para que se sincronice nuevamente.

1.3.4.1. C y C++

Se dice que C y C++ nace después del desarrollo de un proyecto basado en UNIX y es diseñado en el año 1972 aproximadamente, UNIX tuvo que pasar por un largo proceso para poder crear un lenguaje de código abierto como es C++ que tiene como última actualización del estándar en el 2014, y existe un estándar que está en proceso desde el 2017. (Lee, 2018)

1.3.4.2. JAVA

Lee en (2018) partir de C++ que es muy potente, existen programadores que son muy cuidadosos al escribir sus códigos, pero los programas desarrollados con C++ tiene problemas con fugas de memoria, además otro de los grandes problemas es que si los desarrolladores olvidan un

objeto libre esto ocasiona inconvenientes en ejecutar el programa. A partir de estos y más problemas es que se diseñó una Máquina Virtual Java para agilizar el desarrollo.

1.3.4.3. PYTHON

Python es otro lenguaje orientado a objetos como C ++ y Java. A diferencia de C ++, Python es un lenguaje interpretado, Python ejecuta en diferentes plataformas como son Mac OS X, Linux y Microsoft Windows de Apple (Lee, 2018)

Es así como los programadores de todo el mundo se han preocupado por python desarrollando muchas bibliotecas y a su vez muchos programas, esto hizo que python viviera mucha popularidad debido a su portabilidad y la capacidad de sus bibliotecas. La comparación de Python lleva a una filosofía entre los desarrolladores a pensar que el uso de este lenguaje los conlleva a un éxito final, existen muchas versiones creadas en lo largo del tiempo.

1.3.4.4. R

López Perez (2015) R es un lenguaje y entorno de programación que, además, brinda un amplio abanico de herramientas estadísticas y gráficas, enriquecido con la posibilidad de cargar diferentes bibliotecas o paquetes con finalidades específicas de análisis estadístico o gráfico.

Otra definición que nos otorga R Project es que, (R-Project, 2017) R es un lenguaje y entorno para computación estadística y gráficos, es un proyecto GNU que es similar al lenguaje S y al entorno que fue desarrollado en los Laboratorios Bell (anteriormente AT&T, ahora Lucent Technologies) por John Chambers y sus colegas. R puede considerarse como una implementación diferente de S. Hay algunas diferencias importantes, pero gran parte del código escrito para S se ejecuta sin modificaciones bajo R.

R (2017) proporciona una amplia variedad de técnicas estadísticas (modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series temporales, clasificación, agrupamiento, ...) y gráficas, y es altamente extensible. El lenguaje S es a menudo el vehículo de elección para la investigación en metodología estadística, y R proporciona una ruta de código abierto para participar en esa actividad.

R es un conjunto integrado de instalaciones de software para la manipulación de datos, el cálculo y la visualización gráfica que incluye una instalación eficaz de manejo y almacenamiento de datos, un conjunto de operadores para cálculos en matrices, en particular matrices, (R-Project, 2017) Una colección grande, coherente e integrada de herramientas intermedias para el análisis de datos, facilidades gráficas para el análisis y visualización de datos en pantalla o en papel, y un lenguaje de programación bien desarrollado, simple y efectivo que incluye condicionales, bucles, funciones recursivas definidas por el usuario e instalaciones de entrada y salida. (R-Project, 2017)

1.3.5. Aprendizaje Automático

a) Soporte de máquinas de vectores

Los métodos SVM usan condiciones lineales para separar las clases entre sí. La idea es usar una condición lineal que separe las dos clases entre sí lo mejor posible (C. Aggaerwal, 2014).

En tal caso, la condición de división en el caso multivariante también se puede usar como condición independiente para la clasificación. Esto, un clasificador SVM, puede considerarse un árbol de decisión de un solo nivel con una condición de división multivariada muy cuidadosamente elegida. Claramente, dado que la efectividad del enfoque depende solo de un solo plano de separación, es crítico definir esta separación cuidadosamente (C. Aggaerwal, 2014)

Puede usarse tanto para los desafíos de clasificación como de regresión. SVM modela la situación creando un espacio de características, que es un espacio vectorial dimensional finito. El principio fundamental de SVM es separar el espacio de características en dos clases al encontrar el hiperplano que diferencia muy bien las dos clases. Los métodos SVM son sólidos para estudiar la interacción entre variantes comunes y variantes raras en una familia de muestra pequeña. Chen et al han desarrollado un método eficiente y efectivo para el clasificador SVM que utiliza un algoritmo genético paralelo de grano grueso CGPGA que puede usarse ampliamente para seleccionar conjuntamente el subconjunto de características y optimizar los parámetros para SVM en muchas aplicaciones prácticas de la ciencia biológica (Sen, Datta , & Mitra, 2019).

Aggaerwal (2014) En general, se supone para los datos dimensionales d que el hiperplano de separación tiene la forma $\bar{W} \cdot \bar{X} + b = 0$. Aquí W es un vector d -dimensional que representa los coeficientes del hiperplano de separación, y es constante. Sin pérdida de generalidad, puede suponerse (debido a la escala de coeficiente adecuada) que los dos vectores de soporte simétricos tienen la forma $\bar{W} \cdot \bar{X} + b = 1$ y $\bar{W} \cdot \bar{X} + b = -1$. Los coeficientes W y b deben aprenderse de los datos de entrenamiento D para maximizar el margen de separación entre estos dos hiperplanos paralelos. A partir del álgebra lineal elemental, se puede demostrar que la distancia entre estos dos hiperplanos es $2 / \|\bar{W}\|$. Maximizar esta función objetivo es equivalente a minimizar $\|\bar{W}\|^2 / 2$. Las restricciones del problema se definen por el hecho de que los puntos de datos de entrenamiento para cada clase están en un lado del vector de soporte. Por lo tanto, estas restricciones son las siguientes:

$$\begin{aligned} \bar{W} \cdot \bar{X} + b &= 1 \geq +1 \quad \forall i : y_i = +1 \\ \bar{W} \cdot \bar{X} + b &= -1 \leq -1 \quad \forall i : y_i = -1 \end{aligned}$$

b) Naïve Bayes

Naïve Bayes es un conjunto de algoritmos de aprendizaje supervisado que se basan en la aplicación del teorema de Bayes con el supuesto "ingenuo" de independencia condicional entre cada par de características dado el valor de la variable d e clase. El teorema de Bayes influye en la siguiente relación,

dada la variable de clase y vector de características dependientes. (scikit-learn, 2019)

Los aprendices y clasificadores ingenuos de Bayes pueden ser extremadamente rápidos en comparación con métodos más sofisticados. El desacoplamiento de las distribuciones de características condicionales de clase significa que cada distribución se puede estimar independientemente como una distribución unidimensional. Esto a su vez ayuda a aliviar los problemas derivados de la maldición de la dimensionalidad (scikit-learn, 2019)

Zhang (2016) El teorema de Bayes se puede usar para hacer predicciones basadas en el conocimiento previo y la evidencia actual, con la acumulación de evidencia, la predicción cambia; en términos técnicos, la predicción es la probabilidad posterior de que los investigadores estén interesados. El conocimiento previo se denomina probabilidad previa que refleja la suposición más probable sobre el resultado sin evidencia adicional. La evidencia actual se expresa como una probabilidad teniendo como resultado la probabilidad de un predictor.. El conjunto de datos de entrenamiento se utiliza para derivar la probabilidad. El teorema de Bayes se expresa formalmente mediante la siguiente ecuación.

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

Zhang (2016) Donde P (A) y P (B) son probabilidades de eventos A y B sin que se relacionen entre sí. P (A | B) es la probabilidad de A condicional en B y P (B | A) es la probabilidad de B condicional en A. En la clasificación ingenua de Bayes, A son eventos de resultado categóricos y B es una serie de predictores. La palabra "ingenuo" indica que los predictores son independientes entre sí y dependen del mismo valor de resultado. Por lo tanto, P (b1, b2, b3 | A) se puede escribir como P (b1 | A) × P (b2 | A) × P (b3 | A), lo que facilita mucho el proceso de cálculo.

Naïve Bayes usando la librería e1071

El paquete e1071 contiene una función llamada naiveBayes () que es útil para realizar la clasificación de Bayes. La función puede recibir datos categóricos y una tabla de contingencia como entrada. Devuelve un objeto de la clase "naiveBayes", este objeto se pasar para predecir los resultados de sujetos sin etiquetar (Zhang, 2016)

Naïve Bayes usando la librería caret

El paquete caret contiene la función train () que es útil para configurar una cuadrícula de parámetros de ajuste para una serie de rutinas de clasificación y regresión, se ajusta a cada modelo y calcula una medida de rendimiento basada en re muestreo..

c) K-NN

Proporciona funcionalidad para métodos de aprendizaje basados en vecinos no supervisados y supervisados. Los vecinos más cercanos no supervisados son la base de muchos otros métodos de aprendizaje, especialmente el aprendizaje múltiple y la agrupación espectral. (scikit-learn, 2019)

El principio detrás de los métodos del vecino más cercano o KNN es encontrar un número predefinido de muestras de entrenamiento más cercanas en distancia al nuevo punto y predecir la etiqueta a partir de ellas. El número de muestras puede ser una constante definida por el usuario (k-aprendizaje vecino más cercano), o puede variar en función de la densidad local de puntos (aprendizaje vecino basado en el radio). La distancia es en general, cualquier métrica: la distancia euclidiana estándar es la opción más común. Los métodos basados en los vecinos se conocen como métodos de aprendizaje automático no generalizados, ya que simplemente "recuerdan" todos sus datos de entrenamiento (posiblemente transformados en una estructura de indexación rápida, como un árbol de bolas o un árbol KD).

A pesar de ser simple, los vecinos más cercanos son exitosos por la gran cantidad de problemas de clasificación y regresión, los cuales puede resolver

rápidamente. Al ser un método que no usa los parámetros, siempre tiene éxito en problemas de clasificación donde el límite de decisión no es regular.

Willems (2018) El algoritmo KNN o k-vecinos más cercanos es uno de los algoritmos de aprendizaje automático más simples y es un ejemplo de aprendizaje basado en instancias, donde los nuevos datos se clasifican en función de instancias almacenadas y etiquetadas, específicamente entre los datos almacenados y la nueva instancia se calcula mediante algún tipo de medida de similitud, la cual se expresa mediante la distancia euclidiana, la similitud del coseno o la distancia de ManHattan.

Se puede decir, la similitud con los datos que ya estaban en el sistema se calcula para cualquier nuevo punto de datos que ingrese en el sistema. Luego, utiliza este valor de similitud para realizar el modelado predictivo.

(Willems, 2018) El modelado predictivo es clasificación, asignando una etiqueta o una clase a la nueva instancia, o regresión, asignando un valor a la nueva instancia. Si clasifica o asigna un valor a la nueva instancia depende, por supuesto, de cómo componga su modelo con KNN.

Willems (2018) El algoritmo de vecino más cercano a k agrega a este algoritmo básico que, después de calcular la distancia del nuevo punto a todos los puntos de datos almacenados, se ordenan los valores de distancia y se determinan los vecinos más cercanos de k. Se reúnen las etiquetas de estos vecinos y se utiliza un voto mayoritario o un voto ponderado para fines de clasificación o regresión.

Willems (2018) En otras palabras, cuanto mayor sea el puntaje para un determinado punto de datos que ya estaba almacenado, más probable es que la nueva instancia reciba la misma clasificación que la del vecino. En el caso de regresión, el valor que se asignará al nuevo punto de datos es la media de sus k vecinos más cercanos.

1.3.6. RStudio

RStudio es un entorno de desarrollo integrado (IDE) para R. Incluye una consola, editor de resaltado de sintaxis que admite la ejecución directa de código, así como herramientas para el trazado, el historial, la depuración y la gestión del espacio de trabajo (RStudio, 2020).

RStudio está disponible en código abierto y en ediciones comerciales y se ejecuta en el escritorio (Windows, Mac y Linux) o en un navegador conectado a RStudio Server o RStudio Server Pro (Debian / Ubuntu, Red Hat / CentOS y SUSE Linux) (RStudio, 2020).

1.4. Formulación del Problema.

¿Cuál es el mejor clasificador para la detección de subtipos de cáncer?

1.5. Justificación e importancia del estudio.

La presente investigación se facilitó la detección de cáncer caracterizando sus subtipos, para que de esta manera se genere un eficiente diagnóstico y a su vez el correcto tratamiento, agilizando la mejora del paciente.

El tipo de cáncer que se seleccionó es el cáncer de mama ya que es el segundo cáncer causante de muerte en el Perú, presentando 6 985 nuevos casos y 1 858 muertes en el año 2018. (El Comercio, 2019).

Es importante que con la tecnología podamos enfocarnos a la detección de enfermedades con mayor tasa de mortalidad para lograr un amplio aporte a nuestro entorno.

1.6. Hipótesis.

El algoritmo de clasificación que tendrá mejores resultados para la detección de subtipos de cáncer es el Support Vector Machine.

1.7. Objetivos.

1.7.1. Objetivo general.

Comparar clasificadores para la detección de subtipos de cáncer.

1.7.2. Objetivos específicos.

- a) Determinar el tipo de cáncer con el cual se trabajará.
- b) Caracterizar los subtipos de cáncer.
- c) Seleccionar clasificadores de aprendizaje automático.
- d) Implementar clasificadores seleccionador.
- e) Evaluar los resultados de acuerdo a los indicadores.

II. MATERIAL Y MÉTODO

2.1. Tipo y Diseño de Investigación.

2.1.1. Tipo de Investigación

El presente trabajo corresponde a un investigación de tipo Cuantitativa, aplicada y tecnológica, porque interviene con los conocimientos científicos dando el apoyo en las ciencias de la computación y sus resultados resolverán problemas reales en el campo de medicina.

2.1.2. Diseño de investigación

De acuerdo al tipo de investigación el diseño utilizado es Cuasi Experimental, debido a que genera interrogantes mediante la hipótesis y, se podrá resolver las circunstancias por efecto de naturaleza y de no conocer una selección aleatoria.

$$XY \rightarrow M$$

Donde:

X : Causa

Y : Efecto

M : Muestra

2.2. Población y muestra.

2.2.1. Población

La presente investigación tiene como población a los 8 algoritmos de clasificación más importantes según un artículo de dataflair. (DATAFLAIR TEAM, 2019), la tabla se puede visualizar en el Anexo2.

2.2.2. Muestra

La muestra determinada por 3 algoritmos que han sido elegidos por conveniencia para la investigación los cuales son, Naive Bayes Classifier Algorithm, K- Nearest Neighbours Algorithm, Support Vector Machine Algorithm.

2.3. Variables, Operacionalización.

2.3.1. Variables

2.3.1.1. Variable Independiente

Algoritmo de clasificación.

2.3.1.2. Variable Dependiente

Detección de subtipos de cáncer.

2.3.2. Operacionalización de Variables

Variable	Indicadores	Formula	Técnicas
Algoritmos de Clasificación	Tiempo de respuesta	$T = TF - TI$	
Detección de subtipos de cáncer	Precisión	$\frac{VP + VN}{Total}$	
	Error	$\frac{FP + FN}{Total}$	
	Sensibilidad	$\frac{VP}{Total\ de\ Positivos}$	

$$\text{Especificidad} = \frac{VN}{\text{Total de Negativos}}$$

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

La recolección de datos se realizará en la siguiente secuencia:

- Seleccionar las bases de datos con información con mayor precisión de datos biológicos.
- Descargar los datos biológicos que servirán para la investigación
- Caracterizar los datos biológicos encontrados a través de un software.
- Guardar los datos biológicos en una base de datos, previamente normalizados.
- Realizar pruebas.

2.5. Procedimiento de análisis de datos.

Para el análisis de datos en la presente investigación se utilizará una matriz de confusión y promedios para la validez de las pruebas realizadas.

2.6. Criterios éticos.

Veracidad: Los datos presentados serán verdaderos.

Derechos de Autor: El material usado para el desarrollo de la investigación estarán referenciados y citados.

Confidencial: Los datos biológicos que se usará para el desarrollo de la investigación contarán con seguridad y protección de la identidad de los pacientes.

2.7. Criterios de Rigor Científico.

La presente investigación se realizará siguiendo los juicios científicos, las cuales permiten garantizar la calidad de la investigación.

Es de esa manera que se seguirá con coherencia metodológica durante el desarrollo de la investigación, según la muestra de datos, los cuales se seleccionarán al azar.

III. RESULTADOS.

3.1. Resultados en Tablas y Figuras.

3.1.1. Resultados de Clasificador SVM

Tabla 1.

Matriz de Confusión SVM

Clases	Predicción				
	Basal	Her2	LumA	LumB	Normal
Basal	17	0	0	0	1
Her2	0	12	0	0	0
LumA	0	0	12	1	0
LumB	0	0	0	11	0
Normal	0	0	0	0	6

Nota: En la tabla 1 se aprecia la matriz de confusión generada por el desarrollo de la técnica de clasificación SVM, Fuente: Elaboración Propia

Se utilizó las fórmulas para hallar la precisión y el error.

$$\text{Precisión: } \frac{(17 + 12 + 12 + 11 + 6)}{60} = 0.9667$$

$$\text{Error: } \frac{(2)}{60} = 0.0333$$

Las medidas obtenidas en el desarrollo de la técnica SVM se muestran en el siguiente gráfico dando a conocer el porcentaje de sensibilidad y especificidad de cada uno de los subtipos de cáncer.

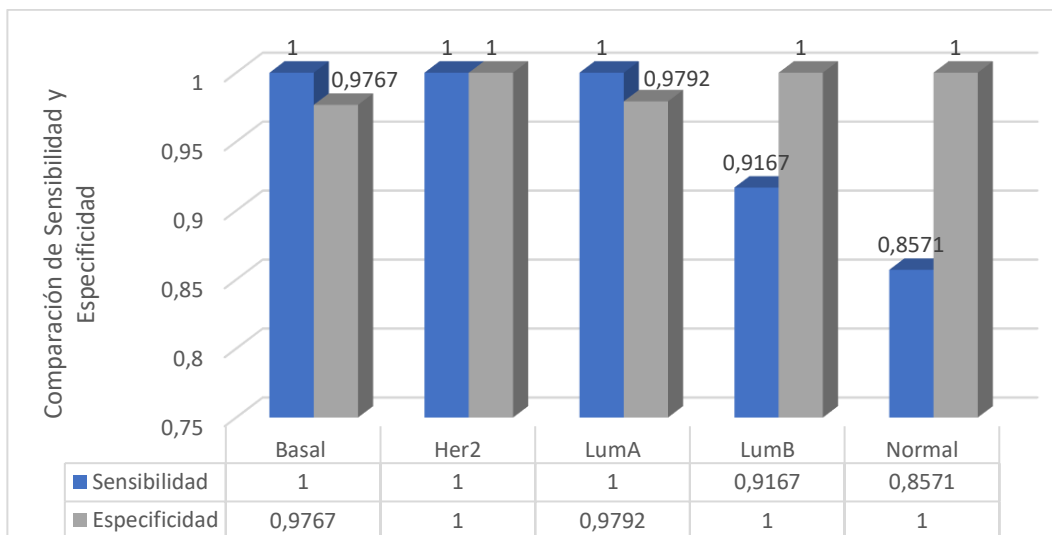


Figura 4 Grafico de Resultados de Sensibilidad y Especificidad con la tecnica SVM, Fuente: Elaboración Propia

$$Promedio (Sensibilidad): \frac{(1 + 1 + 1 + 0.9167 + 0.8571)}{5} = 0.95476$$

$$Promedio (Especificidad): \frac{(0.9767 + 1 + 0.9792 + 1 + 1)}{5} = 0.99118$$

Además el resultado del tiempo de respuesta del algoritmo fue de 0.36 segundos.

3.1.2. Resultados del Clasificador KNN

Tabla 2

Matriz de Confusión KNN

Clases	Predicción				
	Basal	Her2	LumA	LumB	Normal
Basal	32	4	0	0	0
Her2	0	10	0	3	0
LumA	0	0	1	0	0
LumB	0	0	0	9	0
Normal	0	0	0	0	1

Nota: En la tabla 2 se aprecia la matriz de confusión generada por el desarrollo del clasificador KNN, Fuente: Elaboración Propia

Se utilizó las fórmulas para hallar la precisión y el error.

$$\text{Precisión: } \frac{(32 + 10 + 1 + 9 + 1)}{60} = 0.8833$$

$$\text{Error: } \frac{(7)}{60} = 0.1166$$

Las medidas obtenidas en el desarrollo de la técnica KNN se muestran en el siguiente gráfico dando a conocer el porcentaje de sensibilidad y especificidad de cada uno de los subtipos de cáncer.

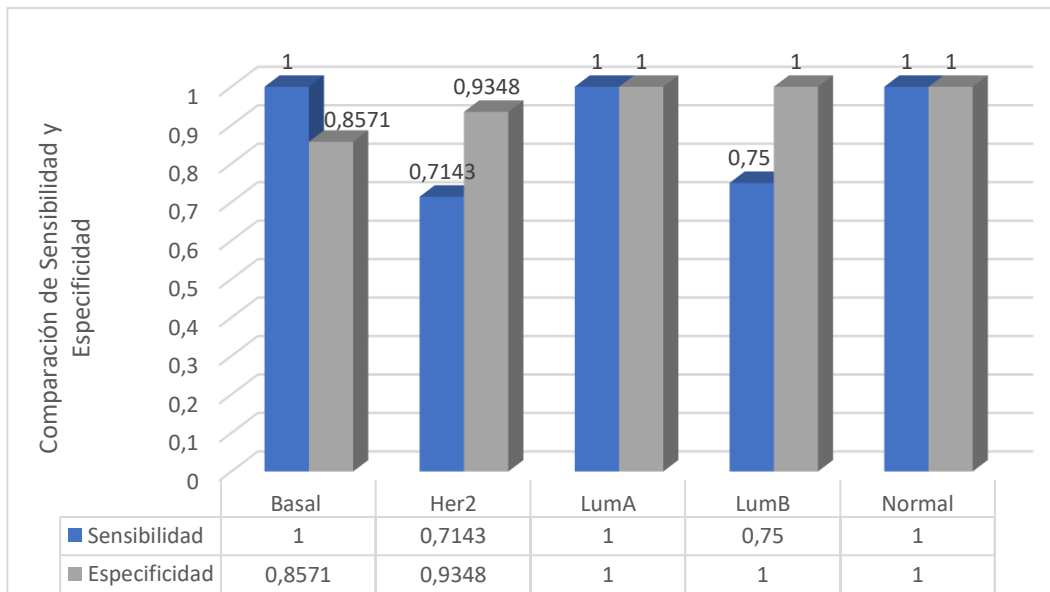


Figura 5 Gráfico de Resultados de Sensibilidad y Especificidad del Clasificador KNN, Fuente: Elaboración propia

$$\text{Promedio (Sensibilidad): } \frac{(1 + 0.7143 + 1 + 0.75 + 1)}{5} = 0.89286$$

$$\text{Promedio (Especificidad): } \frac{(0.8571 + 0.9348 + 1 + 1 + 1)}{5} = 0.95838$$

Además el resultado del tiempo de respuesta del algoritmo fue de 2.79 segundos.

3.1.3. Resultados del Clasificador Naive Bayes

Tabla 3.

Matriz de Confusión NB

Clases	Predicción				
	Basal	Her2	LumA	LumB	Normal
Basal	11	2	0	0	0
Her2	0	8	0	0	0
LumA	0	0	11	1	1
LumB	0	1	0	17	0
Normal	0	0	0	1	7

Nota: En la tabla 3 se aprecia la matriz de confusión generada por el desarrollo del clasificador NB. Fuente: Elaboración Propia

Se utilizó las fórmulas para hallar la precisión y el error.

$$\text{Precisión: } \frac{(11 + 8 + 11 + 17 + 7)}{60} = 0.90$$

$$\text{Error: } \frac{(6)}{60} = 0.10$$

Las medidas obtenidas en el desarrollo de la técnica NB se muestran en el siguiente gráfico dando a conocer el porcentaje de sensibilidad y especificidad de cada uno de los subtipos de cáncer.

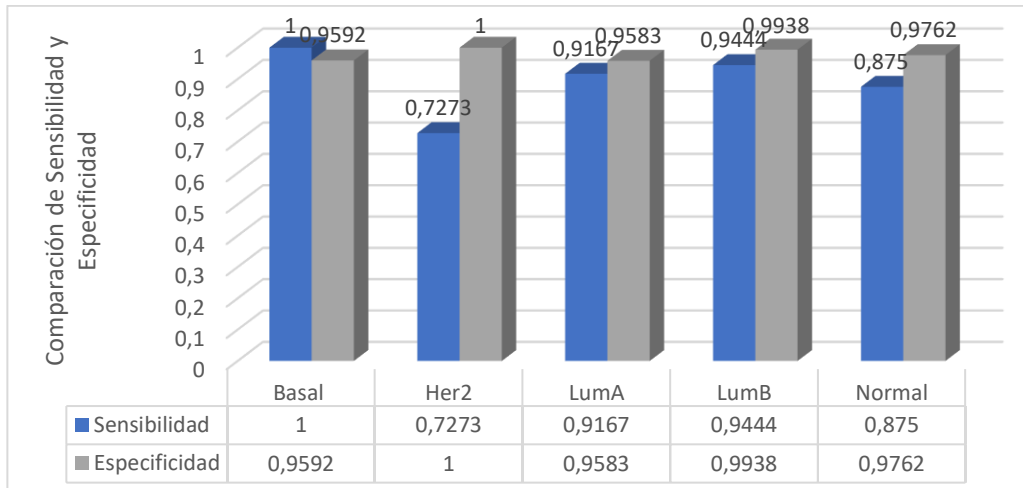


Figura 6 Gráfico de Resultados de Sensibilidad y Especificidad del Clasificador NB, Fuente: Elaboración Propia

$$\text{Promedio (Sensibilidad)}: \frac{(1 + 0.7273 + 0.9167 + 0.9444 + 0.875)}{5} = 0.89268$$

$$\text{Promedio (Especificidad)}: \frac{(0.9592 + 1 + 0.9583 + 0.9938 + 0.9762)}{5} = 0.9775$$

Además el resultado del tiempo de respuesta del algoritmo fue de 0.55 segundos.

3.1.4. Resumen de los Resultados

A continuación se muestra mediante una gráfica los resultados obtenidos del desarrollo de los clasificadores, analizando el promedio de los indicadores considerados.

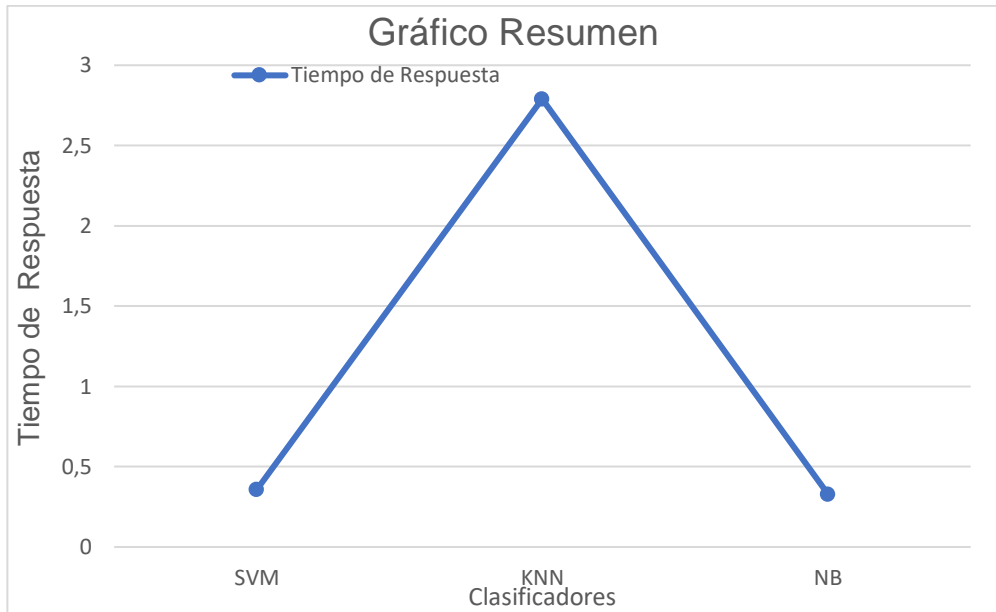


Figura 7. Gráfico Del Resumen de los Resultados (Tiempo de respuesta), Fuente: Elaboración Propia

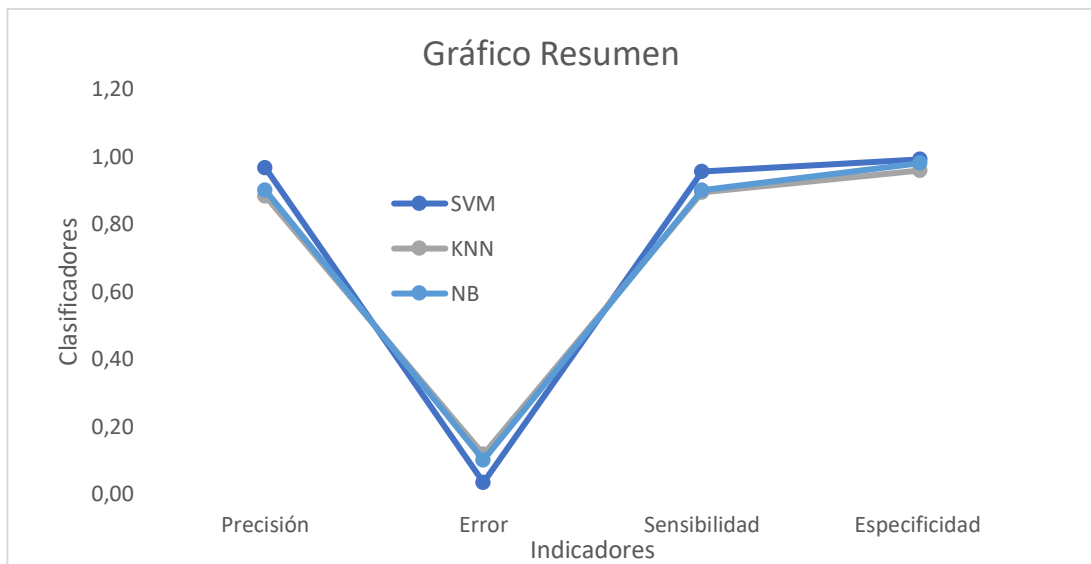


Figura 8. Gráfico del Resumen de los resultados de los indicadores (Precisión, error, Sensibilidad y Especificidad), Fuente: Elaboración Propia

3.2. Discusión de resultados.

Después del desarrollo de los tres clasificadores, se observa en la figura 7 que el clasificador SVM obtuvo un tiempo de respuesta de 0,36 segundos, el clasificador KNN con 2,79 segundos y por último obteniendo mejores

resultados se encuentra el clasificador Naive Bayes con un tiempo de respuesta de 0,33 segundos.

Además se obtuvieron resultados estadísticos entre los 3 clasificadores evaluando precisión, error, sensibilidad y especificidad como se puede visualizar en la figura 8, siendo el clasificador SVM el que obtuvo mejor precisión con una puntuación de 0.97, seguido del clasificador Naive Bayes que obtuvo una precisión de 0.90 y con menor precisión el clasificador KNN con 0.88 .El clasificador SVM obtuvo la mínima tasa de error de 0.03, seguido del clasificador Naive Bayes con una tasa de error de 0.10 y por último el que obtuvo mayor tasa de error fue el clasificador KNN con 0.12. El clasificador SVM obtuvo como resultado de sensibilidad y especificidad de 0.95 y 0.99 respectivamente, así también el clasificador Naive Bayes obtuvo como resultado de sensibilidad y especificidad de 0.89 y 0.98 respectivamente y por último los resultados de sensibilidad y especificidad del clasificador KNN, fue de 0.89 y 0.96 respectivamente.

A comparación de las investigaciones citadas en los antecedentes de estudio que detectaron los subtipos de cáncer, observamos que una investigación realizada por (Yu Lin, Li, Akutsu, Ruan, & See, 2018) utilizaron la estrategia PAM 50 y obtuvieron resultados de precisión del 65 %, y utilizando la estrategia IMS obtuvieron el 87% de precisión siendo la más alta y en la presente investigación con el clasificador SVM obtuvo el 97% de precisión, siendo más alta.

Otra de las investigaciones es la desarrollada por (Ghongade & Wakde, 2017) desarrollo un método logrando una precisión de 97.32%, sensibilidad de hasta 97.45%, especificidad de aproximadamente 98.13%, en comparación con la presente investigación que obtiene un 97% de precisión, una sensibilidad de hasta 95 % y una especificidad de hasta 99%.

3.3. Aporte práctico.

La presente investigación se realizó una comparación de clasificadores para la detección de subtipos de cáncer utilizando datos biológicos sustraídos de los miRNA.

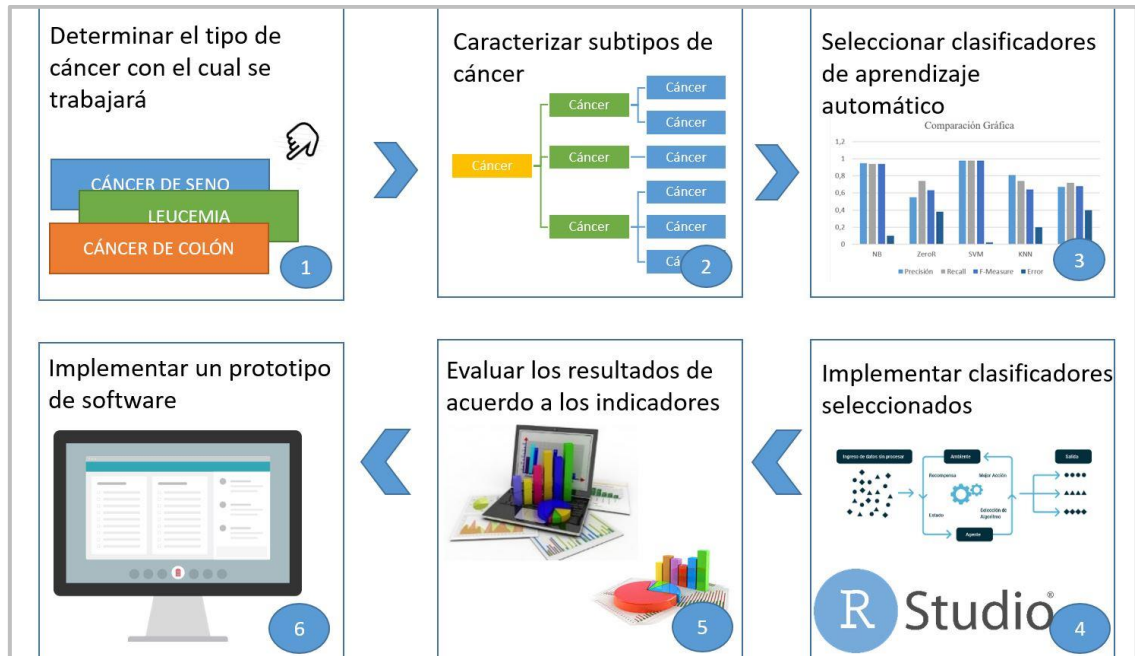


Figura 9 Método Propuesto, Fuente: Elaboración Propia

3.3.1. Determinación de los tipos de cáncer

La determinación del tipo de cáncer que se trabajó resaltando datos estadísticos de los tipos de cáncer que tienen alto índice de mortalidad en el mundo.

Como podemos visualizar en la Tabla 4 que describe el tipo de cáncer, los casos nuevos y las muertes estimadas; el tipo de cáncer más común en la lista es el cáncer de mama, con 271,270 casos nuevos, seguido del cáncer de pulmón y el cáncer de próstata. (NATIONAL CANCER INSTITUTE, 2019)

Tabla 4.

Estadísticas de incidencia y mortalidad por cáncer

Tipo de cáncer	Casos nuevos estimados	Muertes estimadas
Vejiga	80,470	17,670
Pecho(Femenino - Masculino)	268,600 – 2,670	41,760 – 500
Colon y Rectal (Combinado)	145,600	51,020
Endometrial	61,880	12,160
Cáncer de riñón (células renales y pelvis real)	73,820	14,770
Leucemia (Todos los tipos)	61,780	22,840
Hígado y conducto biliar intrahepático	42,030	31,780
Pulmón (incluido el bronquio)	228,150	142,670
Melanoma	96,480	7,230
No linfoma de Hodgkin	74,200	19,970
Pancreático	56,770	45,750
Próstata	174,650	31,620
Tiroides	52,070	2,170

Fuente: (NATIONAL CANCER INSTITUTE, 2019)

Teniendo en cuenta lo antes mencionado, la selección de tipo de cáncer es el cáncer de mama ya que es el cáncer que presenta más casos estimados en mujeres y hombres. Además de acuerdo al cálculo realizado por el centro nacional de epidemiología, prevención y control de enfermedades del MINSA, Lambayeque es una de las cinco regiones con mayor número de muertes por el cáncer de mama, como se puede apreciar en la siguiente figura. (MINSA, 2020)

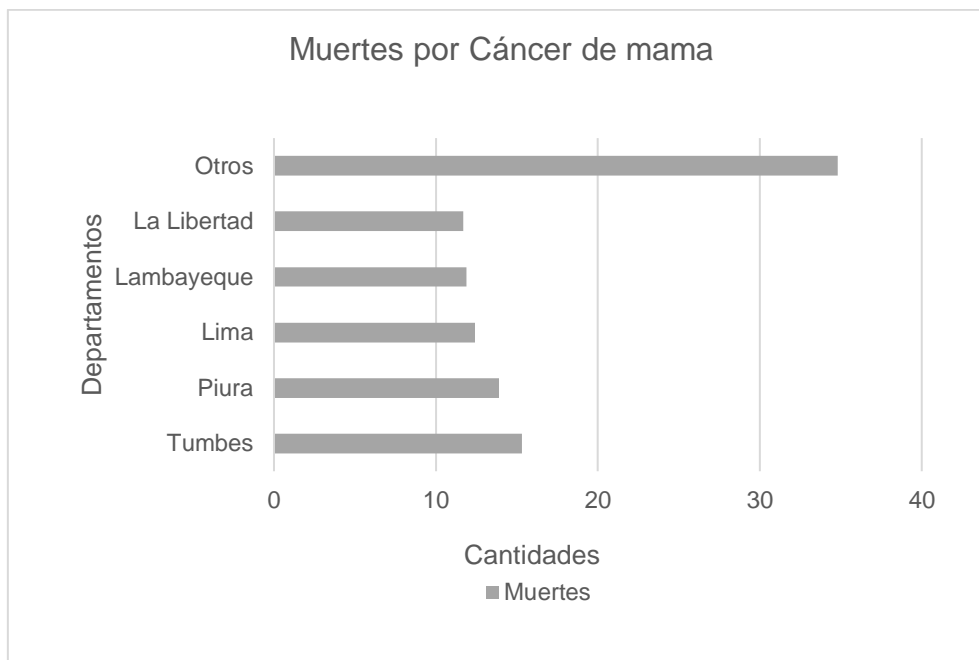


Figura 10 *Muertes por cáncer de mama, Fuente: (MINSA, 2020)*

3.3.2. Caracterización de subtipos de cáncer

La clasificación de los subtipos del cáncer de mama se puede diferenciar de acuerdo del orden de prevalencia, como se puede observar en la Tabla 5.

Tabla 5.

Subtipos de Cáncer de mama

	HR + / HER2-
Subtipos de Cáncer de mama	HR- / HER2-
	HR + / HER2 +
	HR- / HER2 +

Fuente: (Instituto Nacional del Cáncer, 2020)

3.3.2.1. Expresión de los Subtipos por Clasificación PAM50

Por el momento los subtipos se encuentran en entornos de investigación, pero se trabajará con los ya encontrados.

Tabla 6.

Caracterización de los Subtipos de Cáncer de mama

Subtipos		Basado en expresiones ER, PR y HER2
de	"Luminal A"	ER + y/o PR +, HER2 -
Cáncer	"Triple negativo" BL	ER - , PR-, HER2 -
de mama	"Luminal B"	ER + y/o PR +, HER2 +
	"enriquecido en HER2"	ER -, PR -, HER2 +

Fuente: Elaboración propia guiada por el artículo Expresión de los subtipos de cáncer de mama basados en los biomarcadores más importantes (Hadizadeh, Zaferani Arani, & Olya, 2018)

3.3.2.2. Genes expresados

Gracias a una investigación en la que se analizó datos clinicopatológicos, explorados en los repositorios públicos más grandes como NCBI GEO (<https://www.ncbi.nlm.nih.gov/geo>) y EBI ArrayExpress (<http://www.ebi.ac.uk/arrayexpress>) distingue al subgrupo de cánceres de mama con su respectiva expresión genética, utilizando **10086** muestras.

Tabla 7.

Subtipos de Cáncer de mama y su Expresión Genética

Categoría	Gen
Normal V Cáncer	ABCA8
	ADH1B
	ASPM
	AURKA
	BUB1B
	CCNB1
	CCNB2
	CDC20

CDK1
CENPA
CEP55
CKS2
COL10A1
CXCL10
CXCL11
CXCL2
CXCL9
DLGAP5
DTL
FABP4
FOSB KRT14
KRT15
KRT5
MELK
MMP1
NEK2
NUSAP1
OXTR
PBK
PRC1
PTN
RRM2
S100P
SFRP1
SPP1
SYNM
TGFB3
TOP2A
TPX2
UBE2C
WIF1

Basal – like		AGR2
		CA12
		DHRS2
		ELF5
		EN1
		ESR1
		FABP7
		FOXA1
		GABRP
		GATA3
		KRT6B
		MLPH
		NAT1
		PIP
		PROM1
		ROPN1B
		SCB1D2
		SCG2A2
		SCNN1A
		TFF1
		TFF3
		TFF1
		TFF3
		TFF1
		TFF3
Enriquecido	con	CALML5
HER2		CEACAM6
		CLCA2
		CRISP3
		ERBB2
		ESR1
		FGG
		GRB7

Luminal A

KMO
KYNU
NPY1R
PGAP3
PNMT
S100A8
S100A9
S100P
SCUBE2
STARD3
TFAP2B
ABAT
AGR2
AGTR1
BMPR1B
CA12
CPB1
DACH1
ERBB4
ESR1
FABP7
GATA3
GFRA1
GREB1
IGF1R
MMP1
NAT1
NPY1R
PGR
PROM1
RARRES1
S100A8
SCINA2

	TBC1D9
	TFF1
	TFF3
Luminal B	AGR2
	ARMT1
	CA12
	DHRS2
	ESR1
	FABP7
	GABRP
	GATA3
	KRT6B
	NAT1
	PROM1
	SFRP1
	SLPI
	TFF1
	TFF3
Luminal C	COL10A1
	CXCL9
	ESR1
	FABP7
	GABRP
	GATA3
	IFI44L
	SCGB2A2
	TFF1
Apocrina	CALML5
	CLCA2
	CPB1
	CRISP3
	ERBB4
	ESR1

IGF1R
 KYNU
 MMP1
 NPY1R
 S100A8
 S100A9
 SERPINA3
 TFF1

Fuente: Elaboración Propia guiada de El análisis de los datos de la expresión génica de microarrays 10086 revela los genes que sub clasifican los subtipos intrínsecos de cáncer de mama. (Hsuan & Ming-Ta, 2017)

3.3.3. Selección de clasificadores de aprendizaje automático

Para la selección de los clasificadores se analizó características específicas e importantes de los clasificadores como es el tiempo de CPU en el entrenamiento, la precisión, recall, F-Measure y el error de algunos algoritmos, y los mejores en cada análisis serán elegidos para el desarrollo de la presente investigación.

3.3.3.1. Prueba de tiempo de CPU

Para analizar la prueba de tiempo de CPU se muestra en la tabla 6 a los algoritmos, NB, PRISM, J48, KNN, SVM y MLP comparados con el tiempo que demora su entrenamiento. (Martins, y otros, 2020)

Tabla 8.

Estadística de Clasificadores en Tiempo de CPU

	NB	PRISM	J48	KNN	SVM	MLP
Tiempo de respuesta	0.62 ms	61.87 ms	3.85 ms	0.36 ms	311.25 ms	62.011 ms

Fuente: (Martins, y otros, 2020)

3.3.3.2. Prueba de cálculos

Revisamos investigaciones para definir la precisión, recall, F-Measure y error en algoritmos como son, Naive Bayes, Zero R, SVM, KNN, MLP.

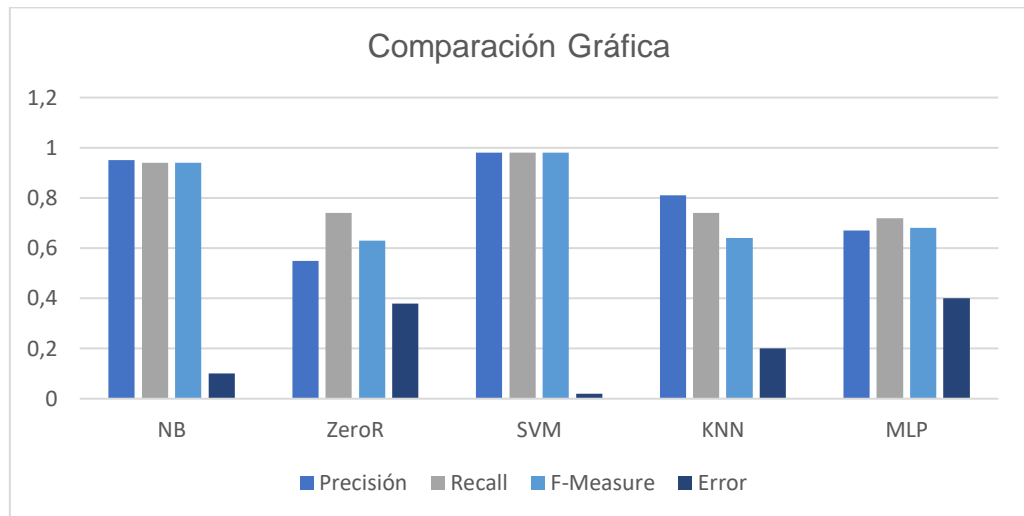


Figura 11. Comparación gráfica de algoritmos

Fuente: Elaboración propia guiada por (Gharouit & Nfaoui, 2017)

Tabla 9 .

Comparación de algoritmos

Algoritmos	Precisión	Recall	F-Measure	Error
NB	0,95	0,94	0,94	0,10
ZeroR	0,55	0,74	0,63	0,38
SVM	0,98	0,98	0,98	0,02
KNN	0,81	0,74	0,64	0,20
MLP	0,67	0,72	0,68	0,40

Nota: En la tabla 9 describe los resultados de la comparación de 5 algoritmos teniendo en cuenta 4 cálculos, observamos que para la precisión, el Recall y F-Measure los algoritmos SVM y el NB tiene mayor porcentaje. Así mismo observamos que el algoritmo SVM tiene 0.02 de error, siendo el que menor porcentaje de error posee seguido del algoritmo NB con 0.10 de error. (Gharouit & Nfaoui, 2017)

Se seleccionó a los algoritmos KNN, SVM Y NB, después de haber realizado el análisis de los indicadores

3.3.4. Implementación de clasificadores

3.3.4.1. Análisis de datos

Para el análisis de datos se utilizó el lenguaje de programación R, con datos extraídos de Microarrays personalizados por Agilent Human – UNC, el laboratorio que realiza experimentos con microarrays de expresión genética humana.

Los experimentos de microarray son una herramienta muy popular, a nivel bioinformático, se recibe una matriz de intensidad con la que calculamos una matriz normalizada dentro del pre procesamiento, además se determina los genes que se encuentran más expresados por estadística inferencial y por último se determina los patrones de comportamiento más comunes a distintos genes que se realiza a través de la estadística exploratoria y métodos de clasificación. (NCBI, 2018)

Los microarrays cuentan con columnas y filas, las columnas son condiciones como arrays, ensayos, casos, muestras, factores experimentales) y las filas ubica sondas o conjuntos de sondas que este caso son genes en el pre procesamiento.

Los datos pre procesados serán extraído de la base de datos de NCBI (Centro Nacional de Información Biotecnológica) <http://www.ncbi.nlm.nih.gov/geo/> , que utiliza la herramienta **GEO2R** que realiza un análisis que identifica genes que se expresan diferencialmente en condición experimentales mediante la comparación de diferentes muestras de conjuntos de datos GEO.

En esta investigación se utilizó el proyecto de bioinformática **GSE10886**, el cual presenta 200 muestras obtenidas de tejidos recién congelados y fijados en formalina, embebidos en parafina, se usaron para seleccionar estadísticamente muestras prototípicas y genes para los subtipos biológicos de cáncer de mama, Las predicciones del subtipo biológico en

un gran conjunto de pruebas de microarrays combinados mostraron importancia pronóstico en todos los pacientes.

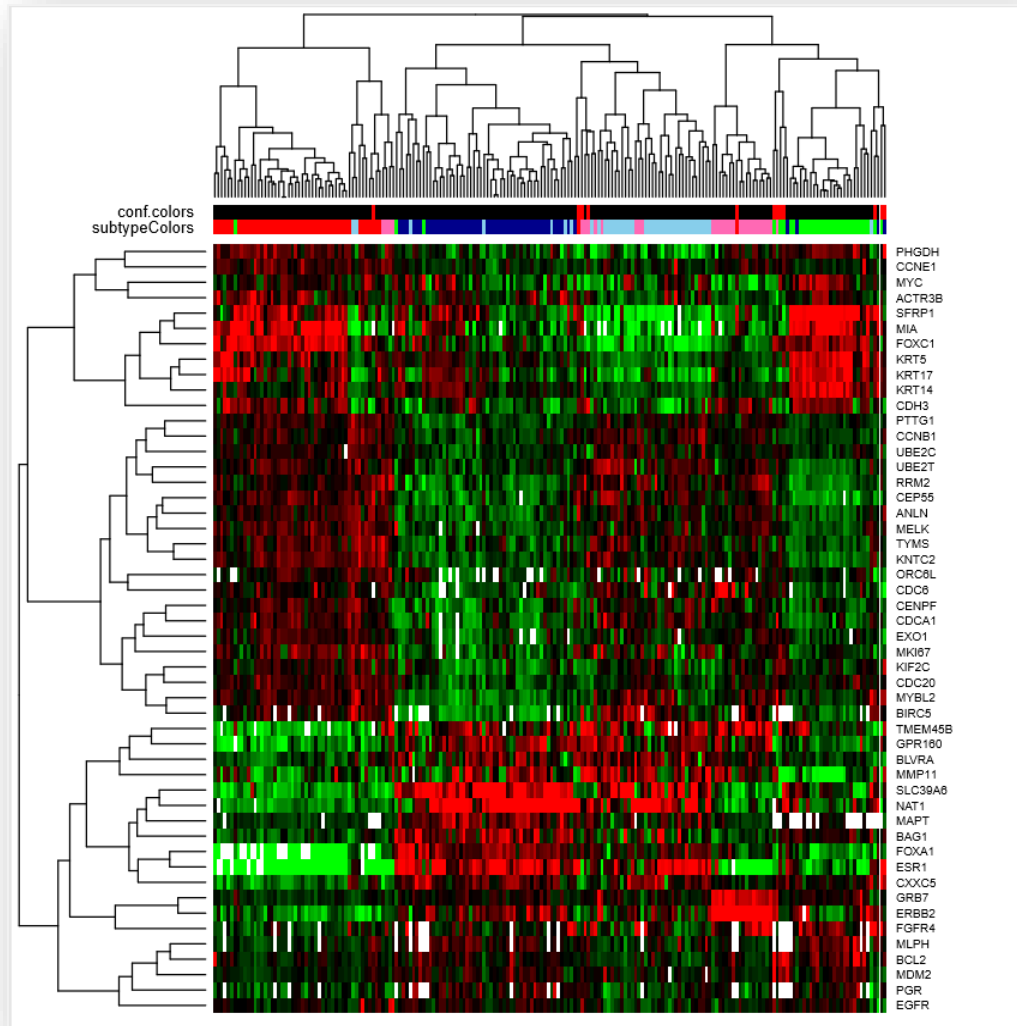


Figura 12. Secuencias de experimentos de 232 muestras

El análisis se realizó en RStudio, y el código se encuentra en un paquete llamado PAM50 en el repositorio de los experimentos del centro nacional de información biotecnológica, se analizó el código encontrando el uso de las siguientes librerías:

BiocManager, este paquete permite acceder al repositorio de paquete del Proyecto Bioconductor, que se encuentran disponible en una versión de lanzamiento para el uso diario, el uso de este paquete ayuda a los

usuarios a instalar con precisión los paquetes de la versión adecuada. (RDocumentación, 2020)

BiocVersion, este paquete proporciona información de repositorios para la versión apropiada de Bioconductor. (RDocumentación, 2020)

heatmap.plus, este paquete también llamado mapa de calor con comportamiento más sensible, tiene de base la función `heatmap()`, contiene dos parámetros opcionales como son `RowSideColors` y `ColSideColors`, y es el paquete que ayuda en esta conversión del microarray de la matriz de colores. (Rdocumentación, 2020)

impute, el paquete `impute`, no ayuda a la imputación de datos de microarrays.

Después de la normalización del bioproyecto **GSE10886**, y la ayuda del análisis experimental PAM50, se obtuvo el dataset que contiene 200 muestras con el porcentaje de pertenencia de cada subtipo de cáncer y el resultado al subtipo que pertenece. Todos los datos se encuentran almacenados en una tabla de Excel en la carpeta RECURSOS situados en el disco C del ordenador, se pueden apreciar en el **Anexo**.

3.3.4.2. Desarrollo del clasificador SVM

En la presente investigación se utilizó SVM para clasificar los pesos de las sondas pertenecientes a cada subtipo de cáncer, que anteriormente ya fueron normalizados, SVM es un clasificador para datos supervisados y además aporta el uso de diferentes núcleos de acuerdo a los datos que se clasificaron. Para el desarrollo se utilizó el núcleo radial básico, por la complejidad de los datos dispersos, que seleccionará secciones de cada conjunto de sondas.

a) Notación Matemática

La notación matemática del Kernel Radial, es la siguiente:

$$K(x, y) = e^{(-\gamma \sum_{j=1}^p (x_{ij} - y_{ij})^2)}$$

Con la fórmula de Kernel Radial o también llamado núcleo RBF evaluaremos los puntos de datos ente X y Y siendo:

X: una matriz de datos, un vector o una matriz dispersa.

Y: un vector de respuesta con la etiqueta para cada fila.

Y:

Si γ es muy grande entonces obtenemos límites de decisión fluctuantes y ondulantes silenciosos que explican una gran variación y sobreajuste.

Si γ es pequeño, la línea o límite de decisión es más suave y tiene poca varianza.

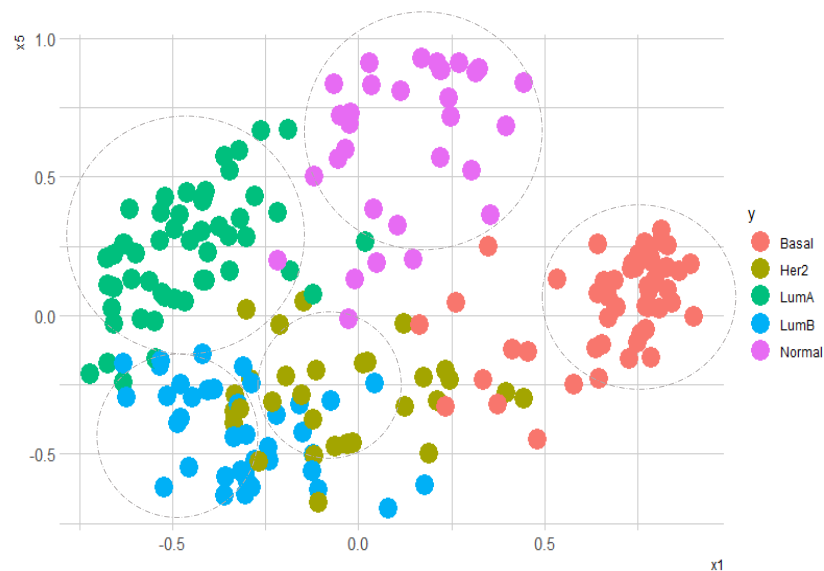


Figura 13 *Diseño de dispersión de datos*

Fuente: Elaboración propia

b) Código Fuente

Se utilizó las siguientes clases (“Normal”, “Basal”, “Her2”, “LumA”, “LumB”) además se usa la estrategia One-Versus-All, que consiste en ajustar K SVMs distintos, cada uno comparando una de las K clases frente a las restantes K-1 clases. (RStudio, 2020)

Tabla 10.

Identificación de Datos

Línea	Código Fuente
3	#Librerías
4	Library(caret)
5	Library(e1071)
6	
7	#Identificación de datos
8	datos <- data.frame(muestra= data_final\$Muestra, x1 = data_final\$Basal)
9	x2 = data_final\$Her2, x3 = data_final\$LumA,
10	x4 = data_final\$LumB, x5 = data_final\$Normal, y = data_final\$Resultado

Se utilizaron librerías como (e1071), que es esencial para el desarrollo de SVM, la librería (caret) para configurar las entradas y salidas y además para seleccionar el grupo de elementos que servirán como entrenamiento y prueba, además se utilizó la librería (ggplot2) para los gráficos. Como se muestra se realiza la identificación de los datos de la tabla 'data_final'.

Se inicia por la muestra, los pesos de los subtipos Basal, Her2, LumA y LumB se renombró como x1, x2, x3 y x4 respectivamente; además la lista de pesos del cáncer de mama tipo normal se renombró como x5 y a los Resultados se le renombró como y.

Tabla 11.

Selección de datos de entrenamiento y datos de prueba

Línea	Código Fuente
18	#Selección de una muestra del 70%de los datos y el 30% serán datos de prueba
19	set.seed(200)
20	tamaño.total <- nrow(datos)
21	tamaño.entreno <- round(tamaño.total * 0.7)
22	datos.muestra <- sample(1:tamaño.total , size=tamaño.entreno)
23	datos.entreno <- datos[datos.muestra,]
24	datos.test <- datos[-datos.muestra,]
25	

Como se observa de las 200 observaciones que es nuestra base de datos, se seleccionó 140 observaciones que servirán como datos de entrenamiento, y 60 observaciones que servirán como datos de prueba.

Tabla 12

Respuesta del Modelo SVM

Línea	Código Fuente
26	#Ejecución del modelo SVM
27	modeloE <- svm(y~. , data = datos.entreno,
28	method = "C-classification" , kernal = "radial" ,
29	gamma = 0.1, cost=10)
30	
31	#Predicción de los restantes
32	predicción <- predict(modeloE, data=datos.test)
33	
34	#Observación del modelo
35	Summary(modeloE)

Como se observa en la tabla 12 , al modelo SVM se le llamó 'modeloE' en el que llamamos a la función **svm()** en el que insertaremos a 'y' que son los resultados y los datos de entrenamiento llamados 'datos.entreno' además se seleccionó el núcleo trabajando con 'radial'. Luego realizamos la predicción de los datos, en el cual se utilizó la función **predict()** en el que insertamos el modelo y los datos de pruebas llamados 'datos.test'.

```
> summary(modeloE)
Call:
svm(formula = y ~ ., data = datos.entreno, method = "c-classification", kernal = "radial", gamma = 0.1,
     cost = 10)

Parameters:
  SVM-Type:  C-classification
 SVM-Kernel: radial
         cost: 10

Number of Support Vectors: 73

 ( 15 14 16 12 16 )

Number of classes: 5

Levels:
 Basal Her2 LumA LumB Normal
```

Figura 14. Resumen

En la figura 14 se utilizó la función **summary()** para ver el resumen de la ejecución del modelo. Y en figura 9 se muestra el resultado de la

clasificación mostrando los vectores de soporte de cada subtipo de cáncer.

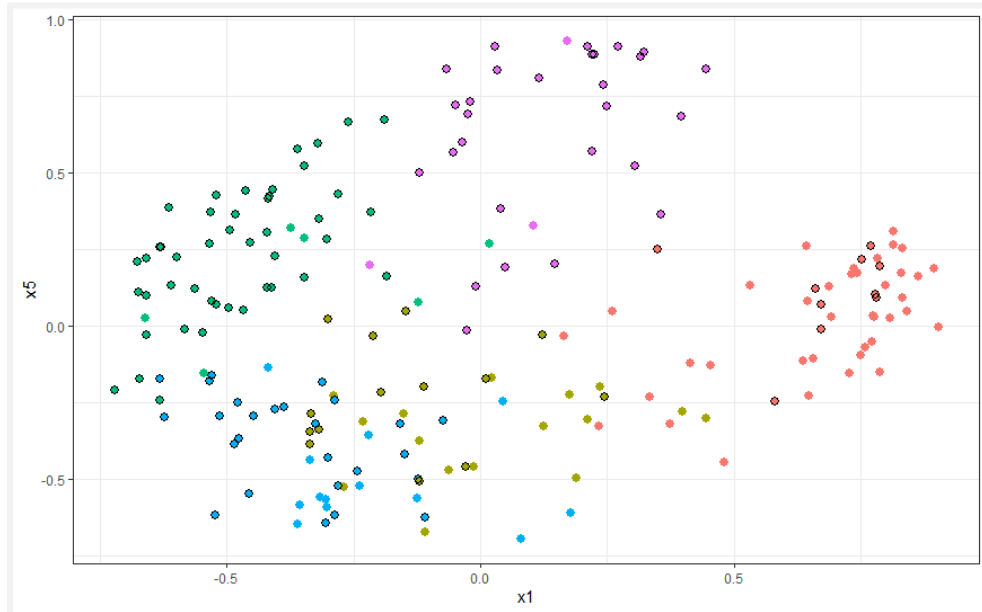


Figura 15. Resultado de la selección de Vectores de Soporte

3.3.4.3. Desarrollo del clasificador KNN

El clasificador KNN es un método de clasificación supervisada y se usará por su imparcialidad y porque no hace suposiciones previas de los datos, además es simple y efectiva, y es fácil de implementar, se usó para analizar la clasificación por mayoría de votos de los k vecinos de los pesos de cada uno de los subtipos de cáncer.

a) Notación Matemática

KNN usa un punto en el espacio asignado a la clase más frecuente entre los K ejemplos de entrenamiento más cercano, generalmente se usa la distancia euclidiana:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^p (x_{ri} - x_{rj})^2}$$

El método KNN supone que los vecinos más cercanos nos dan la mejor clasificación y esto se hace utilizando todos los atributos.

Siendo x un vector en un espacio característico multidimensional, que pertenece a una de las clases de la clasificación.

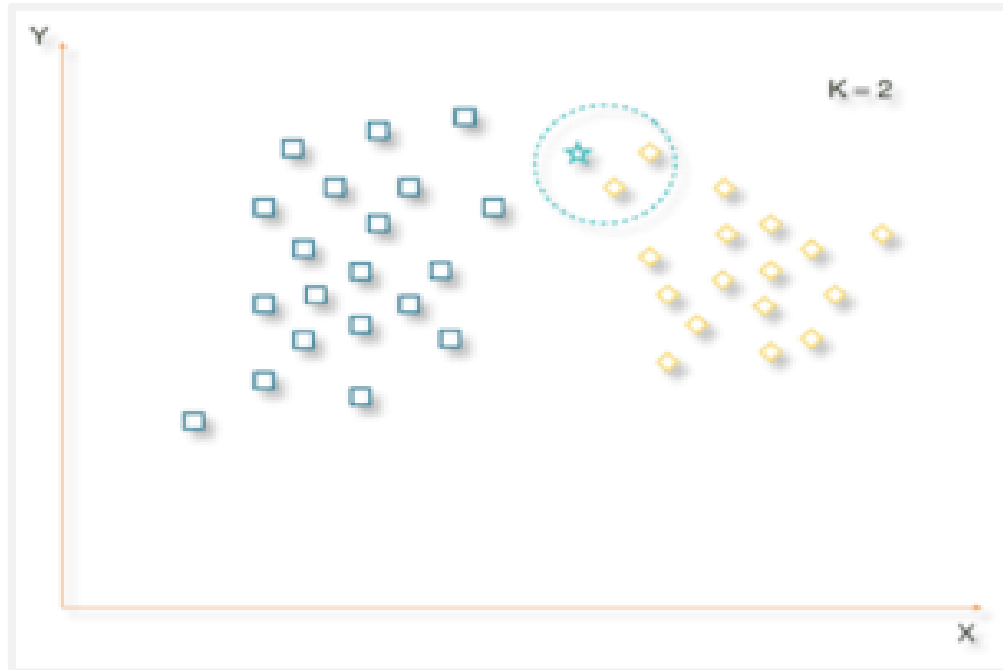


Figura 16 Ejemplo KNN

b) Código Fuente

Se inició importando la dataset y guardando los datos en 'dataknn'. Posteriormente se ejecutó el comando `str()`, para visualizar si los datos están estructurados y `head()` para observar las cabecera de los datos. Y se obtuvo como se muestra en la figura 17

Tabla 13

Importación de datos

Línea	Código Fuente
2	#Inicio guardando lo datos en dataknn
3	dataknn <- data_final
4	
5	#Utilizaremos el comando str para ver si los datos están estructurados
6	str(dataknn)
7	head(dataknn)

```

Console ~/
> str(data_final)
tibble [200 x 7] (s3: tbl_df/tbl/data.frame)
 $ Muestra : chr [1:200] "s1" "s2" "s3" "s4" ...
 $ Basal   : num [1:200] -0.677 -0.629 -0.658 0.211 -0.462 0.323 0.171 -0.609 -0.406 -0.32 ...
 $ Her2    : num [1:200] -0.044 -0.295 -0.131 -0.469 -0.381 -0.544 -0.623 -0.068 -0.304 -0.659 ...
 $ LumA    : num [1:200] 0.722 0.791 0.839 0.327 0.793 0.185 0.396 0.775 0.566 0.778 ...
 $ LumB    : num [1:200] 0.003 -0.02 -0.017 -0.779 -0.192 -0.831 -0.714 0.058 -0.101 -0.329 ...
 $ Normal  : num [1:200] 0.207 0.256 0.219 0.909 0.441 0.891 0.929 0.13 0.228 0.593 ...
 $ Resultado: chr [1:200] "LumA" "LumA" "LumA" "Normal" ...
> head(data_final)
# A tibble: 6 x 7
  Muestra Basal Her2 LumA LumB Normal Resultado
  <chr>    <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 s1      -0.677 -0.044 0.722 0.003 0.207 LumA
2 s2      -0.629 -0.295 0.791 -0.02 0.256 LumA
3 s3      -0.658 -0.131 0.839 -0.017 0.219 LumA
4 s4       0.211 -0.469 0.327 -0.779 0.909 Normal
5 s5      -0.462 -0.381 0.793 -0.192 0.441 LumA
6 s6       0.323 -0.544 0.185 -0.831 0.891 Normal
> |

```

Figura 17 Ejecución de str() y head()

Como se observa en la figura 21 la primera fila contiene el nombre de las muestras, y antes de normalizarlas se deben eliminar temporalmente, para esto se utilizó [-x], siendo x el número de la columna que se eliminará. Resultando como se muestra en la figura 18..

```

Console ~/
> head(dataknn)
# A tibble: 6 x 6
  Basal Her2 LumA LumB Normal Resultado
  <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 -0.677 -0.044 0.722 0.003 0.207 LumA
2 -0.629 -0.295 0.791 -0.02 0.256 LumA
3 -0.658 -0.131 0.839 -0.017 0.219 LumA
4 0.211 -0.469 0.327 -0.779 0.909 Normal
5 -0.462 -0.381 0.793 -0.192 0.441 LumA
6 0.323 -0.544 0.185 -0.831 0.891 Normal
> |

```

Figura 18. Eliminación de columna 1

Tabla 14.

Normalización

Línea	Código Fuente
17	#Normalizar los datos numéricos
18	
19	normalizar <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }

```

20
21 #normalización de todas las variables
22 Data_norm <- as.data.frame(lapply (dataknn[1:5], normalizar) )
23
24 #verificación de la normalización
25 Sumary(data_norm)
26

```

Posteriormente se realizó la normalización de los datos, utilizando function(x) guardados en el objeto 'normalizar', se tiene en cuenta que solo se normalizan los datos numérico solo se seleccionaron las cinco clases, excluyendo la última clase que corresponde a los resultados, por último se ejecutó Summary(), para visualizar el resumen de los datos ya normalizados.

```

Console ~/
> summary(data_norm)
  Basal      Her2      LumA      LumB      Normal
Min.   :0.0000  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000  Min.   :0.0000
1st Qu.:0.2125  1st Qu.:0.2692  1st Qu.:0.2258  1st Qu.:0.3218  1st Qu.:0.2595
Median :0.3732  Median :0.4017  Median :0.4966  Median :0.4764  Median :0.4517
Mean   :0.4431  Mean   :0.4422  Mean   :0.4977  Mean   :0.4931  Mean   :0.4512
3rd Qu.:0.6526  3rd Qu.:0.6204  3rd Qu.:0.7551  3rd Qu.:0.6967  3rd Qu.:0.5928
Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000  Max.   :1.0000
>

```

Figura 19. Datos en consola normalizados

Luego se realizó la creación de conjunto de datos se seleccionó 140 datos para entrenamientos guardándolos en 'data_entreno' y 60 datos para las pruebas guardándolos en 'data_prueba'. Como observamos anteriormente no se tomó en cuenta la clase Resultado por eso tenemos que incluir creando nuevos conjuntos de datos, data_entreno_x y data_prueba_x, como se muestra.

Tabla 15.

Creación de conjunto de datos

Línea	Código Fuente
29	#Creación de conjuntos de datos
30	data_entreno <- data_norm [1:140,]
31	data_prueba <- data_norm [141:200,]
32	
33	#incluimos al Resultado

```

34 data_entreno_x <- dataknn [1:140, 6 ]
35 data_prueba_x <- dataknn [141:200, 6 ]
36 Summary(dataknn)

```

Para la creación del modelo que ejecutará el algoritmo se llama a la librería 'Class' que contiene la función knn(), el modelo se llama 'modeloKNN' llamando a los datos de entrenamiento y a los datos de prueba y además K que es la variable de los vecinos más cercanos, usualmente k siempre es la raíz cuadrada del total de los elementos.

Tabla 16.

Modelo KNN

Línea	Código Fuente
38	#Modelo KNN
39	install.packges("class")
40	library(class)
41	cl = data_entreno_x[, 1, drop = TRUE]
42	modeloKNN <- knn (train = data_entreno, test = data_prueba, cl, k=15)

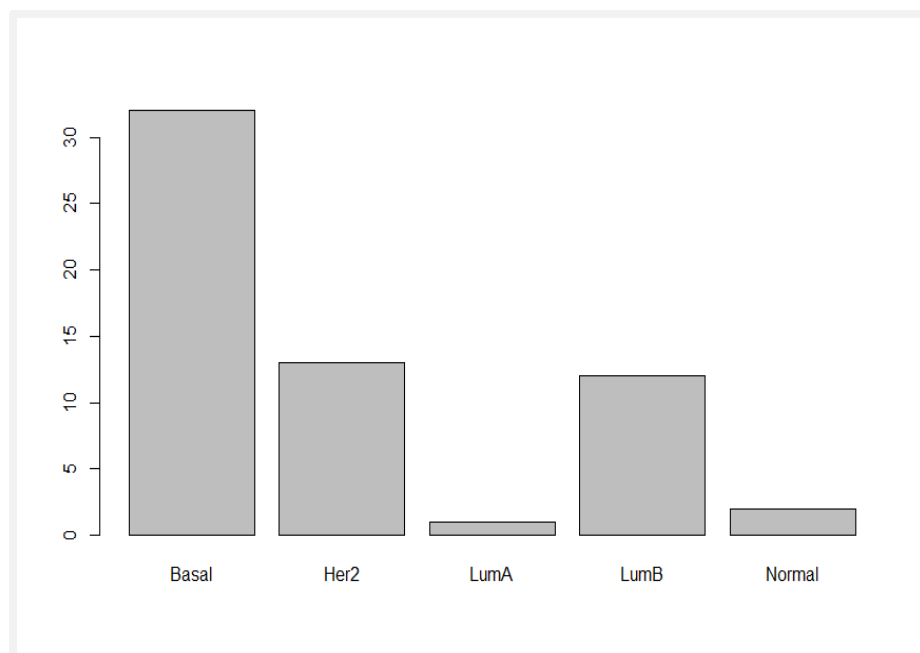


Figura 20. Resultado del Modelo KNN

3.3.4.4. Desarrollo del clasificador Naive Bayes

Se desarrolló el clasificador Naive Bayes para resolver el problema de clasificación siguiendo un enfoque probabilístico.

a) Notación Matemática

El principio del algoritmo de Naive Bayes se basa en el teorema de Bayes, también conocido como la regla de Bayes. Se representa como:

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$

$P(A | B)$: probabilidad condicional de que ocurra el evento A, dado el evento B

$P(A)$: probabilidad de que ocurra el evento A

$P(B)$: probabilidad de que ocurra el evento B

$P(B | A)$: probabilidad condicional de que ocurra el evento B, dado el evento A

b) Código Fuente

Tabla 17.

Inicio de implementación

Línea	Código Fuente
1	#Implementación NB
2	
3	library(e1071)
4	dataNB = data_final

Para la implementación se inició llamando a la librería e1071 y se cargó los datos guardándolos en 'dataNB', como se muestra en la siguiente figura.

Tabla 18.

Modelo NB

Línea	Código Fuente
-------	---------------

```

9 modeloNB = naiveBayes(as.factor(Resultado) ~. , data=dataNB)
10
11 modeloNB
12 pred = predict(modeloNB, dataNB)

```

Posteriormente se creó el modelo del clasificador Naive bayes , llamando a la función `naivebayes()` que tiene como entrada al Resultado que es la observación 7 que contiene los resultados de cada muestra y a la base de datos 'dataNB'.

Como se muestra en la línea 13, se crea el predictor llamado 'pred' llamando a la función `predict()` para verificar el modelo del clasificador creado.

Tabla 19.

Matriz de Confusión

Línea	Código Fuente
15	<code>matrizc = table (pred, dataNB\$Resultado)</code>
16	
17	<code>library(caret)</code>
18	<code>plot(matrizc)</code>

Como se muestra en la figura 30 se creó la matriz de confusión utilizando la predicción y la lista de los Resultados, por último se llamó a la matriz por medio de la función `plot()`, obteniendo la figura 31.

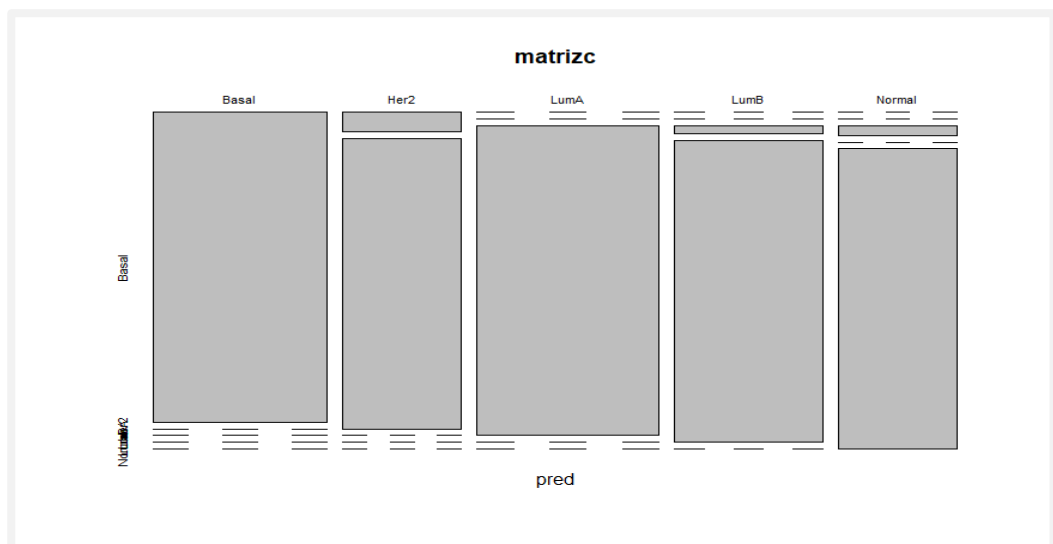


Figura 21. Resultado de modelo NB

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- a)** En la presente investigación se determinó utilizar el tipo de Cáncer de Mama por ser el que posee la mayor alta de tasa de mortalidad.
- b)** La caracterización de subtipos de cáncer se basó en la clasificación PAM 50, y la seleccionándose a los subtipos, Luminal A, Luminal B, Triple Negativo o Basal y al Enriquecido de Her2.
- c)** Para la selección de clasificadores de aprendizaje automático se realizó un análisis de los indicadores, precisión, error y tiempo de respuesta, siendo elegidos los clasificadores Support Vector Machines, K-Nearest Neighbor y al clasificador Naive Bayes.
- d)** La implementación de los clasificadores de la investigación se realizó utilizando el lenguaje de programación R por su amplia variedad de técnicas estadísticas y por la colección grande, coherente e integrada de herramientas intermedias para el análisis de datos, además se concluye que este cumple con todos los objetivos planteados haciendo uso de los algoritmos seleccionados.
- e)** Se concluye que en la evaluación de los resultados obtenidos se utilizó los indicadores de tiempo de respuesta, precisión, error, sensibilidad y especificidad evidenciando que el clasificador con menor tiempo de respuesta fue clasificador Naive Bayes con 0.33 segundos y el que obtuvo el mejor performance en los indicadores precisión, error, sensibilidad y especificidad fue el clasificador SVM con 97%, 3%, 95% Y 99% respectivamente.

4.2. Recomendaciones

- a)** Se recomienda utilizar otro tipo de cáncer, para ampliar las investigaciones en nuestro campo, con respecto a la medicina.
- b)** Se recomienda estar pendientes de las actualizaciones de las investigaciones del Instituto Nacional del Cáncer.
- c)** También se recomienda utilizar otros clasificadores, para realizar más comparaciones.
- d)** Con respecto a la implementación se recomienda utilizar otro lenguaje como Python.
- e)** Finalmente se recomienda tomar los resultados encontrados para ampliar otros estudios similares.

V. REFERENCIAS.

- Alam, J., Alam, S., & Hossan, A. (2018). Multi-Stage Lung Cancer Detection and Prediction Using Multi-class SVM Classifier .
- Buhigas, J. (2018, Setiembre 03). *Machine Learning*. Retrieved from <https://puentesdigitales.com/2018/09/03/para-que-sirve-y-para-que-no-sirve-el-aprendizaje-automatico-machine-learning/>
- C. Aggaerwal, C. (2014). *Data Classification Algorithms and Aplications*.
- DATAFLAIR TEAM. (2019, 10 04). *DATAFLAIR*. Retrieved from <https://dataflair.training/blogs/machine-learning-classification-algorithms/>
- El Comercio. (2019, Enero 31). *El cáncer mató en 2018 a más de 33.000 personas en el Perú*. Retrieved from <https://elcomercio.pe/tecnologia/ciencias/cancer-mato-2018-33-000-personas-peru-noticia-602437-noticia/#>
- Gharouit, K., & Nfaoui, E. (2017). A Comparison of Classification Algorithms for Verbose Queries Detection Using BabelNet.
- Ghongade, R., & Wakde, D. (2017). Computer aided Diagnosis System for breast cáncer using, RF classifier. *IEEE WiSPNET 2017*, 5.
- Hadizadeh, M., Zaferani Arani, H., & Olya, M. (2018). Expression of Breast Cancer Subtypes Based on the Most Important Biomarkers: Comparison of Clinicopathological Factors and Survival.
- HIRALES, C. M. (2015). *ALGORITMO GENÉTICO PARALELO PARA LA CLASIFICACIÓN DE SUBTIPOS DE CÁNCER*. La paz Baja California Sur.
- Hsuan, L., & Ming-Ta, H. (2017). El análisis de los datos de la expresión génica de microarrays 10086 revela los genes que subclasifican los subtipos intrínsecos de cáncer de mama.
- Instituto Nacional del Cáncer. (06, Marzo 2015). *Diagnóstico*. Retrieved from <https://www.cancer.gov/espanol/cancer/diagnostico-estadificacion/diagnostico>
- Instituto Nacional del Cáncer. (2020). *Atlas del Genoma del Cáncer*. Retrieved from Bioinformática:
<https://www.cancer.gov/espanol/publicaciones/diccionario/def/bioinformatica>

- Instituto Nacional del Genoma Humano. (2020). *Tecnologías de Microarrays*. Retrieved from <https://www.genome.gov/es/genetics-glossary/Tecnologia-de-microarrays>
- J, G., Y, H., & Q, L. (2012). Classification Network of Gastric Cancer Construction based on Genetic Algorithms and Bayesian Network.
- Lee, K. (2018). *Foundations of Programming Languages*.
- López Perez, C. (2015). *R, lenguaje de programación y análisis estadístico de datos*. Ibergarceta.
- Martins, J., Sestrem Ochoa, L., Silva, L., Sales Medes, A., Villarrubia Gonzales, G., & De Paz Santana, J. (2020). PRIPRO: A Comparison of Classification Algorithms. *Applied Sciences*.
- MAULI, C. y. (2014). La identificación de biomarcadores del cáncer a partir de microarrays. *ONCOLOGIA*.
- McLachlan, G. J., Do, K.-A., & Ambrise, C. (2019). *Microarray Gene Expression Data*.
- MINSA. (2020). *MINSA*. Retrieved from MINSA: <https://www.gob.pe/minsa/>
- MS, B., TM, L., & CM, T. (1993). *dbEST--database for "expressed sequence tags"*.
- National Cancer Institute. (2015, Febrero 09). *Atlas del Genoma Humano*. Retrieved from <https://www.cancer.gov/about-cancer/understanding/what-is-cancer>
- NATIONAL CANCER INSTITUTE. (2019). *TYPES*. Retrieved from <https://www.cancer.gov/types/common-cancers>
- Organización Mundial de la Salud. (2019). *Who Cancer*. Retrieved from <https://www.who.int/cancer/about/facts/es/>
- Pise, B. U.-A. (2016). Lung Cancer Detection Using Bayasein Classifier and FCM Segmentation.
- Polat, K., & Sentuk, U. (2018). A Novel ML Approach to Prediction of Breast Cancer: Combining of mad normalization, KMC based feature weighting and AdaBoostM1 classifier. *IEEEXPLORE*.
- Prada, F. G. (2017, 09 15). *HIDDEN NATURE*. Retrieved from <https://www.hidden-nature.com/chips-de-adn-o-microarray/>
- R. Catchpoole, Eoberts, A., & Kennedy, P. (2018). Variance-based Feature Selection for Classification of Cancer Subtypes Using Gene Expression Data. *IEEEXPLORE*.

- Ramírez, N. A., Gómez, E., & Forero, O. M. (2019). Clasificadores supervisados del cáncer de próstata a partir de imágenes de resonancia magnética en secuencias T2. *14th Iberian Conference on Information Systems and Technologies (CISTI)*, 4.
- Rdocumentación. (2020). *heatmap.plus.package*. Retrieved from <https://www.rdocumentation.org/packages/heatmap.plus/versions/1.3/topics/heatmap.plus.package>
- RDocumentación. (2020). *Bioconductor*. Retrieved from <https://www.rdocumentation.org/packages/BiocManager/versions/1.30.10>
- Rodríguez, T. (2018, setiembre 26). *Machine Learnig y Deep Learning*. Retrieved from <https://www.xataka.com/robotica-e-ia/machine-learning-y-deep-learning-como-entender-las-claves-del-presente-y-futuro-de-la-inteligencia-artificial>
- R-Project. (2017). What is R?
- RStudio. (2020). *RStudio*. Retrieved from <https://rstudio.com/products/rstudio/>
- scikit-learn. (2019). *scikit-learn*. Retrieved from https://scikit-learn.org/stable/supervised_learning.html#supervised-learning
- Sen, S., Datta , L., & Mitra, S. (2019). *Machine Learning and IoT*.
- Setiawan, A., Harjoko, A., Ratnaningsih, T., Suryani, E., Wiharto, & Palgunadi, S. (2018). Classification of Cell Types In Acute Myeloid Leukemia (AML) of M4, M5 and M7 Subtypes With Support Vector Machine Classifier. *IEEEXPLORE*.
- SZNAJDLEDER, P. (2012). *Java a fondo - estudio del lenguaje y desarrollo de aplicaciones - 2a ed*. México: Alfaomega.
- Turgut, S., Dagtekin, M., & Ensari, T. (2018). Microarray Breast Cancer Data Classification Using Machine Learning Methods. *IEEEXLPORE*.
- Willems, K. (2018). Machine Learning in R for beginners. *DataCamp*.
- World Cancer Research Fund International. (2012). *Estadísticas del Cáncer de mama*. Retrieved from <https://www.wcrf.org/dietandcancer/cancer-trends/breast-cancer-statistics>
- Yu Lin, C., Li, R., Akutsu, T., Ruan, P., & See, S. (2018). Deep Learning with Evolutionary and Genomic Profiles for Identifying Cancer Subtypes. *IEEEXPLORE*.

Zhang, Z. (2016). Naive Bayes classification in R.

ANEXOS.

Anexo 1 Resolución de aprobación del proyecto de investigación



FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO

RESOLUCIÓN N° 1814-2019/FIAU-USS

Chiclayo, 09 de diciembre de 2019

VISTO:

El Acta de Reunión N° de fecha 09 de diciembre de 2019., para la ejecución de la Tesis titulada: *"ANÁLISIS COMPARATIVO PARA LA DETECCIÓN DE SUBTIPOS DE CÁNCER"*, presentada por el(los) estudiante(s) **DIAZ BERNILLA NATALY MARLENE** de la Escuela Académico Profesional de **INGENIERÍA DE SISTEMAS** y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: *"La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas."*;

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

SE RESUELVE:

ARTÍCULO 1°: APROBAR, el Proyecto de Tesis denominado *"ANÁLISIS COMPARATIVO PARA LA DETECCIÓN DE SUBTIPOS DE CÁNCER"*, perteneciente a la Línea de Investigación **INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE - INFRAESTRUCTURA, TECNOLOGÍA Y MEDIO AMBIENTE**, a cargo del(los) estudiante(s) **DIAZ BERNILLA NATALY MARLENE**, de la Escuela Académico Profesional de **INGENIERÍA DE SISTEMAS**.

ARTÍCULO 2°: ESTABLECER, que la inscripción de la Tesis se realice a partir de emitida la presente resolución, y tendrá una vigencia máxima de 02 años.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE


UNIVERSIDAD SEÑOR DE SIPÁN S.A.
Dr. Andrés Alberto Ruiz Gómez
DECANO DE LA FACULTAD DE INGENIERÍA
ARQUITECTURA Y URBANISMO


UNIVERSIDAD SEÑOR DE SIPÁN S.A.C.

Mg. Luis Roberto Carrea Colchado
SEC. ACADEMICO FACULTAD DE INGENIERIA
ARQUITECTURA Y URBANISMO

Cc: Dirección de Investigación, CPGYT, Interesados, Archivo

ADMISIÓN E INFORMES

074 481610 - 074 481632

CAMPUS USS

Km. 5, carretera a Pimentel

Chiclayo, Perú

www.uss.edu.pe

Anexo 2 Población de Algoritmos

Tabla 20 Lista de clasificadores de aprendizaje automático

N°	Clasificación de aprendizaje automático: 8 algoritmos para aspirantes de ciencia de datos.
1	Logistic Regression Algorithm
2	Naïve Bayes Algorithm
3	Decision Tree Algorithm
4	K-Nearest Neighbours Algorithm
5	Support Vector Machine Algorithm
6	Random Forest Algorithm
7	Stochastic Gradient Descent Algorithm
8	Kernel Approximation Algorithm

Fuente: Elaboración propia guiado por DataFlair (*DATAFLAIR TEAM, 2019*)

Anexo 3 Resumen de Resultados SVM

```

Console ~/
Number of Classes: 5

Levels:
Basal Her2 LumA LumB Normal

>
> #mostrar grafico de comportamiento de sondas de cada subtipo
> #plot (modeloE, datos.entreno, x1 ~ x5,
> # slice = list (x2 = 3, x3 = 4, x4 = 5))
> #plot (svm1, datos.entreno, x5 ~ x1,
> # slice = list (x2 = 3, x3 = 4, x4 = 5))
>
> #hallar prediccion
> prediccion <- predict (modelosvm, datos.test)
> Matriz <- table (datos.test $ y, prediccion)
>
> confusionMatrix(Matriz)
Confusion Matrix and Statistics

      prediccion
      Basal Her2 LumA LumB Normal
Basal   17    0    0    0    1
Her2    0   12    0    0    0
LumA    0    0   12    1    0
LumB    0    0    0   11    0
Normal  0    0    0    0    6

Overall Statistics

      Accuracy : 0.9667
      95% CI   : (0.8847, 0.9959)
      No Information Rate : 0.2833
      P-Value [Acc > NIR] : < 2.2e-16

      Kappa : 0.9574

      McNemar's Test P-value : NA

Statistics by Class:

      class: Basal class: Her2 class: LumA class: LumB class: Normal
Sensitivity      1.0000      1.0      1.0000      0.9167      0.8571
Specificity      0.9767      1.0      0.9792      1.0000      1.0000
Pos Pred Value   0.9444      1.0      0.9231      1.0000      1.0000
Neg Pred Value   1.0000      1.0      1.0000      0.9796      0.9815
Prevalence       0.2833      0.2      0.2000      0.2000      0.1167
Detection Rate   0.2833      0.2      0.2000      0.1833      0.1000
Detection Prevalence 0.3000      0.2      0.2167      0.1833      0.1000
Balanced Accuracy 0.9884      1.0      0.9896      0.9583      0.9286
>
>
> proc.time()-t # FIN DEL CRONÓMETRO
      user system elapsed
      0.03   0.03   0.36
>

```

Figura 22 Resumen de Resultados SVM

Anexo 4 Resumen de Resultados KNN

```
Console ~/ /
> confusionMatrix(Matriz)
Confusion Matrix and Statistics

      x
y      Basal Her2 LumA LumB Normal
Basal  32   4   0   0   0
Her2   0  10   0   3   0
LumA   0   0   1   0   0
LumB   0   0   0   9   0
Normal 0   0   0   0   1

Overall Statistics

          Accuracy : 0.8833
          95% CI   : (0.7743, 0.9518)
    No Information Rate : 0.5333
    P-Value [Acc > NIR] : 7.387e-09

          Kappa   : 0.8052

McNemar's Test P-Value : NA

Statistics by Class:

                Class: Basal Class: Her2 Class: LumA Class: LumB Class: Normal
Sensitivity                1.0000    0.7143    1.00000    0.7500    1.00000
Specificity                0.8571    0.9348    1.00000    1.0000    1.00000
Pos Pred Value             0.8889    0.7692    1.00000    1.0000    1.00000
Neg Pred Value            1.0000    0.9149    1.00000    0.9412    1.00000
Prevalence                 0.5333    0.2333    0.01667    0.2000    0.01667
Detection Rate             0.5333    0.1667    0.01667    0.1500    0.01667
Detection Prevalence      0.6000    0.2167    0.01667    0.1500    0.01667
Balanced Accuracy         0.9286    0.8245    1.00000    0.8750    1.00000
>
>
> proc.time()-t # FIN DEL CRONÓMETRO
  user  system elapsed
  0.41   0.26   2.79
>
>
```

Figura 23 Resumen de Resultados KNN

Anexo 5 Resumen de Resultados NB

```

Console ~/
      Normal
Y      [,1]      [,2]
Basal  0.007333333 0.1892072
Her2   -0.291363636 0.1409835
LumA   0.212342105 0.2159517
LumB   -0.394333333 0.1747239
Normal 0.665782609 0.2773231
>
>
> pred = predict(modelNB, testData)
> Matriz <- table (testData$Resultado, pred)
>
>
> library(caret)
>
>
> confusionMatrix(Matriz)
Confusion Matrix and Statistics

      pred
      Basal Her2 LumA LumB Normal
Basal   11    2    0    0     0
Her2    0    8    0    0     0
LumA    0    0   11    1     1
LumB    0    1    0   17     0
Normal  0    0    1    0     7

Overall Statistics

          Accuracy : 0.9
          95% CI   : (0.7949, 0.9624)
    No Information Rate : 0.3
    P-value [Acc > NIR] : < 2.2e-16

          Kappa : 0.8726

Mcnemar's Test P-value : NA

Statistics by Class:

                Class: Basal Class: Her2 Class: LumA Class: LumB Class: Normal
Sensitivity          1.0000      0.7273      0.9167      0.9444      0.8750
Specificity          0.9592      1.0000      0.9583      0.9762      0.9808
Pos Pred Value       0.8462      1.0000      0.8462      0.9444      0.8750
Neg Pred Value       1.0000      0.9423      0.9787      0.9762      0.9808
Prevalence           0.1833      0.1833      0.2000      0.3000      0.1333
Detection Rate       0.1833      0.1333      0.1833      0.2833      0.1167
Detection Prevalence 0.2167      0.1333      0.2167      0.3000      0.1333
Balanced Accuracy    0.9796      0.8636      0.9375      0.9603      0.9279
> plot(Matriz)
>
> proc.time()-t # FIN DEL CRONÓMETRO
  user  system elapsed
 0.10   0.03   0.33
>

```

Figura 24 Resumen de los Resultados NB

Tabla 21 DATASET

Muestra	Basal	Her2	LumA	LumB	Normal	Resultado
S1	-0,677	-0,044	0,722	0,003	0,207	LumA
S2	-0,629	-0,295	0,791	-0,020	0,256	LumA
S3	-0,658	-0,131	0,839	-0,017	0,219	LumA
S4	0,211	-0,469	0,327	-0,779	0,909	Normal
S5	-0,462	-0,381	0,793	-0,192	0,441	LumA
S6	0,323	-0,544	0,185	-0,831	0,891	Normal
S7	0,171	-0,623	0,396	-0,714	0,929	Normal
S8	-0,609	-0,068	0,775	0,058	0,130	LumA
S9	-0,406	-0,304	0,566	-0,101	0,228	LumA
S10	-0,320	-0,659	0,778	-0,329	0,593	LumA
S11	0,316	-0,559	0,217	-0,701	0,876	Normal
S12	-0,360	-0,540	0,808	-0,330	0,575	LumA
S13	-0,415	-0,282	0,744	-0,246	0,423	LumA
S14	-0,722	0,132	0,523	0,437	-0,211	LumA
S15	0,115	-0,510	0,410	-0,761	0,809	Normal
S16	0,034	-0,292	0,442	-0,600	0,832	Normal
S17	-0,453	-0,089	0,606	-0,106	0,269	LumA
S18	-0,021	-0,556	0,606	-0,566	0,730	Normal
S19	-0,119	0,023	0,382	-0,488	0,500	Normal
S20	-0,548	0,005	0,434	0,176	-0,022	LumA
S21	-0,632	-0,229	0,826	-0,018	0,257	LumA
S22	-0,532	-0,295	0,693	-0,202	0,371	LumA
S23	-0,562	-0,201	0,667	0,129	0,120	LumA
S24	0,243	-0,525	0,295	-0,742	0,785	Normal
S25	0,029	-0,615	0,492	-0,645	0,910	Normal
S26	0,224	-0,498	0,243	-0,800	0,884	Normal
S27	-0,673	0,030	0,472	0,425	-0,174	LumA
S28	-0,632	0,302	0,435	0,433	-0,243	LumA
S29	0,220	-0,469	0,251	-0,766	0,884	Normal
S30	-0,419	-0,379	0,706	-0,117	0,413	LumA
S31	-0,054	-0,405	0,390	-0,567	0,566	Normal
S32	-0,067	-0,526	0,630	-0,643	0,836	Normal
S33	-0,533	0,287	0,171	0,422	-0,182	LumB
S34	-0,319	-0,338	0,534	-0,195	0,348	LumA
S35	-0,420	-0,426	0,557	-0,075	0,125	LumA
S36	-0,546	0,000	0,451	0,397	-0,155	LumA
S37	-0,495	-0,380	0,716	-0,202	0,311	LumA
S38	-0,632	0,079	0,323	0,432	-0,174	LumB
S39	-0,658	-0,038	0,658	-0,028	0,099	LumA
S40	-0,261	-0,496	0,798	-0,409	0,665	LumA
S41	-0,184	-0,177	0,331	0,098	0,159	LumA
S42	-0,218	-0,109	0,140	-0,043	0,197	Normal
S43	-0,048	-0,351	0,463	-0,625	0,719	Normal
S44	0,271	-0,599	0,315	-0,721	0,910	Normal

S45	-0,534	-0,329	0,807	-0,116	0,268	LumA
S46	-0,482	-0,251	0,784	-0,124	0,362	LumA
S47	-0,301	0,416	0,171	-0,106	0,021	Her2
S48	-0,280	-0,395	0,713	-0,256	0,429	LumA
S49	-0,674	-0,055	0,697	0,157	0,107	LumA
S50	-0,009	0,008	0,034	-0,394	0,128	Normal
S51	-0,348	-0,284	0,596	-0,067	0,285	LumA
S52	-0,614	-0,286	0,851	-0,170	0,383	LumA
S53	-0,410	-0,479	0,795	-0,166	0,445	LumA
S54	-0,421	-0,234	0,666	-0,050	0,303	LumA
S55	-0,529	-0,211	0,629	-0,016	0,078	LumA
S56	-0,448	-0,104	0,240	0,420	-0,297	LumB
S57	-0,347	-0,350	0,435	-0,015	0,158	LumA
S58	0,040	-0,622	0,315	-0,327	0,381	Normal
S59	-0,662	0,048	0,647	0,251	0,023	LumA
S60	-0,025	-0,286	0,510	-0,569	0,690	Normal
S61	0,396	-0,440	0,047	-0,767	0,681	Normal
S62	-0,146	0,262	0,139	-0,099	0,048	Her2
S63	0,782	-0,091	-0,663	-0,277	0,089	Basal
S64	0,455	-0,117	-0,549	-0,054	-0,131	Basal
S65	-0,477	0,325	0,278	0,442	-0,371	LumB
S66	-0,336	0,557	-0,047	0,371	-0,388	Her2
S67	-0,123	0,356	-0,290	0,648	-0,503	LumB
S68	-0,520	-0,212	0,534	0,105	0,067	LumA
S69	0,221	-0,153	0,073	-0,696	0,567	Normal
S70	-0,335	0,154	0,083	0,605	-0,439	LumB
S71	-0,623	0,093	0,316	0,448	-0,298	LumB
S72	-0,333	0,571	-0,094	0,128	-0,287	Her2
S73	0,446	-0,570	0,040	-0,777	0,838	Normal
S74	-0,216	-0,297	0,466	-0,354	0,369	LumA
S75	-0,412	-0,280	0,594	0,149	0,125	LumA
S76	-0,243	0,201	-0,111	0,783	-0,477	LumB
S77	-0,373	-0,284	0,603	-0,120	0,319	LumA
S78	-0,598	-0,243	0,755	-0,028	0,222	LumA
S79	-0,584	-0,090	0,535	0,251	-0,013	LumA
S80	-0,190	-0,456	0,714	-0,470	0,670	LumA
S81	-0,479	0,003	0,302	0,347	-0,250	LumB
S82	-0,346	-0,574	0,753	-0,289	0,521	LumA
S83	-0,496	-0,254	0,651	0,090	0,057	LumA
S84	-0,521	-0,294	0,844	-0,298	0,425	LumA
S85	0,356	0,046	-0,027	-0,493	0,362	Normal
S86	-0,073	0,226	-0,220	0,443	-0,309	LumB
S87	-0,658	0,060	0,575	0,218	-0,032	LumA
S88	0,245	0,447	-0,488	0,178	-0,234	Her2
S89	-0,195	0,359	0,003	0,168	-0,220	Her2
S90	-0,159	0,257	-0,046	0,391	-0,320	LumB
S91	0,148	-0,263	0,073	-0,175	0,202	Normal

S92	0,736	-0,179	-0,481	-0,352	0,185	Basal
S93	-0,418	-0,070	0,271	0,321	-0,138	LumB
S94	-0,212	0,270	0,235	0,015	-0,033	Her2
S95	0,349	-0,203	-0,277	-0,445	0,247	Basal
S96	-0,302	-0,240	0,530	-0,053	0,280	LumA
S97	-0,513	0,218	0,305	0,354	-0,294	LumB
S98	-0,036	-0,607	0,508	-0,518	0,597	Normal
S99	0,660	-0,222	-0,545	-0,298	0,121	Basal
S100	0,050	-0,135	0,101	-0,167	0,189	Normal
S101	-0,315	0,361	-0,205	0,610	-0,562	LumB
S102	-0,335	0,474	0,005	0,247	-0,348	Her2
S103	-0,109	0,446	-0,301	0,713	-0,628	LumB
S104	-0,530	-0,050	0,325	0,330	-0,165	LumB
S105	-0,290	0,599	0,036	0,093	-0,231	Her2
S106	0,789	-0,085	-0,544	-0,329	0,194	Basal
S107	0,304	-0,291	0,148	-0,685	0,522	Normal
S108	0,012	0,587	-0,163	0,022	-0,173	Her2
S109	0,673	0,027	-0,542	-0,325	0,067	Basal
S110	0,249	-0,332	0,087	-0,762	0,717	Normal
S111	-0,286	0,451	-0,142	0,674	-0,619	LumB
S112	-0,324	0,252	0,120	0,528	-0,320	LumB
S113	-0,120	0,771	-0,281	0,410	-0,508	Her2
S114	-0,467	0,011	0,549	0,011	0,050	LumA
S115	-0,486	0,080	0,183	0,612	-0,388	LumB
S116	0,752	-0,160	-0,591	-0,504	0,217	Basal
S117	0,693	-0,083	-0,458	-0,261	0,027	Basal
S118	-0,524	0,410	0,174	0,753	-0,620	LumB
S119	-0,027	-0,181	-0,090	-0,137	-0,015	Normal
S120	0,023	0,392	-0,274	-0,091	-0,171	Her2
S121	0,779	-0,122	-0,553	-0,217	0,103	Basal
S122	-0,149	0,273	-0,257	0,394	-0,422	LumB
S123	-0,301	0,381	-0,104	0,522	-0,431	LumB
S124	0,580	0,102	-0,658	0,010	-0,248	Basal
S125	-0,456	0,162	-0,001	0,741	-0,549	LumB
S126	-0,311	-0,010	0,321	0,344	-0,187	LumB
S127	-0,221	0,073	0,003	0,443	-0,360	LumB
S128	-0,388	0,120	0,168	0,331	-0,267	LumB
S129	0,123	0,424	-0,103	0,027	-0,030	Her2
S130	-0,287	-0,112	0,234	0,292	-0,244	LumB
S131	0,018	-0,297	0,278	-0,171	0,267	LumA
S132	-0,405	0,099	0,294	0,343	-0,272	LumB
S133	-0,029	0,781	-0,422	0,293	-0,463	Her2
S134	-0,014	0,811	-0,364	0,327	-0,460	Her2
S135	0,770	-0,259	-0,471	-0,397	0,260	Basal
S136	-0,112	0,498	-0,175	0,108	-0,199	Her2
S137	-0,280	0,362	-0,114	0,558	-0,522	LumB
S138	-0,304	0,506	-0,205	0,801	-0,646	LumB

S139	-0,318	0,536	-0,028	0,372	-0,340	Her2
S140	0,673	-0,128	-0,412	-0,202	-0,011	Basal
S141	-0,109	0,757	-0,336	0,530	-0,676	Her2
S142	0,656	-0,059	-0,627	-0,118	-0,108	Basal
S143	-0,304	0,386	-0,110	0,713	-0,568	LumB
S144	-0,356	0,457	-0,092	0,736	-0,585	LumB
S145	-0,122	-0,143	0,193	-0,057	0,075	LumA
S146	0,774	-0,074	-0,621	-0,246	0,031	Basal
S147	0,235	0,465	-0,403	-0,077	-0,199	Her2
S148	0,124	0,256	-0,367	0,116	-0,328	Her2
S149	0,814	-0,285	-0,392	-0,516	0,309	Basal
S150	-0,239	0,405	-0,116	0,618	-0,525	LumB
S151	0,105	-0,069	0,070	-0,504	0,326	Normal
S152	0,842	-0,120	-0,641	-0,228	0,047	Basal
S153	-0,302	0,204	-0,123	0,681	-0,592	LumB
S154	0,212	0,561	-0,490	0,018	-0,308	Her2
S155	-0,360	0,625	-0,149	0,720	-0,650	LumB
S156	0,733	-0,138	-0,407	-0,325	0,166	Basal
S157	-0,063	0,752	-0,243	0,377	-0,472	Her2
S158	0,176	0,683	-0,364	0,130	-0,226	Her2
S159	0,645	-0,039	-0,508	-0,376	0,079	Basal
S160	0,688	-0,264	-0,385	-0,209	0,126	Basal
S161	0,398	0,434	-0,586	0,177	-0,281	Her2
S162	0,415	0,085	-0,604	-0,124	-0,122	Basal
S163	0,533	-0,068	-0,391	-0,506	0,131	Basal
S164	0,261	0,238	-0,146	-0,171	0,045	Basal
S165	0,335	0,069	-0,541	0,035	-0,233	Basal
S166	0,190	0,743	-0,590	0,301	-0,498	Her2
S167	-0,269	0,584	-0,273	0,412	-0,527	Her2
S168	0,777	-0,171	-0,498	-0,122	0,027	Basal
S169	-0,121	0,739	-0,169	0,328	-0,377	Her2
S170	0,647	0,054	-0,704	0,071	-0,230	Basal
S171	0,233	0,227	-0,515	0,175	-0,329	Basal
S172	0,044	0,002	-0,117	0,353	-0,247	LumB
S173	0,814	-0,246	-0,480	-0,437	0,264	Basal
S174	-0,152	0,729	-0,166	0,089	-0,289	Her2
S175	0,375	0,312	-0,641	0,150	-0,320	Basal
S176	-0,232	0,316	-0,076	0,279	-0,314	Her2
S177	0,637	-0,029	-0,705	-0,126	-0,117	Basal
S178	0,165	-0,076	-0,053	0,063	-0,033	Basal
S179	0,798	-0,057	-0,590	-0,496	0,129	Basal
S180	-0,125	0,181	-0,284	0,661	-0,564	LumB
S181	0,643	-0,165	-0,348	-0,491	0,258	Basal
S182	0,833	-0,147	-0,656	-0,268	0,092	Basal
S183	0,903	-0,237	-0,606	-0,220	-0,005	Basal
S184	0,773	-0,057	-0,653	-0,185	-0,052	Basal
S185	0,178	0,366	-0,679	0,478	-0,611	LumB

S186	0,727	0,106	-0,770	-0,067	-0,156	Basal
S187	0,744	-0,284	-0,427	-0,304	0,173	Basal
S188	0,751	0,035	-0,763	-0,075	-0,098	Basal
S189	0,809	-0,091	-0,606	-0,193	0,024	Basal
S190	0,809	-0,091	-0,606	-0,193	0,024	Basal
S191	0,445	0,504	-0,699	0,021	-0,302	Her2
S192	0,832	-0,153	-0,463	-0,487	0,252	Basal
S193	0,894	-0,181	-0,584	-0,441	0,185	Basal
S194	0,759	0,114	-0,724	-0,190	-0,070	Basal
S195	0,863	-0,159	-0,586	-0,311	0,159	Basal
S196	0,481	0,104	-0,616	0,266	-0,446	Basal
S197	0,787	0,063	-0,769	-0,019	-0,154	Basal
S198	0,783	-0,231	-0,451	-0,361	0,219	Basal
S199	0,830	-0,168	-0,558	-0,452	0,172	Basal
S200	0,080	0,390	-0,575	0,707	-0,696	LumB

Anexo 6 Interfaz SVM

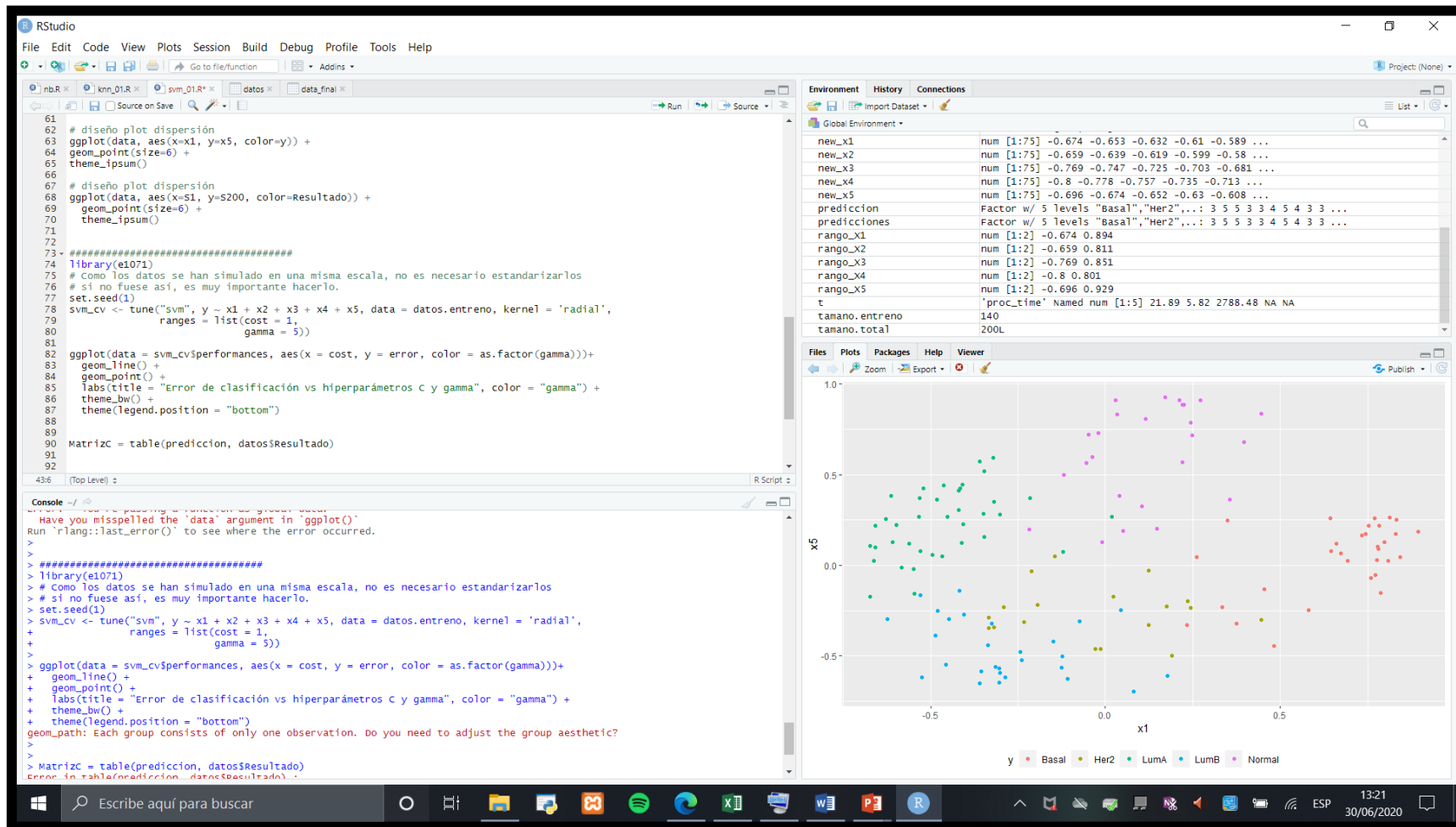


Figura 25 Interfaz SVM - RSTUDIO

Anexo 7 Manual de la implementación SVM

```
t <- proc.time() # INICIA EL CRONOMETRO
#Se inserta la base de datos
library(readxl)
dataset <- read_excel(NULL)
View(dataset)
# Se llama a las siguientes Librerías, sino están instaladas se usa el scrip ins
installed.packages()
library(caret)
library(e1071)
#Identificación de datos, a cada clase se le asigna un nombre en este caso
utilizaremos x1, x2, x3, x4, x5, y y ahora con los nuevos nombres se guardará en
la tabla datos.
datos <- data.frame( x1 = data_final$Basal,
                    x2 = data_final$Her2, x3 = data_final$LumA,
                    x4 = data_final$LumB, x5 = data_final$Normal, y=
data_final$Resultado)

#mostraremos la fila y de la tabla datos. Donde están guardados los resultados
datos$y <- as.factor(datos$y)
head(datos)

# Se selecciona los datos de muestra que será el 70% de los datos y el 30%
serán datos de prueba.
set.seed(200)
tamano.total <- nrow(datos)
tamano.entreno <- round(tamano.total*0.7)
datos.muestra <- sample(1:tamano.total , size=tamano.entreno)
datos.entreno <- datos[datos.muestra,]
datos.test <- datos[-datos.muestra,]

# Ejecución del modelo SVM (Asignamos el nombre modeloSVM, llamaremos a la
función svm() donde insertaremos y que es el resultado, los datos de
entrenamiento, el método de clasificación, el kernel que se usó el Kernel radial y
por ultimo gamma y cost.)
modeloSVM <- svm(y~., data=datos.entreno, method="C-classification",
kernel="radial", gamma=0.1, cost=10)

#Observación del modelo (Se utiliza la función summary() para mostrar el
resumen del modelo)
summary(modeloSVM)

#Se halla la predicción del modelo y de los datos de prueba en la función predict()
prediccion <- predict (modeloSVM, datos.test)
#Se crea la matriz donde insertaremos los resultados de los datos de prueba con
la predicción hallada.
Matriz <- table (datos.test $ y, prediccion)
```

```
#Para mostrar la matriz de confusión solo llamaremos a la función  
confusionMatriz()  
confusionMatriz(Matriz)
```

```
proc.time()-t # FIN DEL CRONÓMETRO
```

Anexo 8 Interfaz KNN

The screenshot displays the RStudio environment with the following components:

- Source Editor:** Contains R code for K-Nearest Neighbors (KNN) implementation, including data loading, normalization, and model training.
- Environment:** Lists objects in the workspace, such as `data_prueba_x` (60 obs. of 1 variable), `data_target_resultado` (180 obs. of 1 variable), and `dataknn` (200 obs. of 6 variables).
- Console:** Shows the execution output, including a P-value of $7.387e-09$, a kappa value of 0.8052, and a detailed confusion matrix for five classes: Basal, Her2, LumA, LumB, and Normal.
- Viewer:** Displays a bar chart showing the distribution of the five classes. The 'Basal' class has the highest count, exceeding 30.

Confusion Matrix Data from Console:

	Class: Basal	Class: Her2	Class: LumA	Class: LumB	Class: Normal
Sensitivity	1.0000	0.7143	1.00000	0.7500	1.00000
Specificity	0.8571	0.9348	1.00000	1.0000	1.00000
Pos Pred Value	0.8889	0.7692	1.00000	1.0000	1.00000
Neg Pred Value	1.0000	0.9149	1.00000	0.9412	1.00000
Prevalence	0.5333	0.2333	0.01667	0.2000	0.01667
Detection Rate	0.5333	0.1667	0.01667	0.1500	0.01667
Detection Prevalence	0.6000	0.2167	0.01667	0.1500	0.01667
Balanced Accuracy	0.9286	0.8245	1.00000	0.8750	1.00000

Figura 26 Interfaz KNN- RSTUDIO

Anexo 9 Manual de la implementación KNN

```
t <- proc.time() # INICIA EL CRONOMETRO
#Inicio guardando los datos en dataknn
dataknn <- data_final
#Utilizamos el comando str para ver si los datos están estructurados
str (dataknn)
head(dataknn)
#Elimina la primera variable
dataknn <- dataknn [-1]
#Obtener el numero de muestras que pertenecen a
#cada subtipo de cancer
table(dataknn $ Resultado)

#NORMALIZAR LOS DATOS NUMERICOS
normalizar <- function(x) { return ((x - min(x)) / (max(x) - min(x))) }

#normalización de todas las variables
data_norm <- as.data.frame(lapply(dataknn[1:5], normalizar))

#verificación de la normalización
summary(data_norm)

#CREACIÓN DE CONUNTO DE DATOS
data_entreno <- data_norm [1:140,]
data_prueba <- data_norm [141:200,]

#incluimos a Resultado
data_entreno_x <- dataknn [1:140, 6]
data_prueba_x <- dataknn [141:200, 6]
summary(dataknn)

#Modelo KNN
##stall.packages("class")
library(class)
cl = data_entreno_x[,1, drop=TRUE]
modeloKNN <- knn(train = data_entreno, test = data_prueba, cl , k=15)

x = matrix(modeloKNN)
#verificación de resultados
plot(modeloKNN)
library(gmodels)
CrossTable(data_prueba_x, x)
y= matrix(data_entreno_x)
Matriz <- table(modeloKNN, y)
#Realizo la matriz de confusión
confusionMatrix(x)

proc.time()-t # FIN DEL CRONÓMETRO
```

Anexo 10 Interfaz NB

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains R code for data sampling, model training, prediction, and confusion matrix generation.
- Environment:** Lists objects in the workspace: data_final (200 obs. of 7 variables), dataNB (200 obs. of 7 variables), modelNB (List of 5), testData (60 obs. of 7 variables), and trainData (140 obs. of 7 variables).
- Console:** Shows the execution output, including a p-value of $< 2.2e-16$, Kappa of 0.8726, and a table of statistics by class.
- Plots:** A confusion matrix plot titled "Matriz" with columns for Basal, Her2, LumA, LumB, and Normal, and rows for Basal and Normal.

```

9 dataNB = data_final
10
11 str(dataNB)
12
13
14 ind <- sample(2,nrow(dataNB), replace = TRUE, prob = c(0.7,0.3) ) #70% entrenamiento y 30% test
15 trainData<- dataNB[ind==1,]
16 testData<- dataNB[ind==2,]
17
18 modelNB = naiveBayes(as.factor(Resultado) ~., data=trainData)
19
20 modelNB
21
22
23 pred = predict(modelNB, testData)
24
25 Matriz <- table(testData$Resultado, pred)
26
27
28 library(caret)
29
30
31 confusionMatrix(Matriz)
32 plot(Matriz)
33
34 proc.time()-t # FIN DEL CRONOMETRO
35
36 #####3
37
38
39
40

```

Console Output:

```

P-value [Acc > NIR] : < 2.2e-16
Kappa : 0.8726
McNemar's Test P-Value : NA
Statistics by Class:
Class: Basal Class: Her2 Class: LumA Class: LumB Class: Normal
Sensitivity      1.0000  0.7273  0.9167  0.9444  0.8750
Specificity      0.9592  1.0000  0.9583  0.9762  0.9808
Pos Pred Value   0.8462  1.0000  0.8462  0.9444  0.8750
Neg Pred Value   1.0000  0.9423  0.9787  0.9762  0.9808
Prevalence       0.1833  0.1833  0.2000  0.3000  0.1333
Detection Rate   0.1833  0.1333  0.1833  0.2833  0.1167
Detection Prevalence 0.2167  0.1333  0.2167  0.3000  0.1333
Balanced Accuracy 0.9796  0.8636  0.9375  0.9603  0.9279

```

Confusion Matrix Plot (Matriz):

	Basal	Her2	LumA	LumB	Normal
Basal	100	0	0	0	0
Normal	0	0	0	0	0

Figura 27 Interfaz NB – RSTUDIO

Anexo 11 Manual de la implementación Naive Bayes

```
t <- proc.time() # INICIA EL CRONOMETRO
#implementación NB

library(e1071)
dataNB = data_final

#Creación del modelonb
str(dataNB)

ind <- sample(2,nrow(dataNB), replace = TRUE, prob = c(0.3,0.7) ) #70%
entrenamiento y 30% test
trainData<- dataNB[ind==1,]
testData<- dataNB[ind==2,]
modeloNB = naiveBayes(as.factor(Resultado) ~., data=trainData)
modeloNB
pred = predict(modeloNB, testData)
Matriz <- table (testData$Resultado, pred)
library(caret)
plot(matriz)
confusionMatrix(Matriz)
proc.time()-t # FIN DEL CRONÓMETRO
```