UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

Selección de Características para la Clasificación de Estrellas y Cuásares Mediante el Cálculo de Testores Típicos

Mateo Martínez Mejía

Física

Trabajo de titulación presentado como requisito

para la obtención del titulo de

Físico

12 de diciembre de 2022

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingeniería

HOJA DE CALIFICACIÓN DE TRABAJO DE FIN DE CARRERA

Mateo Martínez Mejía

Nombre del profesor, Titulo académico: Julio Ibarra, MSc

12 de diciembre de 2022

© Derechos de Autor

Nombres y apellidos:	Mateo Martínez Mejía
Código:	00206438
Cédula de Identidad:	1719153189
Lugar y fecha:	12 de diciembre de 2022

ACLARACIÓN PARA LA PUBLICACIÓN

Nota: El presente trabajo, en su totalidad o cualquiera de sus partes, no debe ser considerado como una publicación, incluso a pesar de estar disponible sin restricciones a través de un repositorio institucional. Esta declaración se alinea con las prácticas y recomendaciones presentadas por el Committee on Publication Ethics COPE descritas por Barbour et al. (2017) Discussion document on best practice for issues around theses publishing, disponible en `http://bit.ly/COPETheses`

UNPUBLISHED DOCUMENT

Note: The following capstone project is available through Universidad San Francisco de Quito USFQ institutional repository. Nonetheless, this project – in whole or in part – should not be considered a publication. This statement follows the recommendations presented by the Committee on Publication Ethics COPE described by Barbour et al. (2017) Discussion document on best practice for issues around theses publishing available on `http://bit.ly/COPETheses`

Agradecimientos

En primer lugar, quisiera agradecer a mi familia, quienes me han dado su apoyo incondicional a lo largo de mi carrera y han sido mi sostén en cada reto y cada decisión. De igual manera, quisiera dar gracias a mis amigos María del Carmen Salazar, José Ochoa, Alejandro Rueda y Ariana Soria quienes me han acompañado a lo largo de este viaje y me han motivado a superarme cada día. Quisiera dar una agradecimiento especial a Milena Mora quien me ha dado su constante soporte y me ha motivado a siempre dar lo mejor. Por último, quisiera dar gracias a mis maestros quienes me han guiado en este viaje y me han mostrado la belleza de la ciencia a través de sus enseñanzas.

Resumen

La teoría de testores presenta grandes ventajas frente otros métodos de reducción de dimensionalidad de bases de datos como el análisis de componentes principales (PCA). A partir de un subconjunto aleatorio de objetos de la base *Stellar Classification Dataset - SDSS17*, se encontró los testores típicos y se realizó una comparación de eficiencia con el PCA a través de un modelo de clasificación SVM. Los resultados de precisión obtenidos de cada método, mostraron la idoneidad de la teoría de testores para reducir el número de características de una base de datos sin afectar su capacidad de entrenar un modelo de clasificación.

Palabras clave: *Reducción de dimensionalidad, teoría testores, PCA, modelo de clasificación, SVM, precisión de entrenamiento.*

Abstract

Testor theory offers great advantages in contrast to other dimensionality reduction methods such as the principal component analysis (PCA). A subset of objects from the *Stellar Classification Dataset - SDSS17* was randomly chosen and the typical testors associated with these objects where found. Furthermore, an efficiency comparison was made between the typical testor and PCA through a SVM classification model. The precision results obtained from each method, showed the suitability of testor theory to reduce the number of characteristics in a dataset, without negatively affecting its capacity to train a classification model.

Keywords: *Dimensionality reduction, testor theory, PCA, classificaction model, SVM, training precision.*

Índice general

Índice de cuadros

Índice de figuras

Capítulo 1

Introducción

En la actualidad, el mundo se mueve a través de los datos, su adquisición, su almacenamiento y, finalmente, su uso. Las bases de datos son esenciales para el desarrollo tecnológico de nuestra sociedad y el uso óptimo y apropiado de esta información es indispensable para lograr este objetivo. En principio, se puede considerar que mientras más información se tenga dentro de una base de datos, mejor será su aplicación en ámbitos como los modelos de clasificación. Si bien esto es cierto, también es importante eliminar la información innecesaria que no tiene un aporte real a los objetivos planteados. Dentro de este trabajo de investigación, nos centraremos principalmente en la búsqueda de esta información "esencial" y analizar la manera en la que esta información reducida se comporta dentro de un modelo de clasificación.

Dentro del análisis de datos, se utiliza el término *reducción de dimensionalidad* de la base de datos para referirse a esta búsqueda de la información más impor-

tante dentro de la misma. En la actualidad, existen algunos métodos que permiten realizar esta tarea y generalmente se basan en procesos estadísticos para poder capturar la información. Sin embargo, el modelo lógico combinatorio conocido como *Teoría de Testores*, presenta una alternativa que ofrece grandes ventajas frente a otros modelos estadísticos. Dichas ventajas serán analizadas a lo largo de este trabajo.

La Teoría de Testores fue desarrollada en la década de 1960 con un enfoque distinto al que va a ser explorado. Inicialmente, esta teoría fue desarrollada con el propósito de detectar fallas en circuitos electrónicos cuya utilidad era la ejecución de funciones booleanas. Sin embargo, esta progresó hacia la selección de variables en distintos problemas y la clasificación supervisada [2, 3]. La utilidad que se le dará a los testores típicos, justamente va de la mano con la selección de subconjuntos de variables que permitan realizar una clasificación supervisada de una base de datos.

Dentro de este trabajo de investigación se presentará una descripción de los métodos utilizados, presentando los conceptos básicos de la Teoría de Testores, el modelo clasificación y un método alterno de reducción de dimensionalidad muy conocido en la actualidad. Posteriormente, se hará uso de ambos procesos para el entrenamiento del modelo de clasificación y se presentará la eficiencia de cada uno en contraste con el entrenamiento realizado sin emplear ninguno de los dos. Para finalizar, se evaluará los resultados obtenidos y se realizará una comparación de ambos métodos para determinar su desempeño.

Capítulo 2

Métodos

Una base de datos de clasificación viene definido por objetos de distintas clases que vienen definidas por un conjunto de características. Estas características permiten distinguir los objetos entre clases dependiendo de los valores de las mismas. De esta manera, podemos realizar un análisis que determine cuáles combinaciones de estas características son suficientes para poder diferenciar entre objetos de diferentes clases. En otras palabras, podemos determinar cuáles son las características más relevantes para diferenciar entre distintas clases de objetos. Este análisis nos dirige a la idea de los testores típicos.

La reducción de características de una base de datos puede ser considerada como una reducción de dimensionalidad de la misma. Dentro del análisis de datos, existe un método de reducción de dimensionalidad conocido como Análisis de Componentes Principales, o PCA por sus siglas en inglés. Este es un proceso estadístico que permite reducir el número de características, pero con resultados

diferentes a los obtenidos a partir de los testores típicos.

Una vez la base de datos ha sido procesada, existen dos maneras de evaluar la eficiencia de los métodos de reducción de dimensionalidad.

La reducción de dimensionalidad de una base de datos debe ser evaluada a través de la eficiencia de la nueva base generada, es decir, que tan adecuada es la reducción de características de manera que sean capaces de identificar los elementos de cada clase. Para poder realizar esta evaluación, se hará uso de un modelo de clasificación que permita predecir la clase a la cuál pertenece un objeto que cuenta tan solo con las características reducidas. En este trabajo, se hará uso del modelo de *Support Vector Machine* (SVM), el cuál es un modelo de aprendizaje supervisado, para realizar esta predicción. Para lograr esta clasificación, se separa los objetos en dos subconjuntos, uno que almacena datos de prueba, y otro que almacena datos de entrenamiento. Los datos de entrenamiento, tal como menciona el nombre, permiten entrenar el modelo de tal manera que al ingresar un nuevo objeto, este pueda determinar a qué clase pertenece. Por otro lado, el resto de datos, permiten evaluar la precisión del modelo. Esta evaluación de la precisión del modelo, será lo que nos permita determinar cuál de los métodos de reducción de dimensionalidad tuvo una mejor efectividad.

2.1. Teoría de Testores

Para comprender la teoría de testores y el concepto que permite realizar una reducción de dimensionalidad, es necesario presentar algunas definiciones impor-

tantes. Las definiciones que se presentarán en cuanto a los testores típicos serán generalizadas para una base de datos de n características y r clases distintas.

Definimos U como un conjunto de objetos de n características y r clases a las cuales pueden pertenecer [3, 4]. Es importante mencionar que las r clases son disjuntas, es decir, si $c_i = \{u \in U : u \textit{ es parte de la clase } i\}$, entonces $c_i \cap c_j = \varnothing \forall i, j \in \{1, 2, \ldots, r\}$.

Definición 1. Sea $|c_i| = p_i$, entonces, el número de parejas de clases distintas será

$$N = \sum_{k=1}^{r} \sum_{i=1}^{k} p_k p_i - \sum_{i=1}^{r} p_i^2 \tag{2.1}$$

A partir de los elementos U, definimos la *matriz de disimilaridad* M de la siguiente manera.

Definición 2. Sea $u \in c_k$, $v \in c_l$ donde $k \neq l$ y sea (u, v) la pareja i de N posibles, entonces definimos las matriz de disimilaridad M

$$M = \{m_{ij}\}_{N \times n} \quad m_{ij} \in \{1, 0\} \tag{2.2}$$

donde $m_{ij} = 0$ si los objetos u y v son similares en la característica j y $m_{ij} = 1$ si los objetos son diferentes en la característica j [4].

Podemos tomar en cuenta que esta matriz M nos encontraremos con mucha

información redundante. Por esta razón, se busca poder simplificar esta matriz booleana de modo que se utilice únicamente las filas que presentan la información esencial.

Definición 3. Sean f y g filas de la matriz M. Decimos que $f < g$ si $\forall j \in \{1, 2, \ldots, n\},\ f_j < g_j$. Además, si $\nexists\ h$ fila de M tal que $h < f$, entonces decimos que f es una fila básica de M [3, 4].

Definición 4. Sea M_B la matriz que solo contiene todas las filas básicas de M, entonces M_B es la matriz básica de M [4].

Una vez obtenida esta matriz que ya no presenta información redundante de las comparaciones de objetos, podemos definir la idea de testor.

Definición 5. Sea J el conjunto de características de los objetos de U. $T \subseteq J$ es un testor si y sólo no existe un par de objetos en U que sean similares en todas las características de T [5, 6]. En términos de la matriz M, la matriz restringida a T, $M_{|T}$, no contiene filas de ceros.

Definición 6. Si $\nexists\ T_2 \subset T$ tal que T_2 también sea testor, entonces decimos que T es testor típico [5, 6],.

Ahora, consideramos que esta definición de testor típico toma en cuenta la matriz de disimilaridad M, sin embargo, buscamos trabajar con la matriz M_B. Para esto, utilizamos la siguiente proposición.

Proposición 1. Sea $\Psi^*(M)$ el conjunto de testores típicos de M, entonces se

cumple que

$$\Psi^*(M) = \Psi^*(M_B) \tag{2.3}$$

En otras palabra, el conjunto de testores típicos de la matriz de disimilaridad es el mismo que el de la matriz básica [4].

Una vez definidas la ideas de los testores típicos, se puede observar que el subconjunto de características que encuentra un testor típico, es el mínimo conjunto que permite diferenciar elementos de clases distintas.

Ya que tenemos una idea preliminar acerca de la teoría de testores. Podemos analizar uno de los algoritmos más importantes de búsqueda de testores, conocido como *Yablonski & Compatible Sets* o *YYC*.

2.1.1. Algoritmo YYC

El algoritmo YYC, es un algoritmo recursivo que toma un enfoque distinto al de otros métodos de cálculo. En lugar de buscar los testores típicos de toda la matriz básica, este algoritmo busca los testores típicos hasta la i-ésima fila de la matriz en la iteración i. En otras palabra, en cada iteración se actualiza el conjunto de testores típicos para adición de una nueva fila. La manera en la que este método funciona es a partir de la validación de compatibilidad entre un elemento del conjunto de testores típicos hasta la fila i y un elemento del conjunto

de testores típicos de la fila $i+1$. Esto quiere decir, que el algoritmo añade a cada testor típico que ha encontrado una característica más, dependiendo de la posición de los valores 1 en la fila $i+1$, y comprueba que no se pierda la propiedad de tipicidad [7].

2.2. Principal Component Analysis (PCA)

El Análisis de Componentes Principales o PCA por su siglas en inglés es un método estadístico multivariable de reducción de dimensionalidad de bases de datos. Este método se encarga de realizar una transformación de los datos con nuevas variables definidas, o componentes principales, que dentro de un menor número sean capaces de retener la mayor cantidad de variación de los objetos originales [8, 9].

Al igual que en el caso de los testores típicos el método PCA, presenta una base matemática que se basa en la descomposición de una matriz a través de sus valores y vectores propios [9]. A continuación se mostrará el tratamiento matemático de los datos que permiten hallar los componentes principales de una base de datos. De manera general consideramos una base de datos con n características.

Definición 7. Sea $\mathbf{p} = \{p_1, p_2, \ldots, p_n\}$ el conjunto de variables de la base de datos. Decimos que $\mathbf{\Sigma}$ es la matriz de covarianza si tenemos que

$$\Sigma_{ij} = cov(p_i, p_j) \tag{2.4}$$

Donde $cov(p_i, p_j)$ es la covarianza entre las características p_i y p_j.

Definición 8. Sea $\mathbf{x}$ un objeto de la base de datos, decimos que z_k es el k-ésimo componente principal (PC) de $\mathbf{x}$ si

$$z_k = \boldsymbol{\alpha}_k^T \mathbf{x} = \sum_{i=1}^{n} \alpha_{ki} x_i \tag{2.5}$$

Donde $\boldsymbol{\alpha}_k = [\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{kn}]$ es el autovector de $\boldsymbol{\Sigma}$ asociado al k-ésimo autovalor, λ_k, más grande [8].

Es importante mencionar que el autovector $\boldsymbol{\alpha}_k$ está normalizado. Además de esto, consideramos que existe una relación entre los autovalores de $\boldsymbol{\Sigma}$ y las varianzas de los PCs calculados.

Proposición 2. Sea $\boldsymbol{\alpha}_k^T \mathbf{x}$ el k-ésimo PC de $\mathbf{x}$ y sea λ_k el k-ésimo autovalor más grande de $\boldsymbol{\Sigma}$, entonces

$$var(\boldsymbol{\alpha}_k^T \mathbf{x}) = \lambda_k \tag{2.6}$$

Donde $var(\boldsymbol{\alpha}_k^T \mathbf{x})$ es la varianza del PC [8].

Este método estadístico ha tenido varias aplicaciones en cuanto a la extracción de características y una de las áreas donde más ha tenido aplicación es en el análisis tráfico[10]. Sin embargo, el PCA presenta algunas limitaciones que serán exploradas a lo largo de este trabajo de aplicación en comparación con la teoría

de testores.

2.3. Support Vector Machines (SVM)

El modelo de Máquinas de Vector de Soporte, o SVM por sus siglas en inglés, permite realizar clasificaciones de datos tal como los modelos de redes neuronales artificiales y los KNNs. Al igual que los ejemplos mencionados, los SVMs tienen una estructura matemática que permite realizar las clasificaciones de una manera optimizada. Dentro de este trabajo de investigación se hablará únicamente del modelo SVM para dos clases.

El objetivo del modelo SVM es ser capaz de hallar un hiper-plano en un espacio de alta dimensionalidad que pueda ser considerado "bueno" en términos de una métrica de rendimiento generalizado [11]. Para definir este modelo, suponemos que tenemos un conjunto de l objetos caracterizados por n parámetros. Definimos $U = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$ como el conjunto de objetos de la base de datos, donde $\mathbf{x}_i \in \mathbb{R}^n$ y $y_i \in \{-1, 1\}$[11]. En otras palabras, $\mathbf{x}_i$ es el conjunto de características que representan al objeto y y_i representa la clase a la cual pertenece el objeto.

Definición 9. Sea $\mathbf{w}$ un vector n-dimensional y b un escalar, entonces definimos el hiper-plano

$$\mathbf{x} \cdot \mathbf{w} + b = 0 \tag{2.7}$$

donde $\mathbf{w}$ es perpendicular al hiper-plano. Este hiper-plano, realiza la separación de los objetos de ambas clases [12].

A partir de este hiper-plano inicial, definimos dos nuevos hiper-planos paralelos que no contengan elementos de ninguna de las dos clases. Estos hiper-planos serán los siguientes.

$$\mathbf{x} \cdot \mathbf{w} + b = 1 \tag{2.8}$$

$$\mathbf{x} \cdot \mathbf{w} + b = -1 \tag{2.9}$$

Proposición 3. Sea γ la distancia que separa los hiper-planos paralelos, entonces tenemos la siguiente ecuación.

$$\gamma = \frac{2}{||\mathbf{w}||} \tag{2.10}$$

Por lo tanto, el objetivo del método es la minimización de $||\mathbf{w}||$ [11].

Definición 10. El problema de optimización es la minimización de $||\mathbf{w}||^2$ bajo la restricción

$$y_i[(\mathbf{x}_i \cdot \mathbf{w}) + b] - 1 \geq 0 \quad \forall i \in \{1, \ldots, l\} \tag{2.11}$$

donde $\mathbf{x}_i$ es el i-ésimo objeto de la base y y_i es su clase respectiva [11].

Finalmente, se busca la resolución del problema a partir de la formulación lagrangiana del mismo.

$$L(\mathbf{w}, \boldsymbol{\alpha}, b) = \frac{\mathbf{w}^T\mathbf{w}}{2} - \sum_{i=1}^{l} \alpha_i[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1] \tag{2.12}$$

Donde $\alpha_i \geq 0 \ \forall i \in \{1, \ldots, l\}$ son los multiplicadores de Lagrange [11].

Para justificar la convexidad del problema, consideramos que tenemos la minimización de una función objetivo cuadrática bajo restricciones puramente lineales. Por lo tanto, este problema puede ser resuelto a partir del problema de maximización equivalente. Para hallar este problema dual, es necesario imponer las siguientes condiciones de optimalidad.

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} y_i\alpha_i\mathbf{x}_i = 0 \tag{2.13}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l} y_i\alpha_i = 0 \tag{2.14}$$

A partir de la ecuación 2.13 podemos obtener una expresión para $\mathbf{w}$ y la podemos reemplazar en la ecuación 2.12 para obtener la función objetivo del problema dual [11]. Sea $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_l\}$, entonces

$$max(F(\boldsymbol{\alpha})) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{2.15}$$

Esta función objetivo, se ve sujeto a la restricción presentada en la ecuación 2.14 y nuevamente considerando que $\alpha_i \geq 0 \ \forall i \in \{1, \ldots, l\}$.

A partir de la resolución del problema dual, se obtienen los valores de los multiplicadores de Lagrange, α_i. A partir de este resultado, se puede definir los vectores de soporte que dan nombre al modelo.

Definición 11. Sean $\mathbf{x}_i$, $\mathbf{x}_j$ dos objetos de clases distintas. Si α_i y α_j son distintos de 0, entonces decimos que $\mathbf{x}_i$ y $\mathbf{x}_j$ son vectores de soporte que generan los hiper-planos de separación de clases [11].

La precisión de este modelo de clasificación aumenta con mayor cantidad de datos ya que las restricciones del modelo de optimización aumenta. De esta manera, se hallarán los hiper-planos necesarios para realizar la clasificación de nuevos objetos ingresados.

2.4. Base de Datos - Stellar Classification

Para este trabajo de investigación, se trabajó con la base de datos Stellar Classification Dataset - SDSS17 de la plataforma Kaggle [1]. Antes de realizar el proceso de búsqueda de testores típicos se realizó algunos cambios y transformaciones para que se facilite la misma.

En primer lugar, se considera que la base de datos contaba con 3 clases: "GALAXY" (galaxia), "STAR" (estrella) y "QSO" (cuásar), pero solo se trabajó con la últimas 2. Esta decisión fue tomado al considerar que la clase "GALAXY" tenía una cantidad considerablemente mayor de elementos que las otras 2 clases que estaban bastante equilibradas entre si. De igual manera, al considerar la necesidad de producir una matriz de disimilaridad para los datos, el tener 3 clases aumentaría en gran manera el número de filas de la misma. Se debe considerar que el tamaño de las matrices que se deben obtener es de importancia para el problema puesto que se debe tomar en cuenta el poder computacional limitado que se tiene para ejecutar los algoritmos relacionados a la teoría de testores.

De la misma manera, dentro de la base de datos existen 17 características que representan a cada uno de los objetos, sin embargo, algunas de estas no contienen información provechosa en cuanto a la identificación de las clases. Las características, y su descripción, que se mantuvieron en la base son las siguientes.

Como se puede observar en la tabla 2.1, 8 de las 11 características son de tipo $float64$, por lo tanto las comparaciones que se hagan entre filas serán muy ambiguas. En otras palabras, es muy probable que al hacer las comparaciones obtengamos un valor 1 las 8 características de este tipo y esto afectaría el cálculo de los testores típicos. Debido a esto, se tomó un enfoque de discretización de la base de datos.

Para poder generar la matriz de disimilaridad, se realizó el siguiente procedimiento. En primer lugar, se realizó un histograma que generó una partición de 10 intervalos uniformes tomando como límites el valor más bajo y más alto de los

Característica	**Descripción**	**Tipo**
alpha	Ángulo de ascensión recta	$float64$
delta	Ángulo de declinación	$float64$
u	Filtro ultravioleta en el sistema fotométrico	$float64$
g	Filtro verde en el sistema fotométrico	$float64$
r	Filtro rojo en el sistema fotométrico	$float64$
i	Filtro infrarrojo cercano en el sistema fotométrico	$float64$
z	Filtro infrarrojo en el sistema fotométrico	$float64$
field_ID	Número de identificación de campo	int
redshift	Valor del corrimiento al rojo basado en el incremento de longitud de onda	$float64$
plate	Identificación de la placa en el SDSS	int
fiber_ID	Identificación de la fibra que apuntó la luz al plano focal	int

Cuadro 2.1: Características de la base de datos [1]

objetos en dicha característica. Para evitar casos en los que tengamos objetos que presenten valores que causen un distorsión en la partición, se analizó los histogramas para que ninguno de los intervalos de la partición se mantenga vacío. Este caso particular ocurrió una única vez al realizar la partición de la característica "u", donde había un único objeto que causaba esta distorsión. Para corregirlo, no se tomó en cuenta este objeto al momento de realizar la discretización. Una vez realizadas las particiones de cada característica, se realizó una transformación de la base de datos donde cada objeto, en lugar de tener un valor en las características de tipo $float64$, tenía el índice de la partición a la cual pertenecía el valor. De esta manera, obtuvimos una base de datos discretizada.

Una vez realizado este proceso, se pudo realizar el cálculo de la matriz de disimilaridad y posteriormente de los testores típicos.

2.5. Cálculo de Testores Típicos

Habiendo definido los testores típicos y sus propiedades, además del tratamiento de la base de datos, se realizó un proceso especial para el cálculo de testores típicos. Como se mencionó anteriormente, se realizó una discretización de las características que eran de tipo $float64$ para poder producir la matriz de disimilaridad. Recordemos que para producir esta matriz, es necesario hacer una comparación de cada objeto de una clase contra cada objeto del resto de clases. En el caso de la base de datos editada, tenemos únicamente 2 clases. Tenemos 18961 objetos de la clase "QSO" y 21543 objetos de las clase "STAR", por lo que la matriz de disimilaridad que deberíamos obtener sería de 408476823×11. La reducción de una matriz de este tamaño, tomaría una gran cantidad de tiempo llevar a cabo debido a que para encontrar filas comparables, sería necesario compara cada fila con el resto. Por esta razón, se tomó un enfoque distinto.

2.5.1. Matriz Básica

Para poder realizar la reducción de la matriz de disimilaridad en un tiempo razonable, se llevó a cabo el siguiente proceso. En primer lugar, se separó los objetos de clases diferentes, se realizó una permutación aleatoria de cada una y se guardó los primeros 3000 datos en grupos de 1000. Posteriormente, se realizó las matrices de disimilaridad con cada uno de los grupos de 1000, es decir, se hizo la comparación entre el primer grupo de cada clase, luego la comparación con el segundo grupo de cada clase y finalmente la comparación con el tercer grupo de

cada clase. De esta manera, se obtuvo 3 matrices de 1000000×11. Luego de hacer la reducción de cada una de estas matrices, se unió a las mismas y se repitió la reducción de la misma hasta hacerla básica.

2.5.2. Cálculo

Una vez obtenida esta matriz básica, el cálculo de los testores típicos es más eficiente, puesto que se reduce el número de filas que se está analizando. Para realizar el cálculo de los testores típicos que se buscaban, aplicamos el algoritmo YYC.

Habiendo definido los elementos necesarios para la búsqueda de testores típicos, el procedimiento de tratamiento de datos y el método de evaluación de rendimiento de los modelos de reducción de dimensionalidad, se puede presentar los resultados obtenidos y discutir los mismos en el contexto de la teoría de testores y sus aplicaciones.

Capítulo 3

Resultados

Como se mencionó en la sección anterior, se hizo uso de una submuestra de los datos de la base. A partir de estas 3 matrices de disimilaridad obtenidas, buscamos los testores típicos de estos datos. Si bien estos subconjuntos de características no representan la información de toda la base datos, serán una muestra del poder de proyección de los mismos frente a nuevos datos. A lo largo de esta sección se mostrará los resultados en cuanto a la reducción de las matrices disimilaridad, los testores típicos y el rendimiento del entrenamiento del modelo de clasificación con todas las características, el testor típico elegido y el resultado del PCA con el mismo número de variable que el testor típico. El rendimiento del modelo de clasificación será expuesto a través de matrices de confusión que muestre la cantidad y porcentaje de datos de prueba que fueron bien clasificados por el mismo. Esto nos dará la base para poder contrastar la eficiencia de cada método de reducción de dimensionalidad contra el uso de todas las características.

3.1. Matriz Básica y Testores Típicos

En la siguiente tabla se muestra los resultados obtenidos de la reducción a matriz básica de cada una de las matrices de disimilaridad que se realizaron.

Grupo	**Número de Filas**
1	14
2	11
3	17
Final	9

Cuadro 3.1: Número de filas de matrices básicas resultantes

Como se puede observar a partir de la tabla 3.1, las matrices resultantes se redujeron en gran manera. Este hecho facilita en gran manera el cálculo de los testores típicos de la matriz básica final. Al tener una matriz tan reducida, el tiempo que le toma al algoritmo YYC calcular todos los testores típicos de la matriz se reduce a un valor inferior a 1 segundo. Al aplicar el algoritmo en cuestión, se obtuvo como resultado un total de 19 testores típicos cuya longitud variaba entre 2 y 4. A partir de estos resultados, se eligió un testor típico de longitud 4 que contenía las siguientes características.

	Característica
1	r
2	field_id
3	redshifit
4	plate

Cuadro 3.2: Subconjunto de características del testor típico

Una vez encontrado el testor típico, se puede iniciar con el entrenamiento del modelo de clasificación SVM para contrastar la eficiencia de cada uno de los métodos de reducción de dimensionalidad.

Para entrenar al modelo SVM, separamos el conjunto de datos en 2 subconjuntos, el de entrenamiento y el de prueba. Para este trabajo, el porcentaje de datos de prueba fue el 33 % de la muestra, el resto de los datos fueron destinados al entrenamiento del modelo de clasificación.

3.2. Entrenamiento del modelo SVM con todas las características

Iniciamos la prueba de eficiencia entrenando el modelos SVM con todas las características de la base de datos, es decir, usando 11 características. Iniciamos utilizando los mismos 6000 datos que permitieron hallar el testor típico.

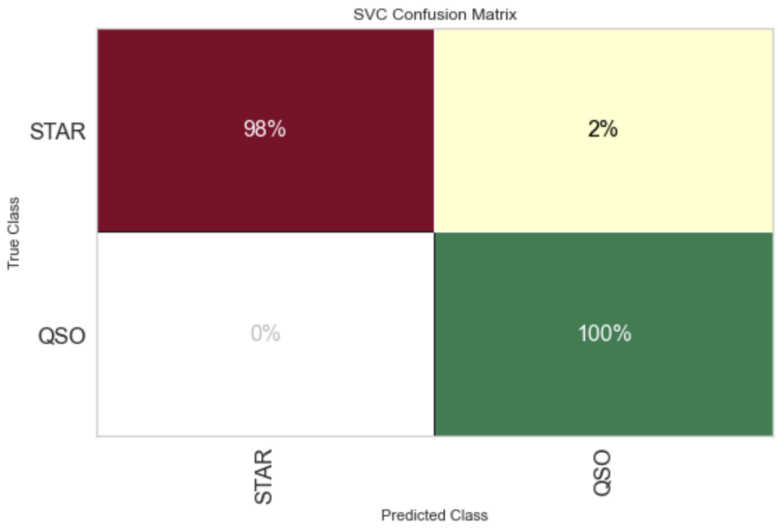

Figura 3.1: Matriz de confusión 11 características - 6000 datos (%)

Como se puede observar en la figura 3.1, tan solo el 2 % de los elementos de la

clase “STAR” fueron clasificados de manera incorrecta. Por otro lado, dentro de la clase “QSO” se no hubo errores en la clasificación.

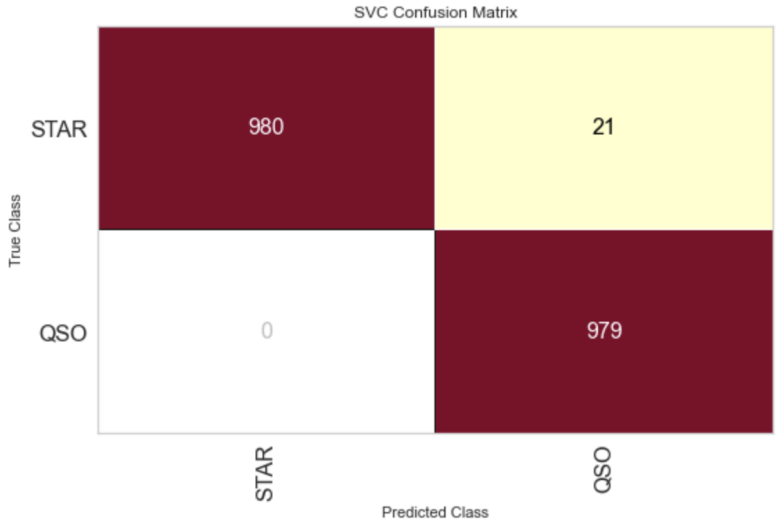

Figura 3.2: Matriz de confusión 11 características - 6000 datos

La figura 3.2 nos muestra la cantidad de elementos que fueron mal clasificados en cuanto a cada clase y podemos observar que de los 1980 datos de prueba, tan solo 21 de ellos no fueron correctamente reconocidos por el modelo.

Una vez realizada la prueba con los 6000 datos, se realizó el entrenamiento del modelo pero esta vez separando subconjuntos de prueba y entrenamiento a los 40504 datos que se tenían en la base de datos. El porcentaje de elementos de prueba fue el mismo que el utilizado para caso anterior.

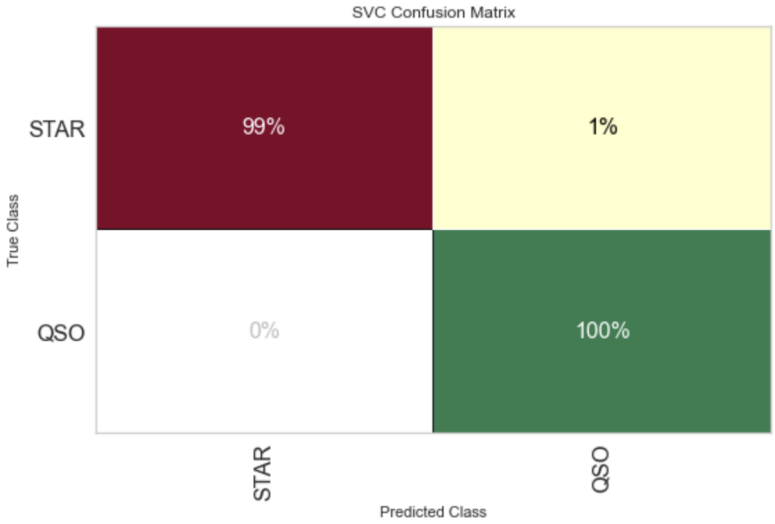

Figura 3.3: Matriz de confusión 11 características - 40554 datos (%)

En la figura 3.3, se puede observar un comportamiento similar que para el caso anterior. La clasificación fue excelente en términos del porcentaje de elementos correctamente clasificados. Este resultado era esperado puesto que un modelo tendrá un mejor rendimiento mientras más datos lo puedan entrenar.

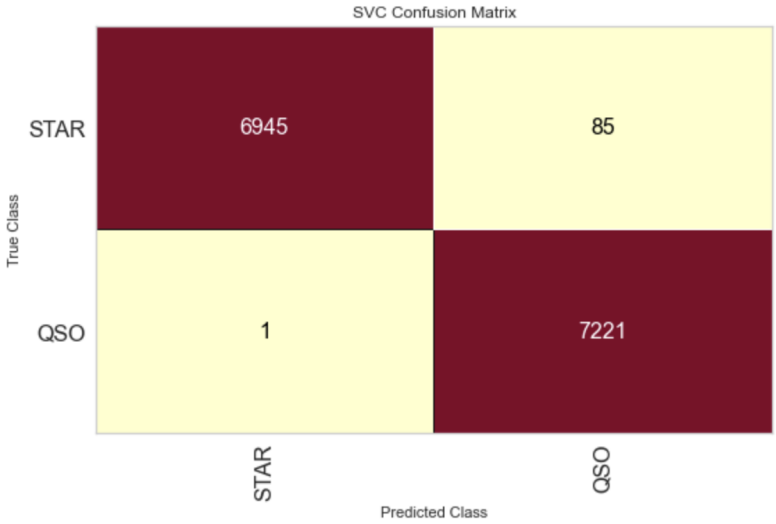

Figura 3.4: Matriz de confusión 11 características - 40554 datos

La matriz de confusión presentada en la figura 3.4 muestra el buen desempeño del entrenamiento del SVM haciendo uso de la base de datos completa. Tan solo 86 de los 14252 elementos, no fueron ubicados en la clase correcta.

Estos resultados muestran que la clasificación de datos haciendo uso de todas las características, presenta un buen rendimiento. Ahora, es momento de realizar una comparación de estos resultados con los métodos de reducción de dimensionalidad mencionados anteriormente.

3.3. Entrenamiento del modelo SVM con el testor típico

El primer método que se utilizó para reducir el número de características fue la teoría de testores con un testor típico de longitud 4.

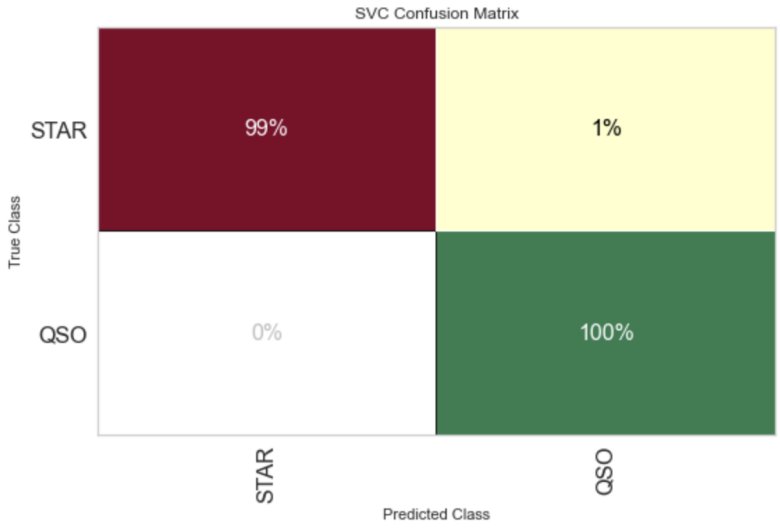

Figura 3.5: Matriz de confusión testor típico - 6000 datos (%)

En esta primera prueba de entrenamiento con el testor típico mostrada en la figura 3.5, se puede observar la facultad de la teoría de testores de lograr una clasificación óptima de los elementos de la base de datos. Sin embargo, este resultado solo muestra el resultado obtenido a partir de los 6000 datos que fueron empleados para hallar el testor típico, es decir, era el resultado esperado a partir de la definición de la teoría de testores.

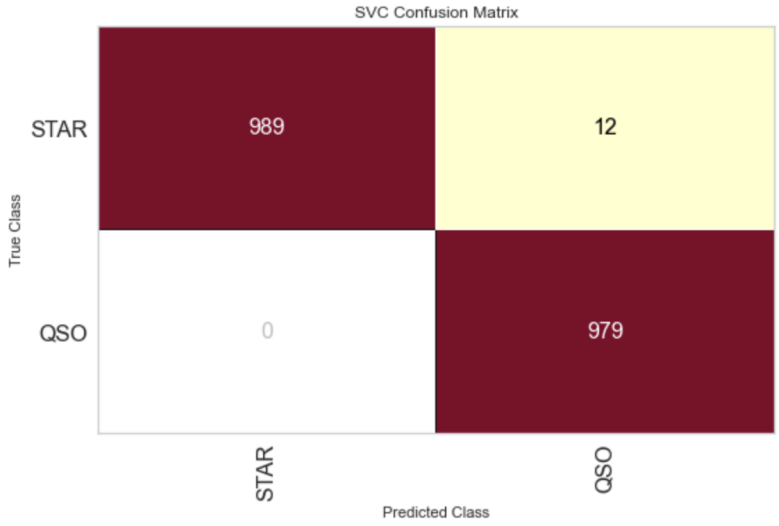

Figura 3.6: Matriz de confusión testor típico - 6000 datos

La figura 3.6 nos muestra que a diferencia del entrenamiento hecho con todas las características, el entrenamiento realizado con el testor típico tan solo clasificó mal 12 elementos del conjunto de prueba. Ya que se ha visto el uso del testor típico dentro de un conjunto de datos a partir de los cuales se obtuvo el mismo, se busca determinar si su valor se mantiene antes nuevo datos que no han sido analizados previamente a través de la teoría de testores.

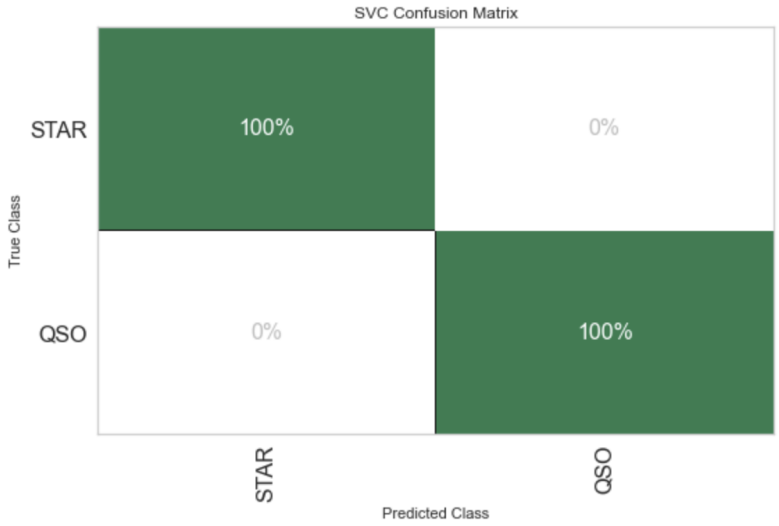

Figura 3.7: Matriz de confusión testor típico - 40554 datos (%)

El resultado obtenido en cuanto al entrenamiento del modelo de clasificación en la figura 3.7 muestra que la proyección de los testores típicos frente a nuevos datos es óptima. Se debe mencionar que la matriz realiza un redondeo para valores superiores a 99.5 %.

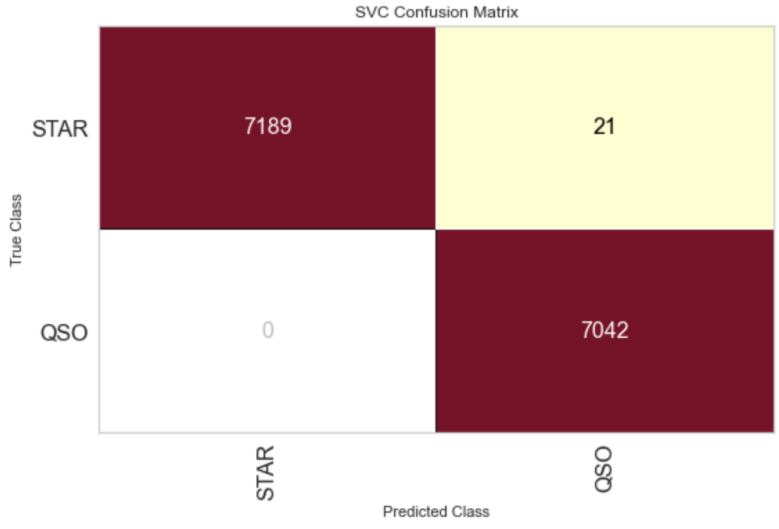

Figura 3.8: Matriz de confusión testor típico - 40554 datos

La figura 3.8 permite observar la eficiencia de los testores típicos en comparación con el uso de todas las características para el entrenamiento del modelo de clasificación. En este caso, solo 21 de los elementos del conjunto de prueba fueron mal clasificados en comparación con los 86 del caso anterior.

3.4. Entrenamiento del modelo SVM con PCA

Como se mencionó en el capítulo 2, el PCA es un método estadístico que realiza una transformación total de la base de datos. Por esta razón, difiere de la manera en la que se reduce las características a través de los testores típicos. Para usar este método, se lo tuvo que aplicar tanto para los 6000 datos como para la base de datos completa.

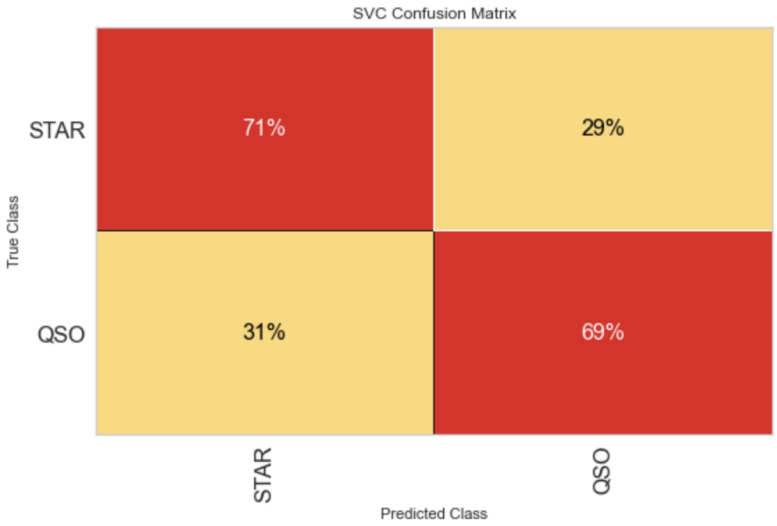

Figura 3.9: Matriz de confusión PCA - 6000 datos (%)

Para los 6000 datos, la figura 3.9 muestra que la clasificación de elementos

presenta muchos errores ya que apenas llega a clasificar alrededor del 70 % de cada clase. Tomando en cuenta los casos anteriores, este método de reducción de dimensionalidad no muestra una clasificación aceptable de los datos.

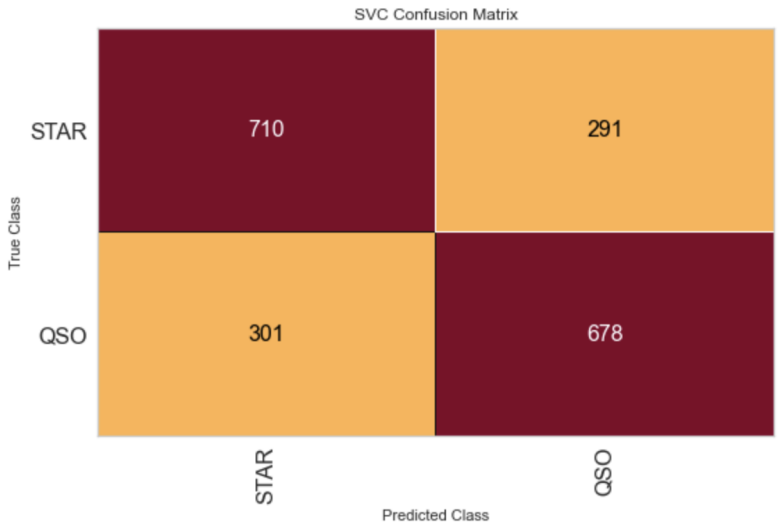

Figura 3.10: Matriz de confusión PCA - 6000 datos

En la figura 3.10 podemos ver que, en total, casi 600 de los 1980 datos de prueba han sido mal clasificados con este entrenamiento del modelo.

Para realizar el entrenamiento con los 40554 elementos de la base, volvemos a realizar la transformación estadística de los datos . Esta transformación será diferente a la realizada para los 6000 datos, puesto que se tendrá una matriz de covarianza distinta. En otras palabra, no es posible proyectar la reducción de características para nuevos datos obtenidos.

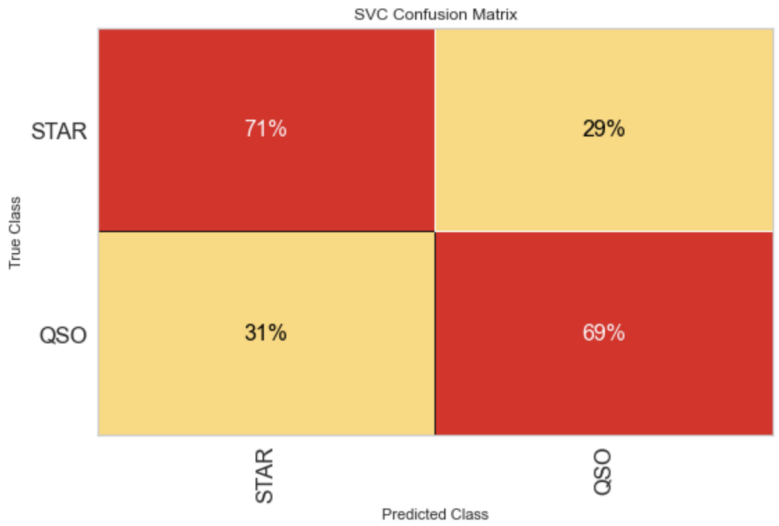

Figura 3.11: Matriz de confusión PCA - 40554 datos (%)

En términos del porcentaje de elementos clasificados, la figura 3.11 muestra que la clasificación fue la misma que con los 6000 datos. Nuevamente, el método del PCA no permite generar un modelo adecuado para realizar la identificación de los elementos de cada clase.

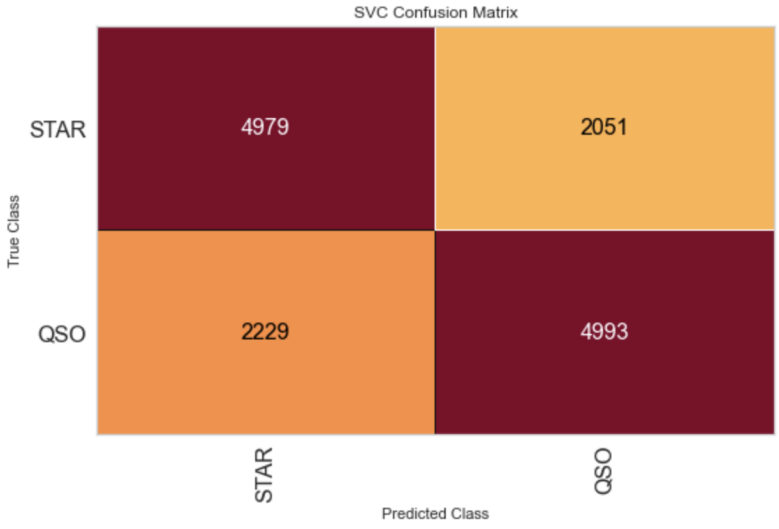

Figura 3.12: Matriz de confusión PCA - 40554 datos

De la misma manera, la figura 3.12 muestra que aproximadamente 4300 elementos de la clase de prueba fueron mal clasificados. Esto, a su vez, es una clara muestra de que el modelo de clasificación no es confiable al hacer uso de esta base de datos transformada y reducida a través del método PCA.

3.5. Precisión de Entrenamiento

No. Características	**Método de Reducción**	**Precisión**
11	Ninguno	98.94 %
4	Testor Típico	99.39 %
4	PCA	70.10 %

Cuadro 3.3: Precisión de entrenamiento con 6000 datos

No. Características	**Método de Reducción**	**Precisión**
11	Ninguno	99.40 %
4	Testor Típico	99.85 %
4	PCA	69.97 %

Cuadro 3.4: Precisión de entrenamiento con 40554 datos

Las tablas 3.3 y 3.4 muestran la precisión total del modelo en cuanto al porcentaje de los datos bien clasificados. Claramente, se destaca la precisión obtenida por parte de los testores típicos en comparación con el uso de todas las características. Se puede observar que tanto para los 6000 datos como para la base de

datos completa, el método de la teoría de testores supera en porcentaje de elementos correctamente clasificados incluso al entrenamiento realizado con todas las características.

Dados estos resultados acerca de la precisión del modelo de clasificación SVM, podemos analizar las ventajas que presenta la teoría de testores como un método de reducción de dimensionalidad. De igual manera, podemos contrastar su utilidad frente a los resultados que se obtuvo del método PCA y ampliar los métodos de búsqueda y selección de testores para la aplicación en distintas áreas.

Capítulo 4

Conclusiones

A partir de los resultados obtenidos en el capitulo anterior podemos realizar el análisis pertinente sobre los beneficios que presenta el uso de los testores típicos para reducir la dimensionalidad de una base de datos. En primer lugar, podemos considerar la precisión obtenida en cuanto a la clasificación de los datos de prueba para ambos métodos de reducción. De igual manera, se puede tomar en cuenta las ventajas en cuanto a la proyección de la teoría de testores hacia nuevos elementos de la base de datos.

Tomando en cuenta los resultados de precisión del entrenamiento del modelo de clasificación SVM, podemos observar la capacidad de los testores típicos de realizar una clasificación adecuada y confiable. Como se puede observar en los resultados de la tabla 3.4, la precisión del entrenamiento usando las 4 características del testor típico, no solo igualaron, sino que superaron la precisión obtenida en el entrenamiento con todas las características. Por lo tanto, podemos considerar que

la reducción de características para el entrenamiento de un modelo de clasificación fue un éxito.

Habiendo analizado a los testores típicos como modelo de reducción exitoso, podemos realizar una comparación con un método conocido y utilizado como el PCA. Nuevamente, considerando la tabla 3.4, podemos observar la poca confiabilidad de clasificación que presentó el método PCA con el mismo número de características que el testor típico. Recordando el funcionamiento de este método, consideramos que el número de nuevas características no es suficiente para representar la mayor parte de la variabilidad de los datos de la base. En otras palabras, este método no es capaz de realizar una reducción de características tan grande como la de los testores típicos.

Dentro de la comparación entre la teoría de testores y el método PCA, podemos hallar algunos inconvenientes el segundo método. Como se mencionó anteriormente, los testores típicos tienen una capacidad de proyección sobre nuevos datos. Esto quiere decir que si se encuentra un testor típico dentro de una base de datos, cualquier nuevo elemento que sea clasificado a partir del mismo, mantendrá el mismo patrón. Por otro lado, la reducción de variable del método PCA se da través de una transformación total de la base de datos. Por lo tanto, si se ingresan nuevos elementos a la base, se debe realizar un nuevo PCA para poder transformar estos nuevo datos de la misma manera. Esta ventaja presentada en los testores típico presenta una gran utilidad en cuanto a la toma de datos dentro de la investigación. En el caso de la base datos utilizada, podríamos observar que haciendo uso del testor típico hallado, sería necesario solo tomar y almacenar mediciones de las

4 características del mismo.

En definitiva, la teoría de testores presenta grandes ventajas frente al método PCA no solo en términos de la precisión de entrenamiento de un modelo de clasificación sino también en cuanto a su capacidad de proyección frente a nuevas mediciones. Esta teoría aun cuenta con varias mejoras por realizar en términos de optimización de algoritmos y búsqueda de testores, sin embargo, muestra resultados alentadores como un nuevo método de reducción de dimensionalidad en comparación con métodos conocidos como el PCA.

4.1. Trabajos Futuros

4.1.1. Elección de Testores Típicos

Uno de los principales retos que se presentan en cuanto a la teoría de testores, es la elección de los mismos luego de ser calculados. Frente a este problema, surge una idea que se basa en el mismo concepto de los testores. Dentro de la teoría, la búsqueda de testores típico se centra en la discriminación entre clases distintas, es decir, se busca las características que distinguen a las clases. Ahora bien, la distinción entre clases no asegura la uniformidad de los elementos dentro de una misma clase. Por lo tanto, se puede explorar la idea de realizar la búsqueda de las características que permiten identificar la equivalencia de los elementos que provienen de la misma clase.

4.1.2. Aplicaciones

Dentro de este trabajo de investigación se pudo mostrar la utilidad de los testores típicos en una base de datos de clasificación de objetos astronómicos. Sin embargo, los problemas de clasificación puede hallarse en muchas otras áreas tales como la medicina. La teoría de testores presenta un gran potencial como método de reducción de características y se puede hallar una gran cantidad de aplicaciones. Como se pudo observar a partir de los resultados obtenidos, los testores típicos no solo tienen la capacidad de reducir una base de datos sino reducir el número de características necesarias para determinar la clase a la cual pertenece un objeto. Por consiguiente, es necesario continuar con el desarrollo de esta teoría para mostrar los grandes beneficios que ofrece en una sociedad donde la clasificación de información es indispensable.

Bibliografía

[1] Fedesoriano. Stellar classification dataset - sdss17, 2022. Last accessed 21 September 2022, https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17.

[2] A. Gallego, D. Torres, F. Álvarez, and A. Torres. Identificación de características de células de cáncer de mama por medio de testores típicos. *Research in Computing Science*, pages 43–54, 2017.

[3] M. Lazo-Cortes, J. Ruiz-Shulcloper, and E. Alba-Cabrera. An overview of the evolution of the concept of testor. *Pattern Recognition*, pages 753–762, 2001.

[4] E. Alba and R. Santana. Generación de matrices para evaluar el desempeño de estrategias de búsqueda de testores típicos. *Avances en Ciencias e Ingenierías*, pages 30–35, 2010.

[5] J. F. Martínez, J. A. Santos, and A. Carrasco. Feature selection using typical testors applied to estimation of stellar parameters. *Computación y Sistemas*, pages 15–23, 2004.

[6] R. A. Vásquez and S. Godoy-Calderón. Using testor theory to reduce the dimension of neural network models. *Research in Computing Science*, pages 93–103, 2007.

[7] E. Alba, J. Ibarra, S. Godoy, and F. Cervantes. Yyc: A fast performance incremental algorithm for finding typical testors. *Iberoamerican Congress on Pattern Recognition*, page 416–423, 2014.

[8] Jollifre I.T. *Principle Component Analysis*. Springer, 2 edition, 2002.

[9] H. Abdi and L. J. Williams. Principle component analysis. *WIREs Computational Statictics*, pages 433–459, 2010.

[10] K. K. Vasan and B. Surendiran. Dimensionality reduction using principal component analysis for network intrusion detection. *Perspectives in Science*, pages 510–512, 2016.

[11] A. Mammone, M. Turchi, and N. Cristianini. Support vector machines. *WIREs Computational Statistics*, pages 283–289, 2009.

[12] D. K. Srivastava and L. Bhambhu. Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, pages 1–7, 2010.