Escuela de Ingeniería en Computación

IC-6200 Inteligencia Artificial

Proyecto I de Machine Learning

Andrey Marín Chacón
David González Agüero

Campus Tecnológico Local San Carlos

II Semestre, 2023

Solución Planteada

El presente proyecto de Inteligencia Artificial tiene como objetivo el poder desarrollar una solución de Inteligencia Artirficial que aborde problemas de Machine Learning, en donde se construyan y analicen algoritmos para representación del conocimiento, búsqueda de control y aprendizaje. Todo esto integrado con una API el cuál funcionaría como un tipo de asistente personal que respondería preguntas relacionadas sobre 10 modelos creados de aprendizaje supervisado ya sean de predicción o clasificación.

Por lo tanto, se han escogido 10 sets de datos, entre los cuales 6 de estos son para predecir algún valor y los 4 restantes para clasificar características específicas. A cada uno de los sets de datos se les aplicará un algoritmo de Machine Learning dependiendo de lo que se quiere realizar (Clasificación o Predicción).

Arquitecturas de ML

Primero, según Arthur Samuel el machine learning es un subdominio de la Inteligencia Artificial que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente a partir de la experiencia. Se basa en la hipótesis subyacente de construir un modelo y tratar de mejorarlo ajustando más datos en el modelo a lo largo del tiempo.[2] Para este proyecto los algoritmos se entrenan por medio de las técnicas de machine learning basados en Aprendizaje supervisado. 'El aprendizaje supervisado es la tarea de aprendizaje automático que consiste en aprender una función que mapea una entrada a una salida basada en pares de entrada-salida de ejemplo'. [3] Por lo que a continuación, se van a detallar los algoritmos de aprendizaje supervisado utilizados en los conjutos de datos seleccionados.

Regresión Lineal.

El algoritmo de regresión lineal está basado en aprendizaje supervisado, esto quiere decir que va a recibir un conjunto de datos etiquetado. El modelo generado por este algoritmo se va a corresponder con una función lineal la cuál el algoritmo va a intentar predecir valores continuos.

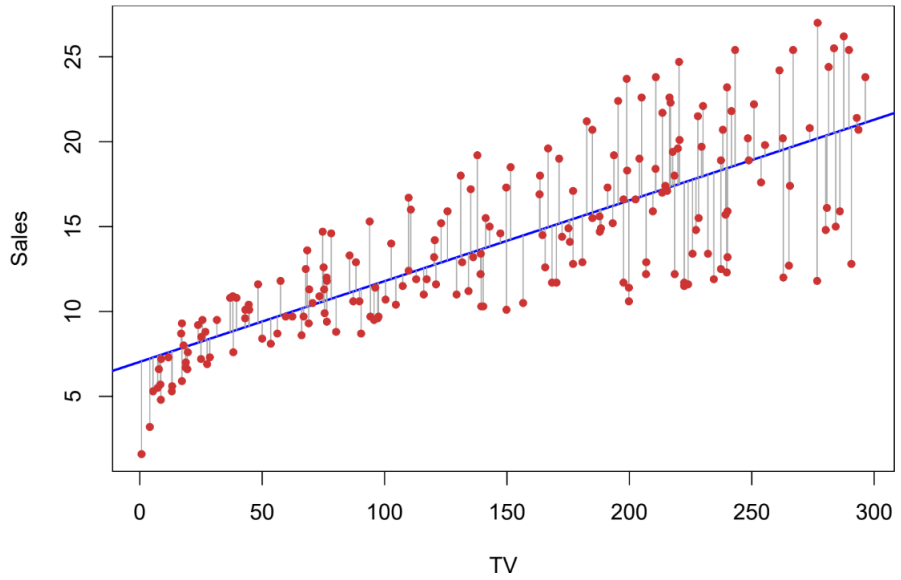

Figura 1: Ejemplo Gráfico de Regresión Lineal Simple[1]

Los algoritmos de Regresión Lineal generalmente son utilizados para calcular predicciones. En los casos en que solamente se le pasa una característica o variable de entrada al algoritmo para predecir la variable de salida, se le conoce como Regresión Lineal Simple. Para los casos en los que se le pasa al algoritmo más características de entrada para predecir una característica de salida, se le conoce como Regresión Lineal Múltiple.

Máquinas de Soporte Vectorial(SVM).

Las Máquinas de Soporte Vectorial o en inglés conocidas como Support Vector Machines (SVM) son algoritmos con un gran potencial, ya que una de sus virtudes es que SVM permite realizar tareas de regresión(predecir valor continuo) y tareas de clasificación(valores discretos). También una de las características de este algoritmo es que funciona bien con conjunto de datos pequeños y complejos. Las Máquinas de Soporte Vectorial están basadas en aprendizaje supervisado, por lo que va a recibir un conjunto de datos etiquetado. Existen diferentes tipos de algoritmos de SVM y todos dependen en función del conjunto de datos que recibiría:

- Para conjunto de datos linealmente separables se tienen el *Hard Margin Classification* y el *Soft Margin Classification*.
- Para conjunto de datos que no son linealmente separables se tiene los

Kernel, entre estos podemos encontrar los que son con Kernel Polinómicos y con Kernel Gaussiano.

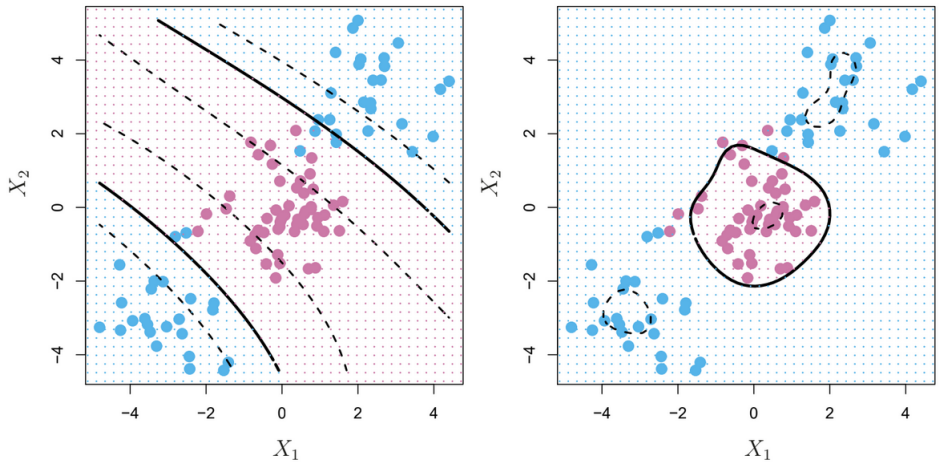

Figura 2: Ejemplo de SVM con Kernel Polinómico[1]

Árbol de decisión.

Los árboles de decisión generalmente proporcionan buenos resultados y se engloban dentro de la categoría de algoritmos basados en aprendizaje supervisado, lo que quiere decir que este algoritmo recibirá un conjunto de datos etiquetado. Son clasificadores no lineales por lo que van a ser capaces de construir modelos no lineales. Una de las principales características de los árboles de decisión es que pueden predecir valores continuos y valores discretos.

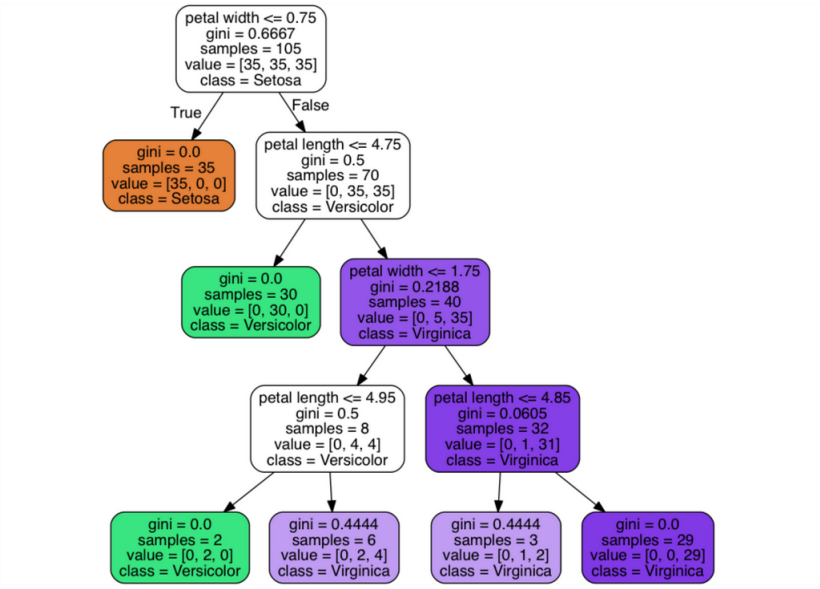

Figura 3: Ejemplo Árbol de decisión en sklearn[4]

Random Forest (Conjunto de árboles).

El algoritmo de Random Forest se conforma de un ensamble de árboles de decisiónes, en donde para un conjunto de entrenamiento se van a seleccionar varias instancias del algoritmo de árbol de decisión para entrenar el modelo. Random Forest se entrena utilizando la técnica de Ensemble Learning conocida como Bagging. También lo que hace este algoritmo es una seleccion de un un subconjunto de características de manera aleatoria, esto hace que entre las clasificaciones que realiza cada una de las instancias del algoritmo entrenadas con diferentes subconjuntos varían entre ellas, llegando así a splucionar una de las limitantes que los árboles de decisiones tienen, que es el sobreentrenamiento.

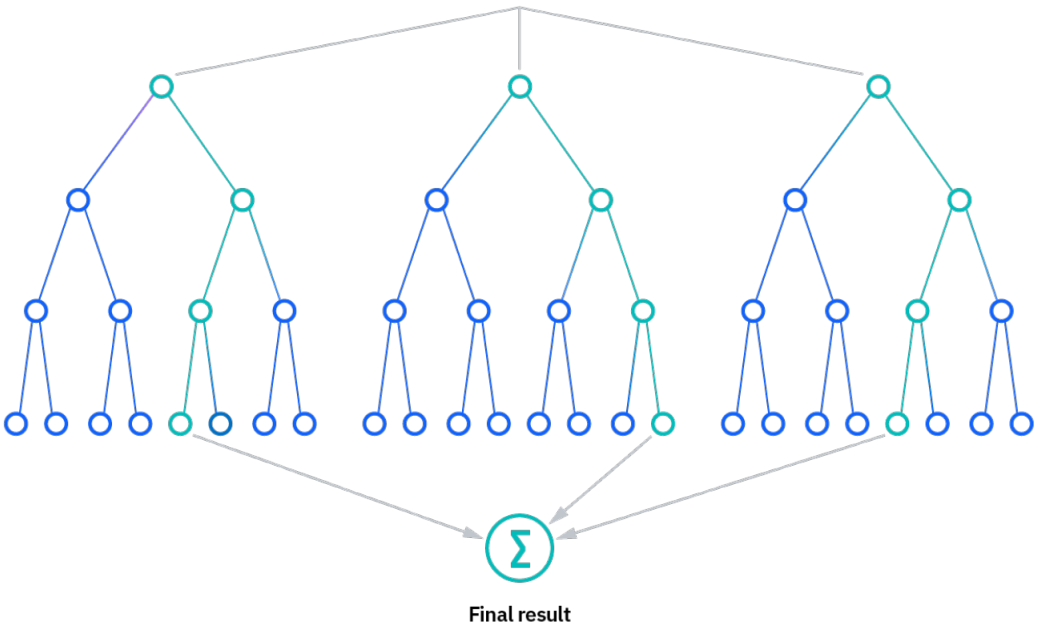

Figura 4: Ejemplo Random Forest

Modelos desarrollados

En la presente sección se presentará los temas y algoritmos que se escogieron para realizar los 10 modelos de Machine Learning en donde 6 corresponden a algoritmos de predicción de datos y 4 de clasificación de datos.

3.1. Predecir el precio de un automovil.

El conjunto de datos contiene información sobre diferentes automóviles. El Objetivo principal es crear un modelo el cuál logre predecir el precio de un automóvil con resptecto al conjunto de características que este posea. El conjunto de datos contiene 301 observaciones de las cuáles 4 son categóricas y el resto son numéricas. Esto quiere decir que para poder entrenar el modelo, los algoritmos que contienen las librerias de sklearn, necesitan datos numéricos, por lo que hay que transformar los datos categóricos a numéricos para poder pasarlos como parámetros de entrada.

Una vez transformados los datos categóricos a numéricos, se procede a observar la correlación que existe entre las características, para ello se toma la variable de salida que en este caso se llama **Selling Price** y se compara con el resto de características de entrada. En la siguiente figura se puede apreciar

los valores de cada característica con respecto a la variable a predecir:

```
Selling_Price    1.000000
Present_Price    0.878983
Car_Name         0.499198
Year             0.236141
Kms_Driven       0.029187
Owner           -0.088344
Transmission    -0.367128
Fuel_Type       -0.509467
Seller_Type     -0.550724
Name: Selling_Price, dtype: float64
```

Figura 5: Correlación entre Selling Price

Seguidamente el tamaño del conjunto de datos es de 301 filas o observaciones, por lo que a la hora se hacer la división entre datos para entrenar el modelo y los datos de prueba, se hacen por medio de 80/20, en donde el ochenta porciento le corresponde a los datos de entrenamiento (240 observaciones) y el veinte porciento le corresponde a los datos para probar el modelo (61 observaciones).

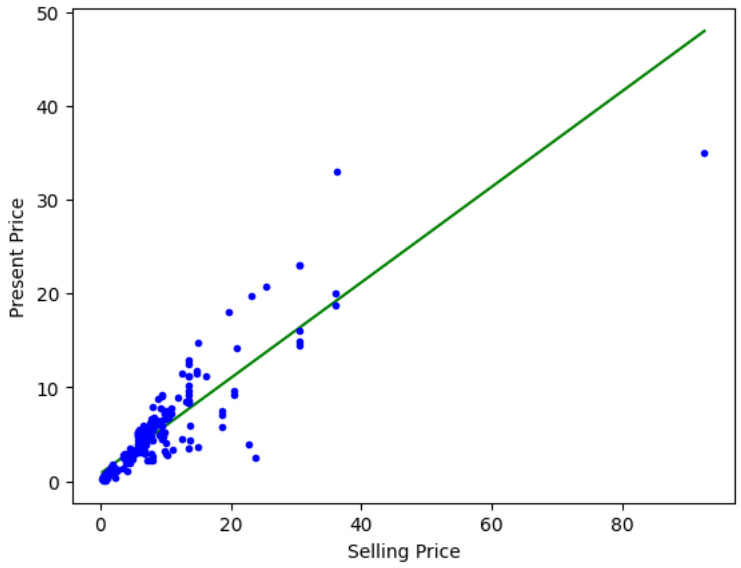

Figura 6: Present Price y Selling Price, RLS

En la Figura 6 se observa un gráfico en donde en el eje X tenemos el valor de **Selling Price** y en el eje Y se tiene la caracteristica de **Present Price**. La línea de color verde es el modelo que se ha generado al entrenar el algoritmo de regresión lineal con los datos de entrenamiento. Para este

ejercicio se hicieron pruebas utilizando los Algoritmos de Regresión Lineal Simple y el de Regresión Lineal Múltiple.

3.2. Predecir el precio de acciones.

El conjunto de datos contiene información histórica de precios de acciones de empresas que forman parte del índice S&P 500 durante los últimos 5 años. Se pretende predecir el precio de Cierre de la acción.

El conjunto de datos contiene una cantidad de 619040 observaciones las cuales entre sus características observamos que 2 son de tipo object o categóricas y 5 son numéricas.

En este caso se el conjunto de datos presenta valores nulos para tres características: open, high, low. Por lo que se aplica la técnica de rellenar valores nulos aplicando la media de cada una de sus columnas en específico. Se verifica la correlación que existe entre las características contra la variable a predecir, que en este caso se llama **close**:

```
  corr_matrix = df.corr()
close     1.000000
low       0.999939
high      0.999936
open      0.999872
Name     -0.032868
volume   -0.142802
Name: close, dtype: float64
```

Figura 7: Correlación entre close.

Después de aplicarle a los datos categóricos la transformación a datos numéricos, se procede a realizar la división del set de datos en un set de datos para entrenamiento y otro set de datos para las pruebas. Como en este escenario se tienen bastantes datos 619040 específicamente, se realiza una división de 70/30, esto quiere decir que el 70 % de los datos serán para entrenamiento del modelo mientras que el 30 % restante será utilizado para las pruebas quedando de la siguiente manera:

- Datos para entrenamiento: 433328.
- Datos para las pruebas: 185712.

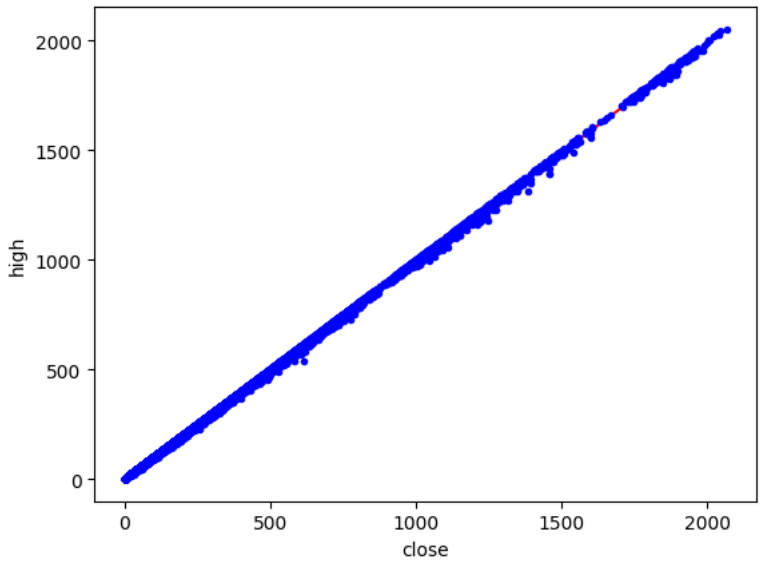

Figura 8: high y close, RLS.

En la Figura 8 se observa un gráfico en donde en el eje X tenemos el valor de **close** y en el eje Y se tiene la caracteristica de **high**. De color Azul están representados los datos mientras que la línea de color rojo es el modelo que se ha generado al entrenar el algoritmo de regresión lineal con los datos de entrenamiento. Para este ejercicio se utilizó el algoritmo de Regresión Lineal Simple, ya que los resultados con la Regresión Lineal Simple fueron bastante buenos.

3.3. Predecir el cantidad de crímenes en Londres.

El conjunto de datos se refiere a informes de crímenes en áreas metropolitanas importantes, como Londres. Los datos incluyen información sobre el número de informes criminales registrados mensualmente en diferentes distritos (LSOA borough) y categorías tanto principales como secundarias, abarcando el período desde enero de 2008 hasta diciembre de 2016.

El conjunto de datos contiene 13490604 observaciones, en donde sus características poseen tipos categóricos y numéricos. De estos 4 son categóricos y 3 numéricas. Continuando con el análisis del conjunto de datos, se observó que todas las observaciones están completas, sin valores nulos. Después de transformar los datos categóricos a numéricos se aplicó la correlación entre la variable a predecir con el resto de caracteristicas. La variable a predecir es **value**.

```
value             1.000000
major_category    0.035985
borough           0.027491
lsoa_code         0.025487
minor_category    0.020980
month             0.001821
year             -0.002198
Name: value, dtype: float64
```

Figura 9: Correlación entre value

Seguidamente se procede a realizar la división del set de datos en un set de datos para entrenamiento y otro set de datos para las pruebas aplicando 70/30 ya que se tienen 13490604 observaciones. El 70 % de los datos serán para entrenamiento del modelo mientras que el 30 % restante será utilizado para las pruebas quedando de la siguiente manera:

- Datos para entrenamiento: 9443422.
- Datos para las pruebas: 4047182.

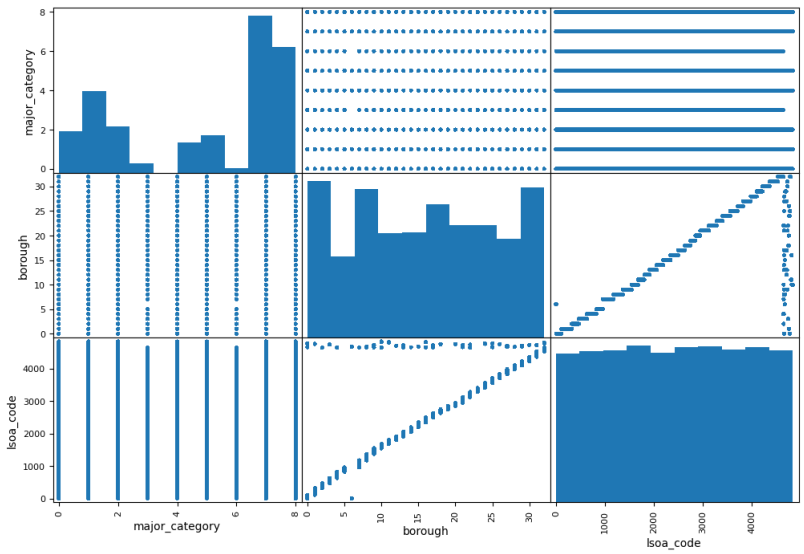

Figura 10: Correlación entre todas las caracteristicas

En la figura 10 se observa gráficamente la correlación que existe entre todas las características.

3.4. Clasificar si un cliente abandona un servicio.

El conjunto de datos contiene información sobre una empresa de telecomunicaciones la cuál se pretende clasificar si un cliente abandona o no el servicio ofrecido por la empresa.

Para este conjunto de datos se obtuvieron un total de 7043 observaciones entre las cuales sus atributos contienen 18 de tipo categórico y 3 de tipo numéricos. La variable a clasificar es **Churn**. En el siguiente histograma podemos observar la cantidad de opciones que nos ofrece esta variable:

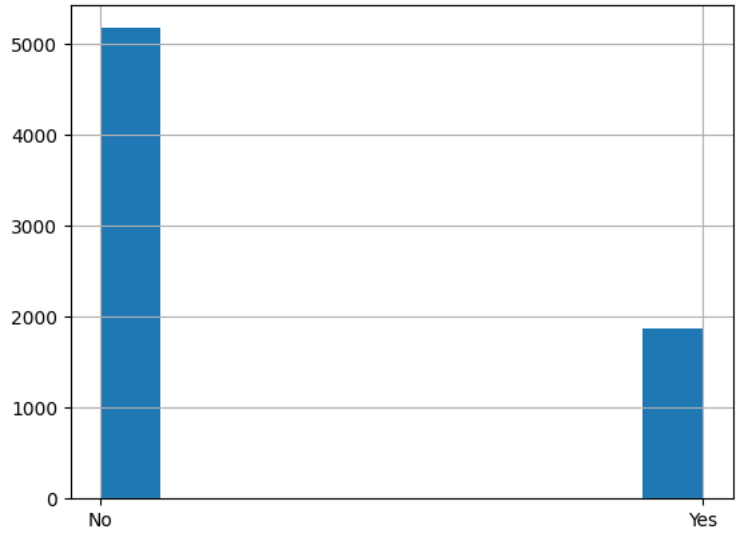

Figura 11: Histograma de la categoría Churn

También cabe resaltar que el conjunto de datos no presenta valores nulos, por lo que no hay necesidad de aplicar alguna técnica para resolver este problema. A continuación, se va a representar de forma gráfica las características de **TotalCharges** junto con la variable a clasificar **Churn**:

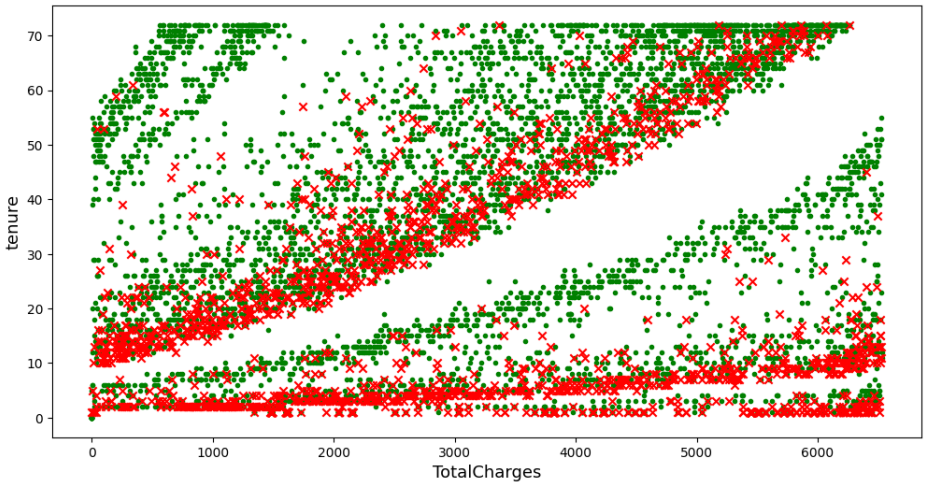

Figura 12: Representación entre TotalCharges y Churn

Los puntos de color verde tenemos los valores de Churn que equivalen a No, y las X de color Rojo representan aquellos valores que tienen como categoría Yes.

Similar a los casos anteriores, a los atributos categóricos se les tiene que aplicar la transformación de categóricos a numéricos por medio de la librería de sklearn LabelEncoder.

Después de aplicar la transformación se verificó la correlación que existe entre las características de entrada y la característica de salida:

```
Churn              1.000000
PaperlessBilling   0.191324
MonthlyCharges     0.188574
SeniorCitizen      0.147078
PaymentMethod      0.100844
MultipleLines      0.034952
TotalCharges       0.019082
PhoneService       0.008886
gender            -0.011729
StreamingMovies   -0.032716
StreamingTV       -0.037937
InternetService   -0.047171
Partner           -0.142266
Dependents        -0.162356
DeviceProtection  -0.172989
OnlineBackup      -0.192645
TechSupport       -0.278303
OnlineSecurity    -0.288143
tenure            -0.344925
Contract          -0.394085
Name: Churn, dtype: float64
```

Figura 13: Correlación entre Churn

Seguidamente al realizar la división del conjunto de datos en datos de entrenamiento y datos de pruebas, se obtuvo lo siguiente:

- Datos para entrenamiento: 5634.
- Datos para las pruebas: 1409.

3.5. Clasificar accidente cerebrovascular.

El conjunto de datos se centra en predecir la probabilidad de que un paciente experimente un derrame cerebral. Por lo que se pretende clasificar esta probabilidad segpun los atributos proporcionados como entrada.

Para este conjunto de datos se obtuvieron un total de 5110 observaciones entre las cuales sus atributos contienen 5 de tipo categórico y 7 de tipo numéricos. De todas estas observaciones la característica **bmi** contiene valores nulos. Para corregir o rellenar los valores nulos se utiliza la media de la tabla bmi para sustituir los nulos con el resultado de la mediana.

La variable a clasificar es **stroke**. En el siguiente histograma podemos observar la cantidad de opciones que nos ofrece esta variable:

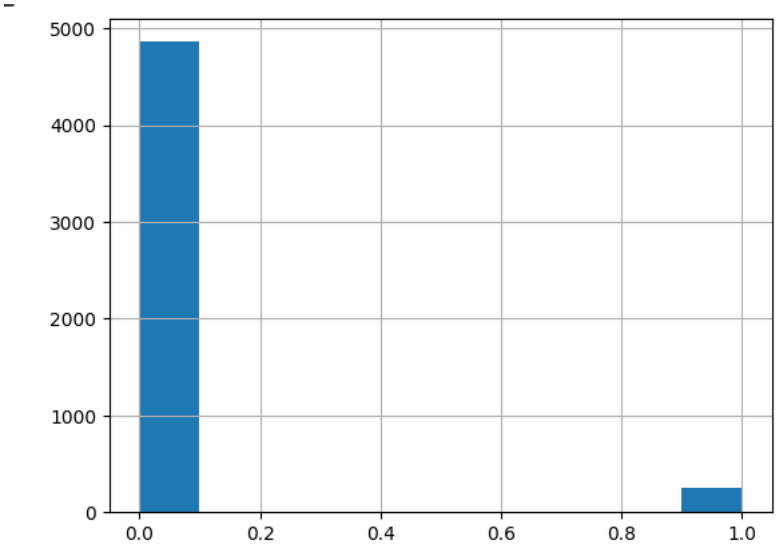

Figura 14: Histograma de la categoría Stroke

Al ver el histograma anterior, se ve un gran desbalanceo en los datos. La etiqueta 1 quiere decir los pacientes que han tenido un ataque cerebrovascular, y la etiqueta 0 son los pacientes que no han tenido un ataque de este tipo.

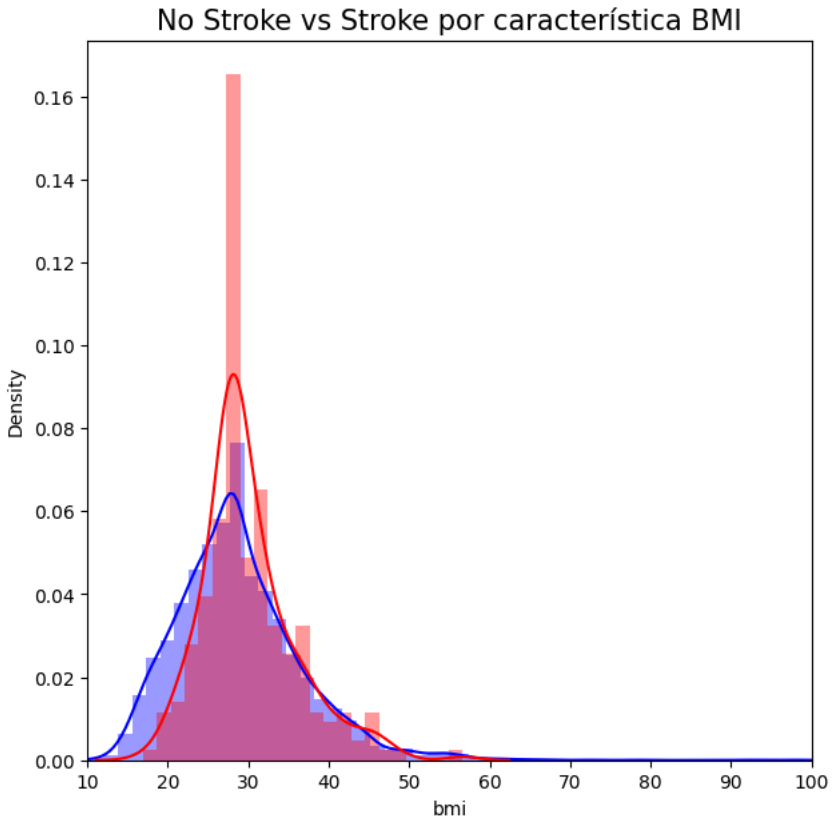

Figura 15: Histograma de Stroke por BMI

La Figura 15 nos muestra los casos que han tenido un ataque cerebrovascular vs los que no lo han tenido, haciendo una relación con la variable **Bmi**. La linea continua de color rojo representa aquellos pacientes que han sufrido un ataque, mientras que la azul representa los que no lo han sufrido,

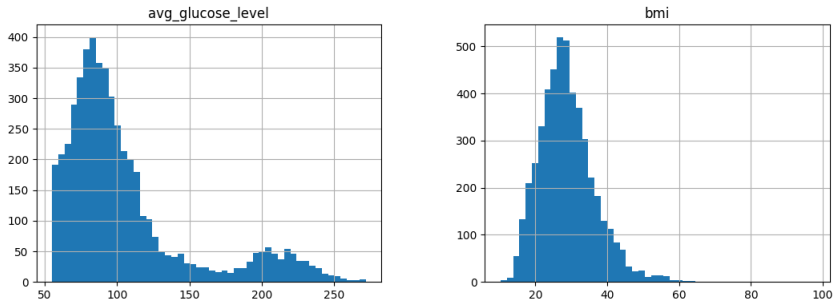

Figura 16: Histograma del nivel de glucosa y el bmi

Los atributos categóricos se les tiene que aplicar la transformación de

categóricos a numéricos por medio de la librería de sklearn LabelEncoder.

Después de aplicar la transformación se verificó la correlación que existe entre las características de entrada y la característica de salida:

```
stroke               1.000000
age                  0.245257
heart_disease        0.134914
avg_glucose_level    0.131945
hypertension         0.127904
ever_married         0.108340
bmi                  0.036110
smoking_status       0.028123
Residence_type       0.015458
gender               0.008929
work_type           -0.032316
Name: stroke, dtype: float64
```

Figura 17: Correlación entre stroke

Seguidamente al realizar la división del conjunto de datos en datos de entrenamiento y datos de pruebas, se obtuvo lo siguiente:

- Datos para entrenamiento: 4088.
- Datos para las pruebas: 1022.

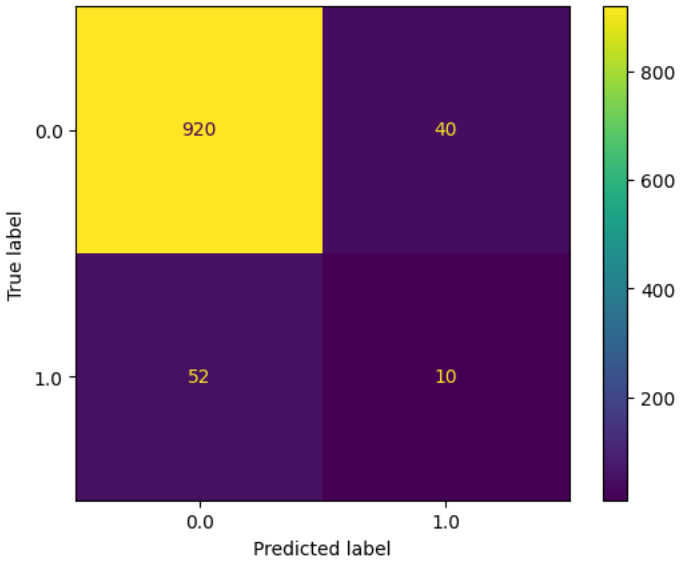

Figura 18: Matriz de confusión de stroke

3.6. Clasificar tipo hepatitis.

El conjunto de datos contiene valores de laboratorio de donantes de sangre y pacientes con hepatitis C. Se pretende categorizar el tipo de hepatitis de un paciente.

El data set presenta 615 observaciones las cuales 2 de sus características son categóricas y 12 son numéricas. Además, existen varias características que poseen valores nulos por lo que se procede a aplicar la media ya que son pocos valores nulos y así llegar a rellenar estos valores nulos. La variable a clasificar es **Category** y en la siguiente figura podremos ver de manera visual el gráfico de barras que muestra la cantidad de cada categoría de enfermedad:

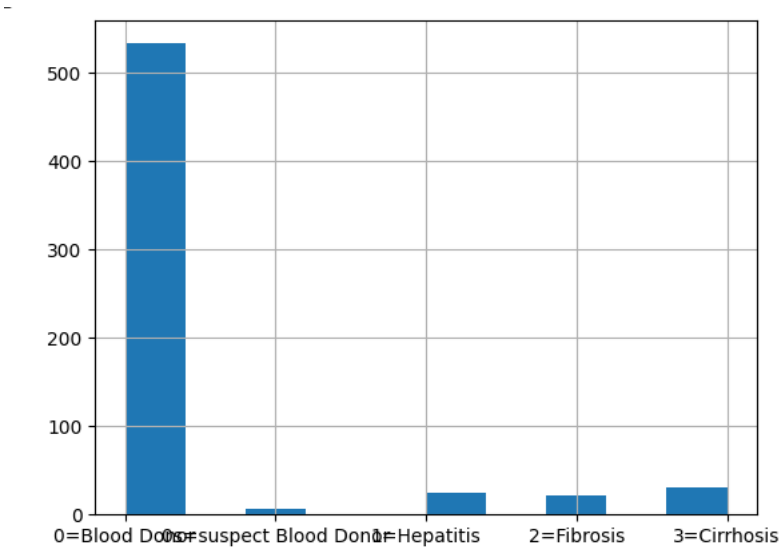

Figura 19: Histograma de Category

En el siguiente gráfico se muestra la relación de la característica **ALT** con **Category**. De color rojo son los casos en los que se tiene Hepatitis, y de color azul los pacientes donadores de sangre:

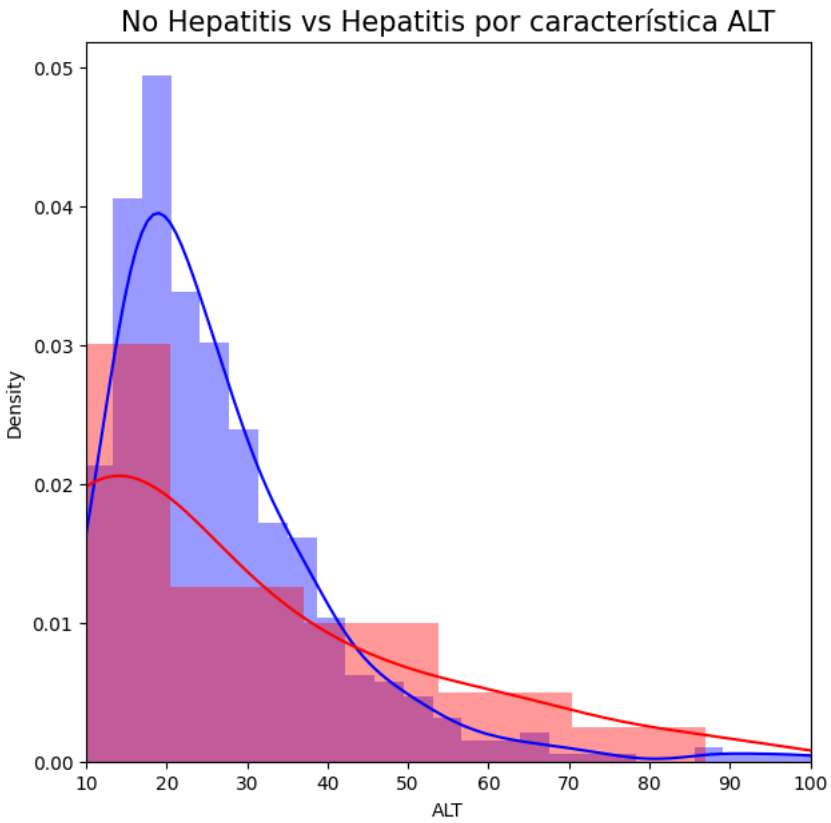

Figura 20: Histograma de Category con ALT

Seguidamente, despues de aplicar la transformación de los datos categóricos a numéricos, tenemos la siguiente correlación de las características contra la variable a predecir Category:

```
Category    1.000000
AST         0.648341
BIL         0.473006
GGT         0.471164
CREA        0.182040
Age         0.106341
ALT         0.105831
Sex         0.060657
ALP         0.022193
PROT        0.007321
ALB        -0.285147
CHOL       -0.300914
CHE        -0.329472
Name: Category, dtype: float64
```

Figura 21: Correlación con Category

Después de realizar la división de los datos, los conjuntos de prueba y

entrenamientos quedaron con la siguiente cantidad de observaciones:

- Datos para entrenamiento: 492.
- Datos para las pruebas: 123.

Matriz de Confusión obtenida después de aplicar el modelo de Árbol de decisión:

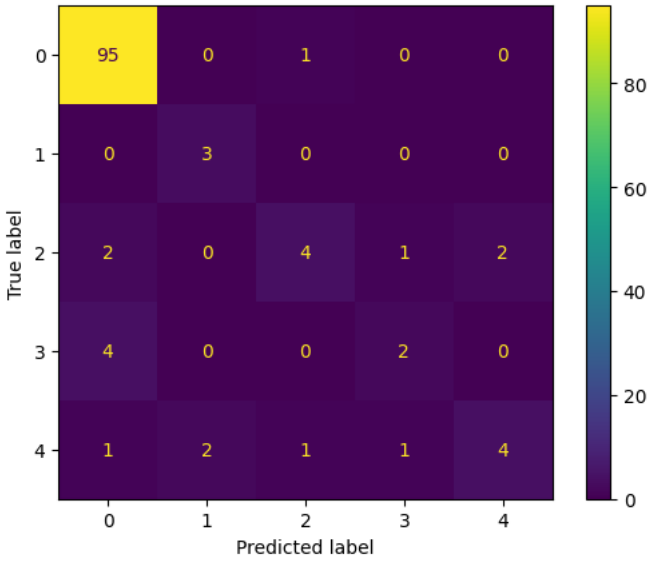

Figura 22: Matriz de confusión de Category

3.7. Predecir masa corporal.

El conjunto de datos contiene información sobre el porcentaje de grasa corporal en 252 hombres. Estas estimaciones se basan en mediciones de densidad corporal obtenidas mediante pesaje bajo el agua y diversas medidas de circunferencia corporal. El objetivo principal es llegar a predecir la masa corporal de una persona con respecto al conjunto de características que posee.

El conjunto de datos contiene 252 observaciones, en donde todas sus características son de tipo numéricos. También todas las observaciones no poseen valores nulos, y la distribución de sus atributos en la mayoría tienden a la campana de Gauss:

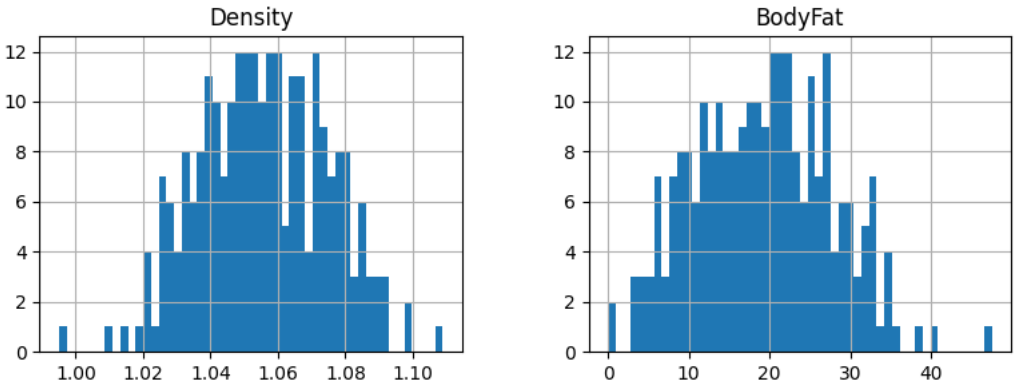

Figura 23: Distribución de Density y BodyFat

La variable a predecir es **BodyFat** por lo que se verificó su correlación con todos los atributos:

```
value            1.000000
major_category   0.035985
borough          0.027491
lsoa_code        0.025487
minor_category   0.020980
month            0.001821
year            -0.002198
Name: value, dtype: float64
```

Figura 24: Correlación entre value

Para la división de los datos de entrenamiento y los datos de prueba, se aplicó una división de 80/20 debido a que son poquitos datos. Quedando de la siguiente manera:

- Datos para entrenamiento: 201.
- Datos para las pruebas: 51.

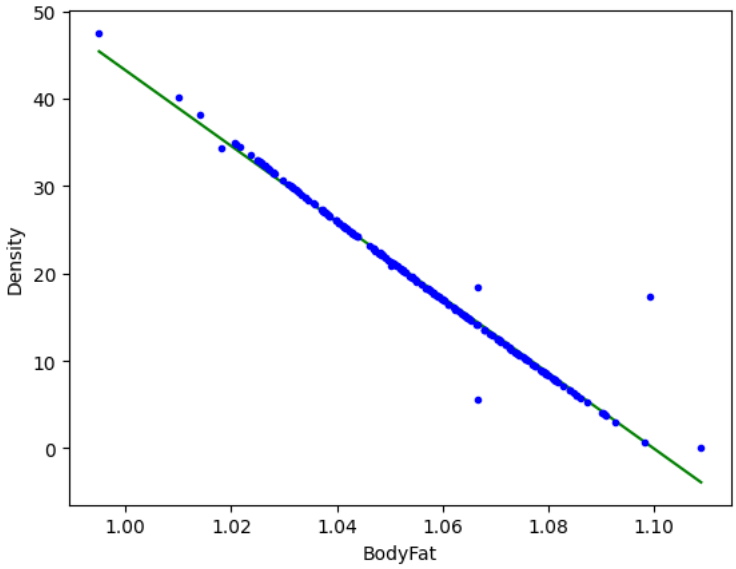

Figura 25: Density y BodyFat, RLS.

La Figura 25 muestra de color azul el conjunto de datos a favor de las características Density y BodyFat. Claramente se observa su adaptación a la línea verde que es el modelo de Regresión Lineal generado a partir de los datos de entrenamiento.

3.8. Clasificar el estado de la Cirrosis.

Para este conjunto de datos se quiere clasificar la Etapa de cirrosis en la que se encuentra un paciente. Los datos contienen la información recopilada del ensayo clínico de la Clínica Mayo sobre la cirrosis biliar primaria (CBP) del hígado, realizado entre 1974 y 1984.

El set de datos contiene un total de 418 observaciones. En donde sus columnas o características tienen 7 de tipo categórico y 13 de tipo numérico.

En este caso la mayoría de características poseen valores nulos, sin embargo en vez de eliminar estas filas, se decide aplicar la técnica de calcular la media en cada característica y añadir estos valores en las columnas que contengan nulos. Para este set de datos la variable a categorizar es **Stage**.

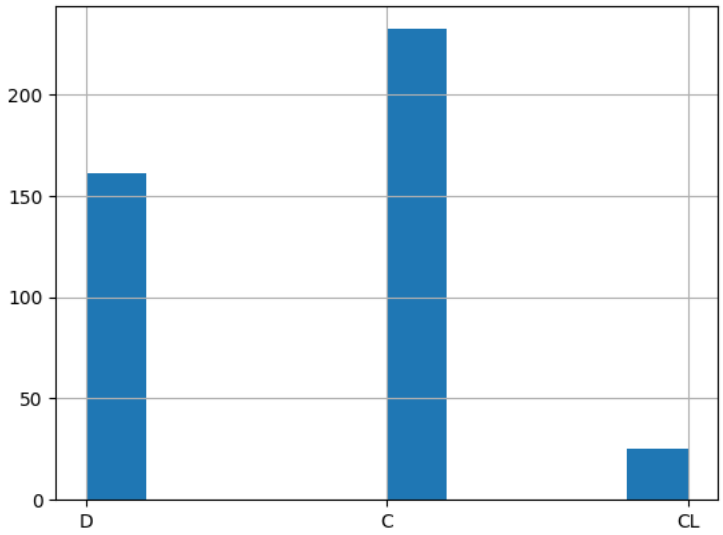

Figura 26: Histograma de Status

Seguidamente podemos ver un diagrama de Stage con respecto a la categoría Albumin:

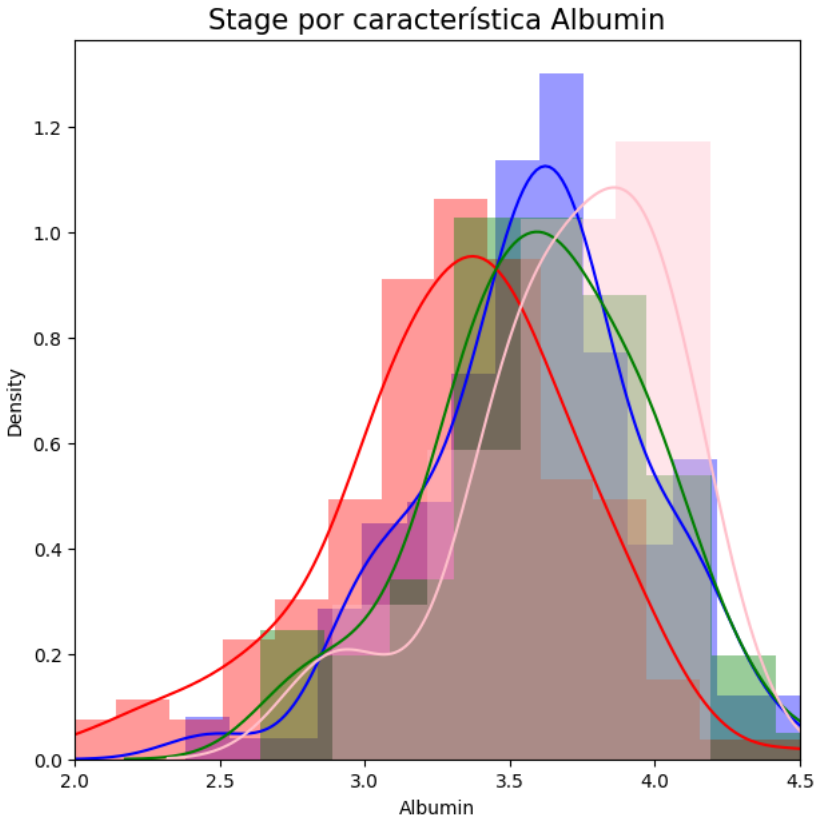

Figura 27: Stage respecto al Albumin

La línea continua de color Azul representa la categoría 3, la línea contínua de color Rojo representa la categoría 4, la de color verde representa la categoría 2 y de color rosado representa la categoría 1.

```
Stage            1.000000
Status           0.317177
Edema            0.243093
Copper           0.232149
Hepatomegaly     0.211329
Prothrombin      0.205981
Bilirubin        0.200314
Age              0.187852
SGOT             0.143568
Spiders          0.103316
Trygliceride     0.099879
Ascites          0.042093
Alk_Phos         0.037905
Drug             0.018274
Sex              0.017356
Cholesterol      0.009930
Platelets       -0.239594
Albumin         -0.302190
N_Days          -0.362013
Name: Stage, dtype: float64
```

Figura 28: Correlación respecto al Stage

Este set de datos se ha probado utilizando varios algoritmos de Machine Learning para comparar los resultados. En la sección de análisis de resultados 4.8 se muestra los score obtenidos para cada algoritmo.

La Matriz de Confusión muestra que para las categorías el modelo de Random Forest ha estado no tan acertivo:

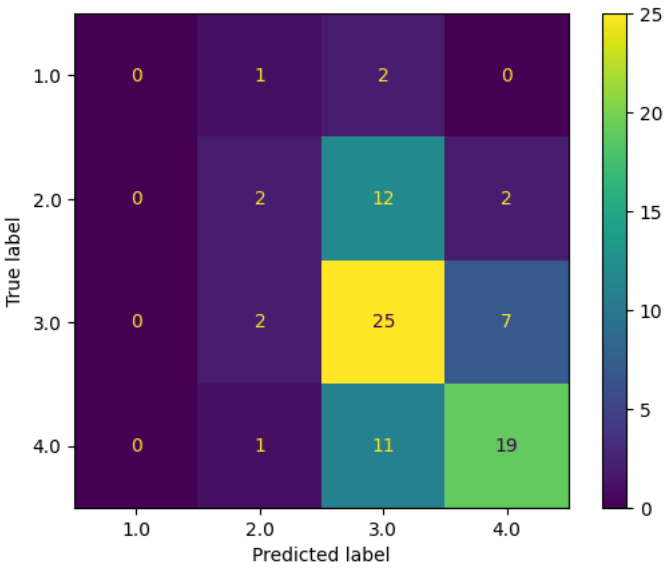

Figura 29: Matriz de Confusión respecto al Stage

3.9. Predecir el precio del aguacate.

El conjunto de datos contiene información de datos de escaneo minorista que muestra información semanal de ventas minoristas de aguacates Hass en 2018. Como principal objetivo se pretende predecir el precio del aguacate.

Para este conjunto de datos se obtienen unas 18249 observaciones las cuáles ninguna de estas poseen valores nulos. De todos los atributos se tienen 3 de tipo categórico y 11 de tipo numérico. Después de aplicar la transformación a los 2 datos categóricos a numéticos, se procedió a verificar la correlación que hay entre la variable de salida **AveragePrice** con el resto de características.

```
AveragePrice              1.000000
type                      0.615845
order_week_of_the_year    0.146383
order_month               0.119089
order_year                0.093197
region                   -0.011716
XLarge Bags              -0.117592
season                   -0.162649
4225                     -0.172928
Large Bags               -0.172940
Small Bags               -0.174730
Total Bags               -0.177088
4770                     -0.179446
Total Volume             -0.192752
4046                     -0.208317
Name: AveragePrice, dtype: float64
```

Figura 30: Correlación entre AveragePrice

Al ser más de 1000 observaciones se decidió dividir los datos usando 70/30, quedando de la siguiente manera:

- Datos para entrenamiento: 12774.
- Datos para las pruebas: 5475.

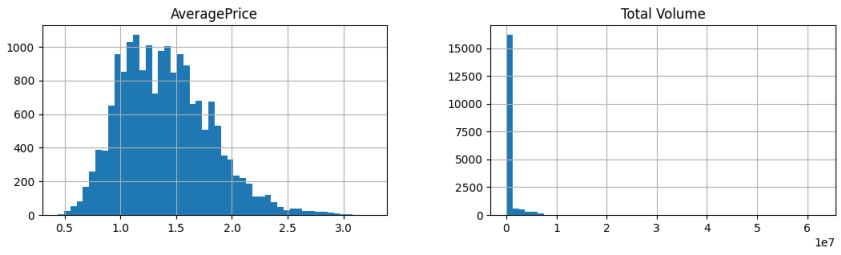

Figura 31: Distribución de AveragePrice y TotalValue

En la Figura 31 se muestra la distribución de los atributos AveragePrice y TotalValue. El resto de atributos tienen una distribución parecida a la de TotalValue.

3.10. Predecir las ventas de una compañia.

El conjunto de datos contiene información sobre datos históricos de ventas de 1,115 tiendas Rossmann. Por lo que se pretende predecir las ventas de con respecto al conjunto de características que posee.

Entre la cantidad de datos que posee el set de datos de esta compañia se obtienen un total de 1017209 observaciones las cuáles de estas tienen 2 atributos categóricos y 7 numéricos. Se verificó si el set de datos contiene valores nulos y al mostrar los resultados no aparecieron nulos por lo que solamente abría que hacer la transformación de las características categóricas a numéricas para que el algoritmo de Machine Learning pueda procesar los datos. Al realizar la correlacion de los atributos contra la variable de salida se obtuvo los siguientes resultados:

```
Sales            1.000000
Customers        0.894711
Open             0.678472
Promo            0.452345
SchoolHoliday    0.085124
Store            0.005126
StateHoliday    -0.229029
DayOfWeek       -0.462125
Name: Sales, dtype: float64
```

Figura 32: Correlación entre Sales

La división del conjunto de datos se realizo de la manera 70/30:

- Datos para entrenamiento: 712046.
- Datos para las pruebas: 305163.

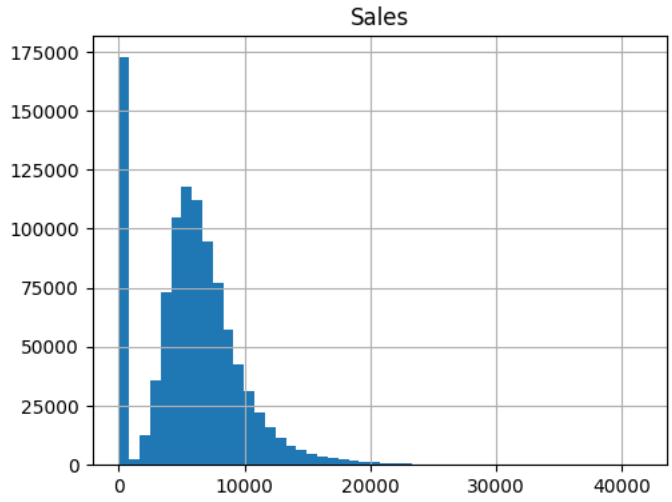

Figura 33: Distribución de Sales

A continuación, se muestra un gráfico el cuál de color azul tiene representado los datos y de color verde la línea que representa el modelo de Regresión lineal ajustado al conjunto de datos:

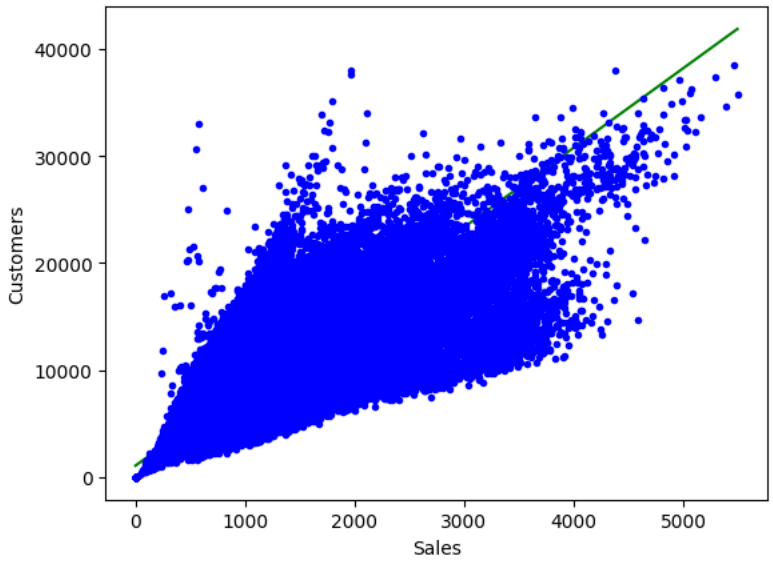

Figura 34: Modelo Generado entre Sales y Customers, RLS

Análisis de resultados obtenidos

El análisis de resultados nos ayuda a comparar si las predicciones o clasificaciones de los 10 modelos creados son buenas o malas. Para esto se hace uso de diferentes métricas que permiten evaluar los resultados como el rendimiento de los algoritmos creados para cada uno de los sets de datos.

4.1. Predecir el precio de un automovil.

Para el data set relacionado al precio de un automóvil, observamos en la sección anterior que la correlación que existe entre la variable de salida **Selling Price** con el restro de característica(ver Figura 5) mostraba una correlación positiva de 0.87 de la variable **Present Price** y la siguiente que más se le acercaba era la característica **Seller Type** con una correlación negativa de -0.55.

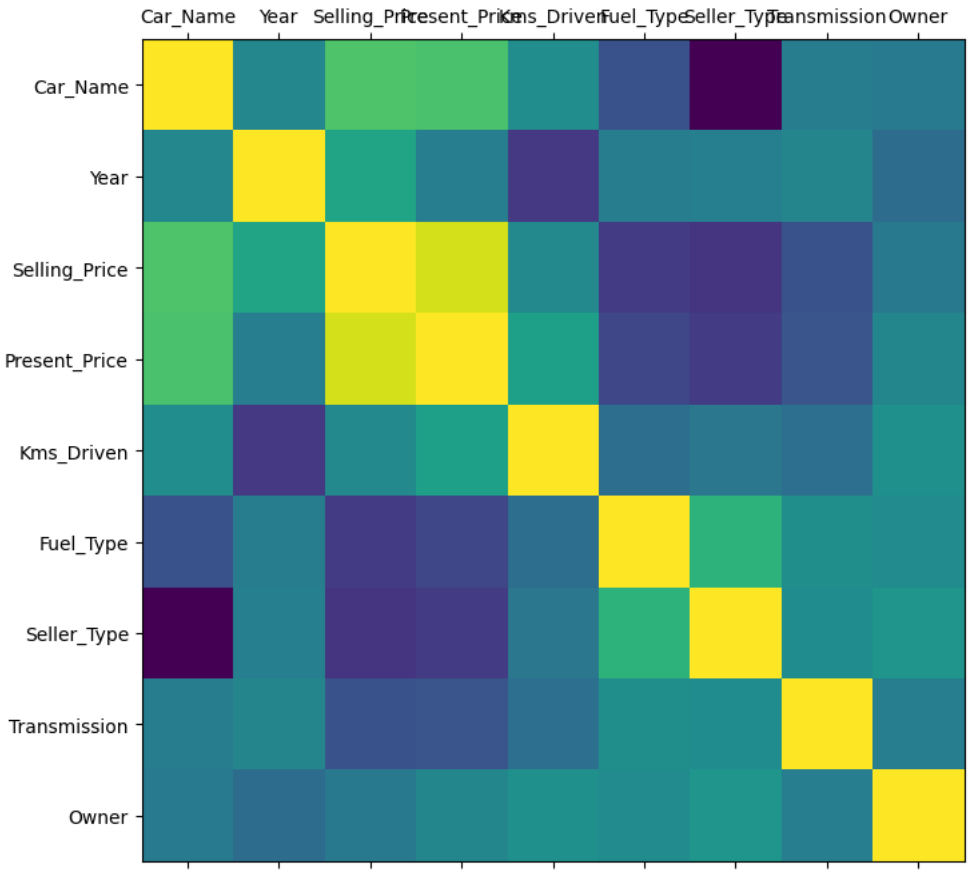

Figura 35: Matriz de correlación entre Selling Price

Al conjunto de datos se le realizó un escalado a la columna **Kms Driven** la cuál generó nuevos valores nulos. Al generar nuevos valores nulos se sustituyeron los valores nulos por la mediana de esta característica.

Para el caso de **Regresión Lineal Simple**, los resultados obtenidos son los siguientes:

- Parámetro Theta 0: 0.81.
- Parámetro Theta 1: array([0.50970327]).
- Score: 0.69.

Para el caso de **Regresión Lineal Múltiple**, se obtuvo lo siguiente:

- Score: 0.83.

Como podemos observar en la Figura 6 el modelo de regresión lineal simple se adapta en cierta parte al conjunto de datos de entrenamiento. Solamente utilizando la característica de Present Price, el modelo predice un 69 % de la variabilidad de los datos. Por lo que no se recomienda en ambientes donde se necesita una precisión más alta. Sin embargo haciendo una regresion lineal multiple con las tres caracteristicas escogidas, el modelo incrementó su precisión a un 83 %.

4.2. Predecir el precio de acciones.

Para el data set relacionado al precio de las acciones de empresas que pertenecen al S&P, pudimos observar que hay 3 variables con una correlación positiva del 99 % entre la variable de salida **close** (ver Figura 7), estas tres características son: low, high y open.

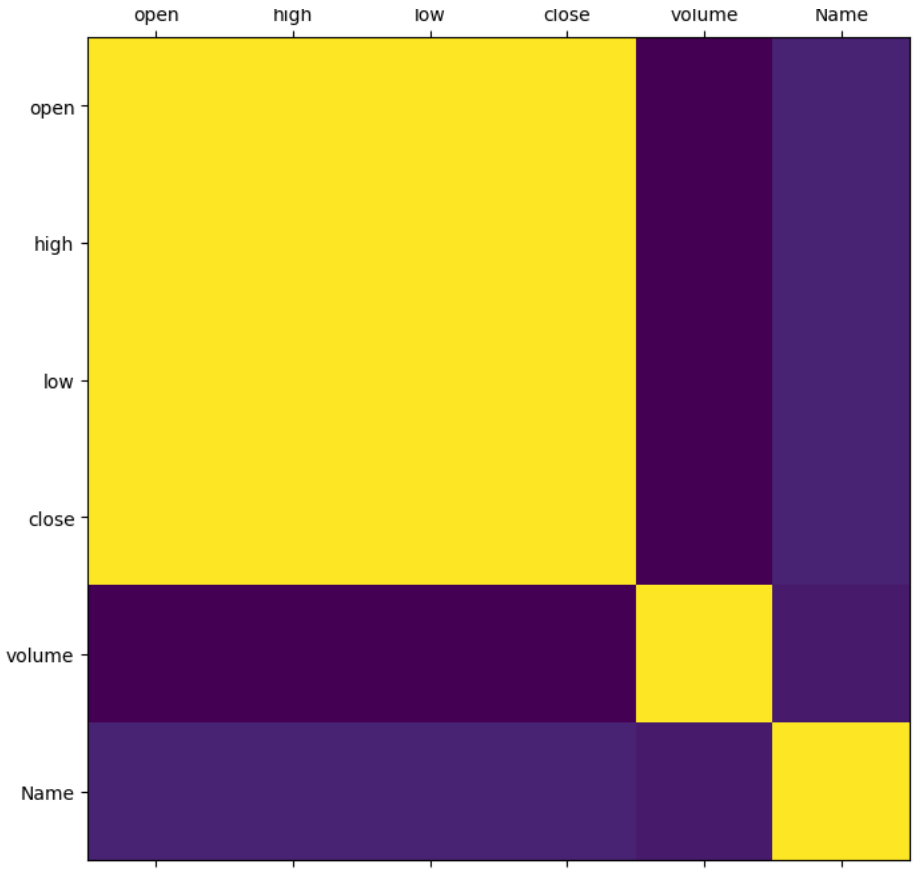

Figura 36: Matriz de correlación entre close

Para el algoritmo de **Regresión Lineal Simple**, los resultados obtenidos fueron los siguientes:

- Mean squared error: 2.48.
- Varianza: 1.00.
- Score: 0.99.

Como podemos observar en la gráfica el modelo de regresión lineal simple se adapta bastante bien al conjunto de datos de entrenamiento. El modelo predice un 99 % de la variabilidad de los datos. Por lo que se recomienda en su uso en este escenario.

4.3. Predecir el cantidad de crímenes en Londres.

Para el data set relacionado a la cantidad de crímenes en Londres, pudimos observar que las correlaciones son bastante bajas, ninguna llega ni al

1 % entre la variable de salida **value** (ver Figura 9), por lo que eso nos puede llegar a afectar bastante en los resultados finales. A continuación, se muestra la matriz de correlación:

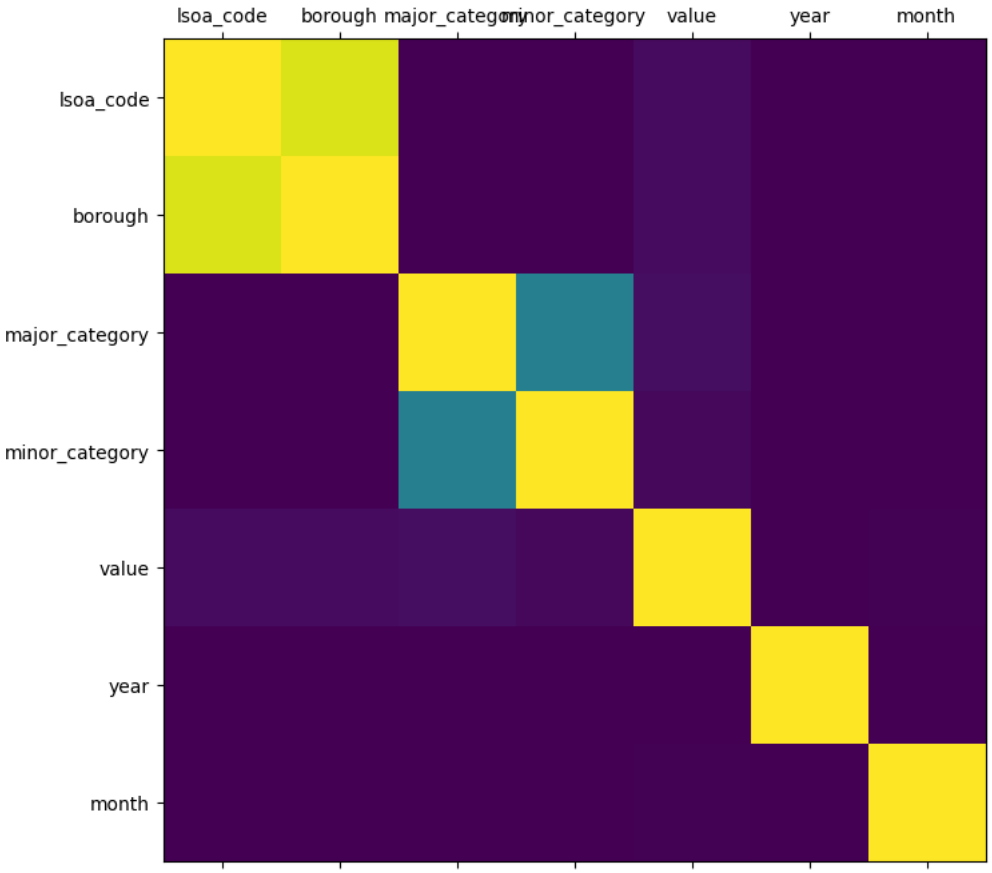

Figura 37: Matriz de correlación entre value
123

Para el algoritmo de **Regresión Lineal Simple**, los resultados obtenidos fueron los siguientes:

- Mean squared error: 3.06.
- Varianza: 0.00.
- Score: 0.00.

Para el algoritmo de **Regresión Lineal Múltiple**, los resultados obtenidos fueron los siguientes:

- Score: 0.002.

Como podemos observar el modelo predice bastante mal, esto debido a que la variable a predecir llamada value, no tiene una fuerte correlación con ninguna de las características. Prácticamente la mayor o mejor correlación es **major category** con 0.035985.

4.4. Clasificar si un cliente abandona un servicio.

Observando la Figura 13, podemos analizar que para la variable a predecir **Churn** el resto de características tienen una correlación muy baja:

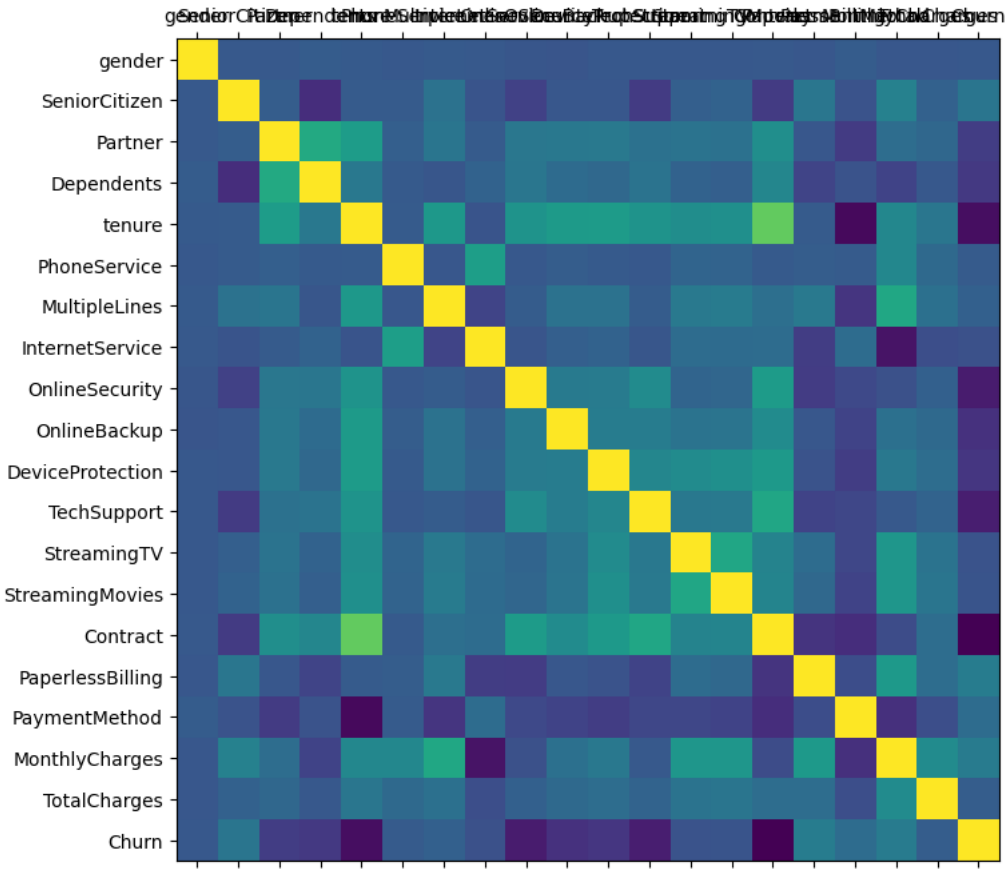

Figura 38: Matriz de correlación entre Churn

En esta ocasión se aplicaron Dos algoritmos de Machine Learning, para compararlos y ver cuál de estos se logra adaptar de mejor manera al set de datos. Para el algoritmo de **Máquinas de Soporte Vectorial**, aplicando el Kernel Gaussiano los resultados obtenidos fueron los siguientes:

- Score: 0.84.

Para el algoritmo de **Árbol de Decisión**, con una profundidad máxima de 20, los resultados obtenidos fueron los siguientes:

- Score: 0.80.

Utilizando el algoritmo de máquinas de soporte vectorial haciendo uso del kernel gaussiano, se obtuvo un F1 Score de 84 %, mientras que aplicandolo en Arboles de decisión se obtuvo un F1 Score del 80 %. Lo que quiere decir que el algoritmo de máquinas de soporte vectorial parece predecir mejor para este conjunto de datos.

4.5. Clasificar accidente cerebrovascular.

Observando la Figura 17, podemos analizar que para la variable a predecir **stroke** el resto de características tienen una correlación baja:

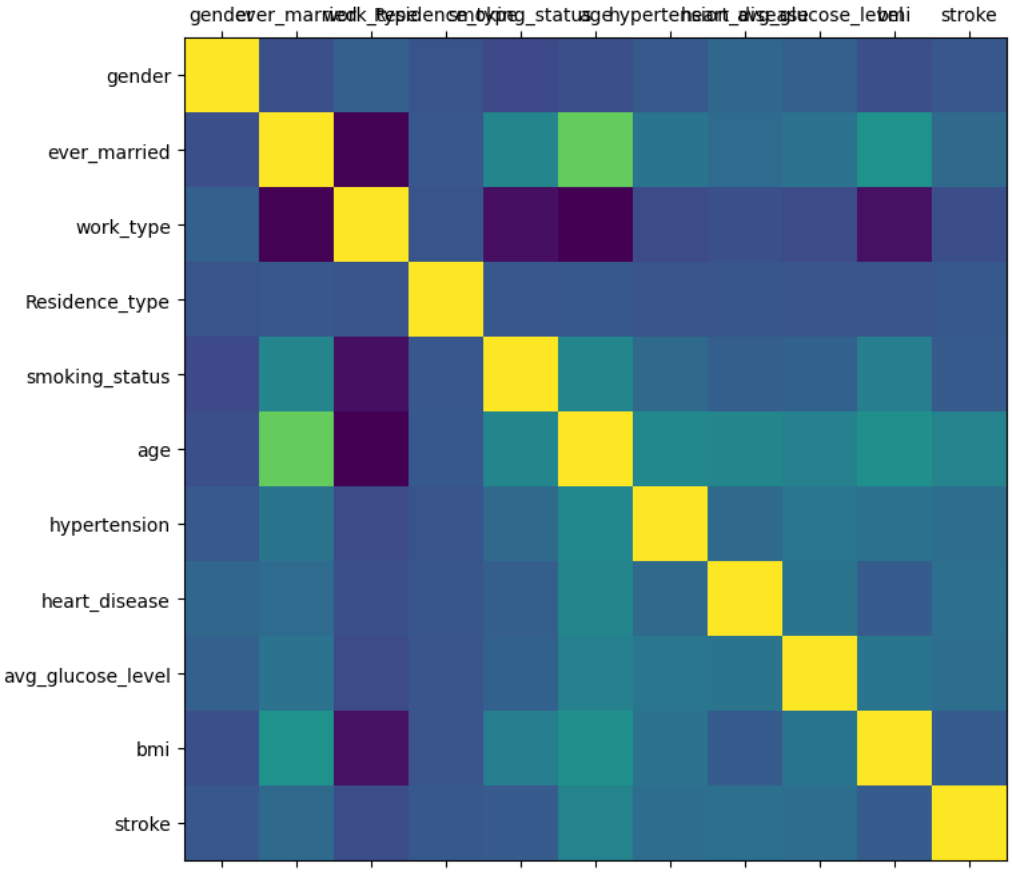

Figura 39: Matriz de correlación entre stroke

Para el algoritmo de **Árbol de Decisión**, con una profundidad máxima de 20, los resultados obtenidos fueron los siguientes:

- F1 Score: 0.95.

Podemos ver que el algoritmo de árbol de decisión para este conjunto de datos presenta una Precisión del 94 %, también su Recall es de 95 % al igual que si F1 Score que es de 95 %. Podemos ver por medio de la matriz de decisión que predice bastante bien para los casos en donde no se tiene un ataque cerebrovascular, pero para los casos en los que sí se tiene no lo hace tan bien, esto puede ser debido a la gran diferencia entre los datos de salida en donde se tenía más de 4000 datos que tenían la etiqueta de No y menor de 500 datos en los que se tenía la etiqueta de sí.

4.6. Clasificar tipo hepatitis.

A continuación, se mostrará la matriz de correlación entre la variable **Category**:

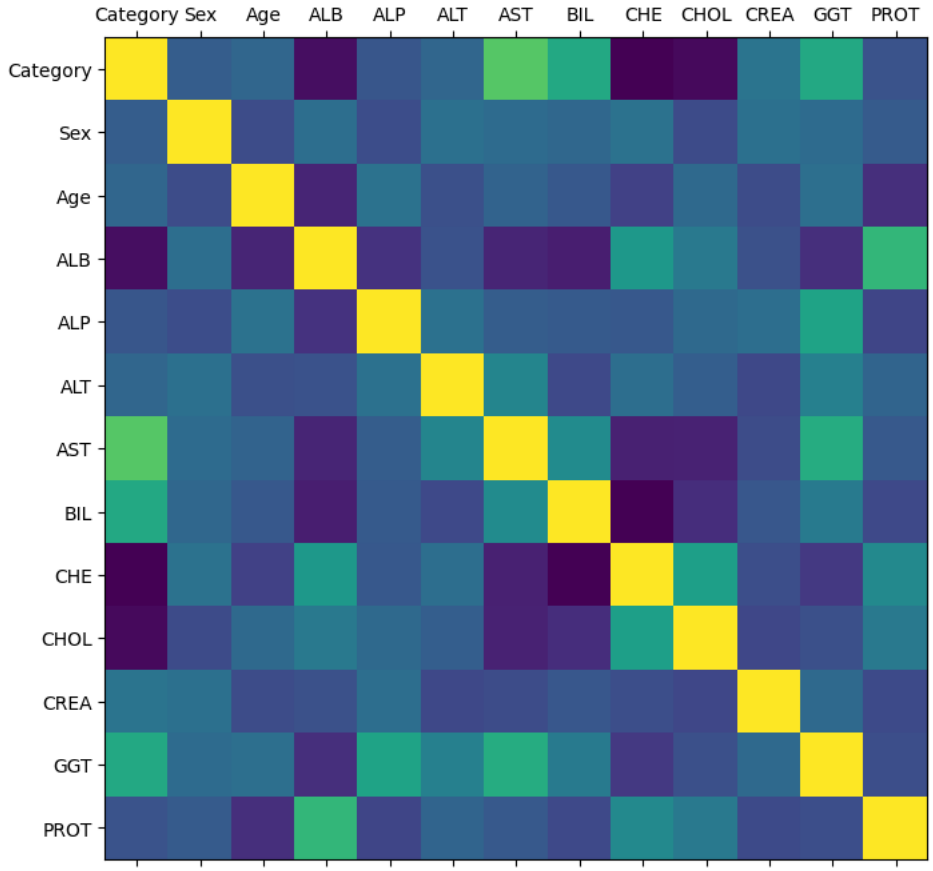

Figura 40: Matriz de correlación entre Category

A simple vista, vemos que para Category y el resto de características existe entre ellas poca correlación.

Para el algoritmo de **Árbol de Decisión**, con una profundidad máxima de 20, los resultados obtenidos fueron los siguientes:

- F1 Score: 0.86.

Podemos ver que el algoritmo de árbol de decisión para este conjunto de datos presenta una Precisión del 87 %, también su Recall es de 87 % al igual que si F1 Score que es de 86 %. Podemos ver por medio de la matriz de confusión que clasifica bastante bien para los casos en donde es donador de sangre, pero para los casos de hepatitis y el resto de opciones da resultados no tan buenos, esto puede ser a la gran desigualdad de las muestras en donde claramente la primera muestra se tienen muchos datos mientras que para el resto no.

4.7. Predecir masa corporal.

Para el data set relacionado al porcentaje de grasa corporal, las correlaciones mostradas en la Figura 24 nos muestra una correlación positiva del 0.81 %, otra del 0.70 %, también una correlación negativa bastante fuerte del -0.98 % que es la característica **Density**. Todas estas correlaciones se comparan con la variable de salida **BodyFat** por lo que a primera impresión podemos ir deduciendo que probablemente obtengamos buenos resultados:

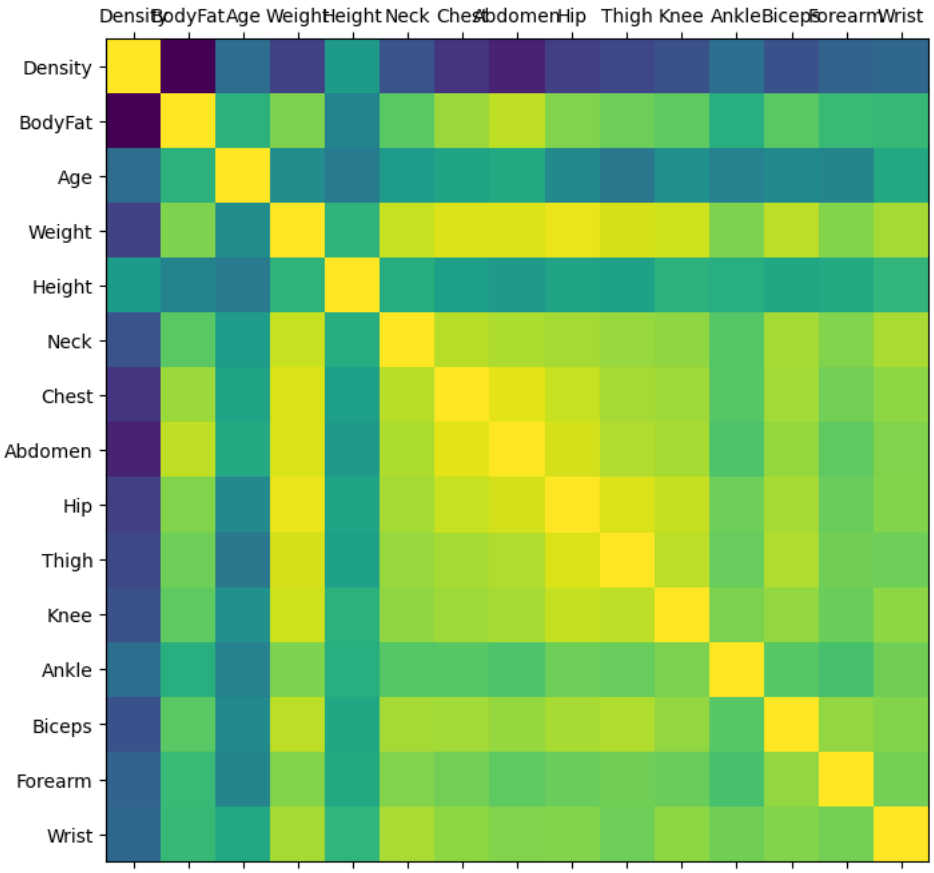

Figura 41: Matriz de correlación entre BodyFat

Para el algoritmo de **Regresión Lineal Simple**, los resultados obtenidos fueron los siguientes:

- Mean squared error: 0.07.
- Varianza: 1.00.

- Score: 0.99.

Para el algoritmo de **Regresión Lineal Múltiple**, los resultados obtenidos fueron los siguientes:

- Score: 0.99.

Para este ejemplo, el modelo predice un 99 % de la variabilidad de los datos. Utilizando la regresión lineal múltiple o regresión lineal simple.

4.8. Clasificar el estado de la Cirrosis.

A continuación, se mostrará la matriz de correlación del presente conjunto de datos:

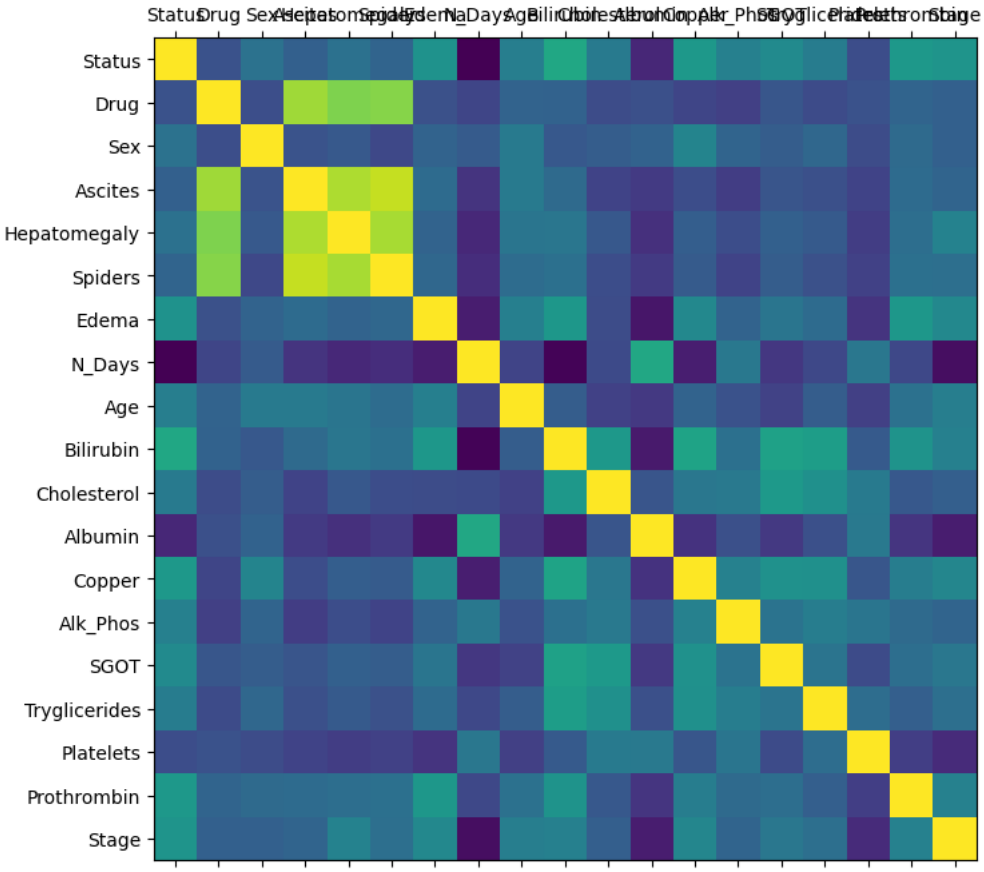

Figura 42: Matriz de correlación

La variable a clasificar es Status, pero analizando visualmente la matriz de correlación, se puede observar que no existe ninguna característica que tenga una correlación fuerte con la variable Status.

Para este set de datos, se realizaron varias pruebas con diferentes tipos de algoritmos y sus variantes, por lo que a continuación, se mostrarán los resultados obtenidos con cada uno de estos algoritmos:

Para el algoritmo de **Máquina de Soporte Vectorial Polinomial**, , los resultados obtenidos fueron los siguientes:

- F1 Score: 0.5.

Para el algoritmo de **Máquina de Soporte Vectorial aplicando Kernel Gaussiano**, aplicando escalado de datos, los resultados obtenidos fueron los siguientes:

- F1 Score: 0.47.

Para el algoritmo de **Árbol de Decisión**, sin datos escalados con una profundidad máxima de 20, los resultados obtenidos fueron los siguientes:

- F1 Score: 0.42.

Para el algoritmo de **Árbol de Decisión**, con datos escalados y una profundidad máxima de 20, los resultados obtenidos fueron los siguientes:

- F1 Score: 0.42.

Para el algoritmo de **Random Forest Sin escalar**, con 100 estimadores, los resultados obtenidos fueron los siguientes:

- F1 Score: 0.53.

Para el algoritmo de **Random Forest con escalado**, con 100 estimadores, los resultados obtenidos fueron los siguientes:

- F1 Score: 0.54.

Después de probarlo con distintos algoritmos de clasificación, entre estos SVM Polinomial, SVM utilizando Kernel Gaussiano, Árbol de decisión y Random Forest, también aplicando escalado por medio de StandarScaler o RobustEscaler, se llegó a la conclusión que los mejores resultados se obtuvieron por medio del algoritmo de Random Forest, aplicando la técnica de escalado StandarScaler obteniendo así una precisión del 54 %. Sin embargo, estos resultados para este set de datos no son buenos.

4.9. Predecir el precio del aguacate.

Las correlaciones mostradas en la Figura 30 que la variable con mejor correlación es **type** sin embargo su correlación es del 0.61 porciento respecto a la variable de salida **AveragePrice**. A continuación se mostrará la Matriz de Correlación para este set de datos, si observamos vemos que para la variable a predecir no se tiene ninguna correlación fuerte, la más cercana es type, pero se logra observar de color amarillo que hay una correlación grande entre las variables Total Value, 4046, 4225, 477., Total Bags, Small Bags, LArge Bags, XLarge Bags:

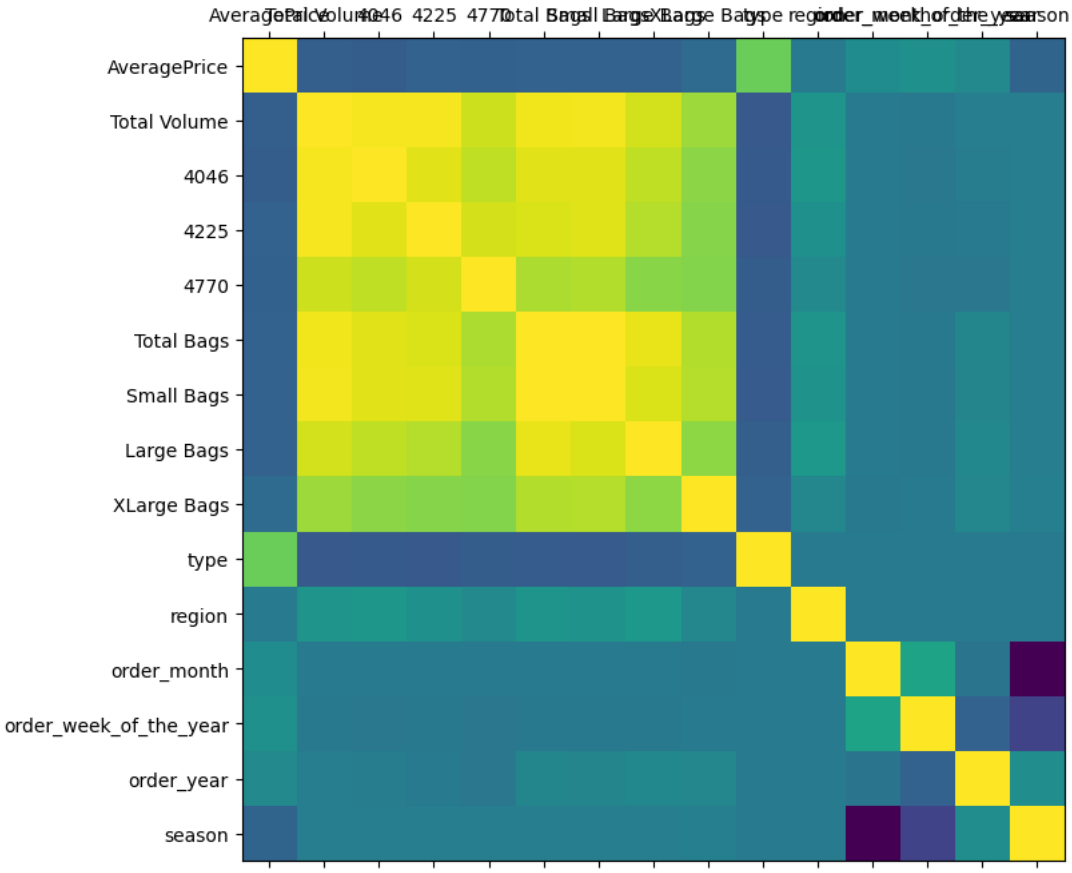

Figura 43: Matriz de correlación entre AveragePrice

Para el algoritmo de **Regresión Lineal Múltiple**, los resultados obtenidos fueron los siguientes:

- Score: 0.37.

En esta primera versión del modelo nos ha quedado un modelo bastante malo con solo un 37 % de presición. Aplicar escalado a los datos tal vez podría mejorar un poco el score, o también cambiar el algoritmo de Machine Learning.

4.10. Predecir las ventas de una compañia.

En la Figura 32 podemos observar que existe una correlación positiva bastante fuerte entre **Sales** y **Customers**. A continuación, se mostrará la Matriz de Correlación visual:

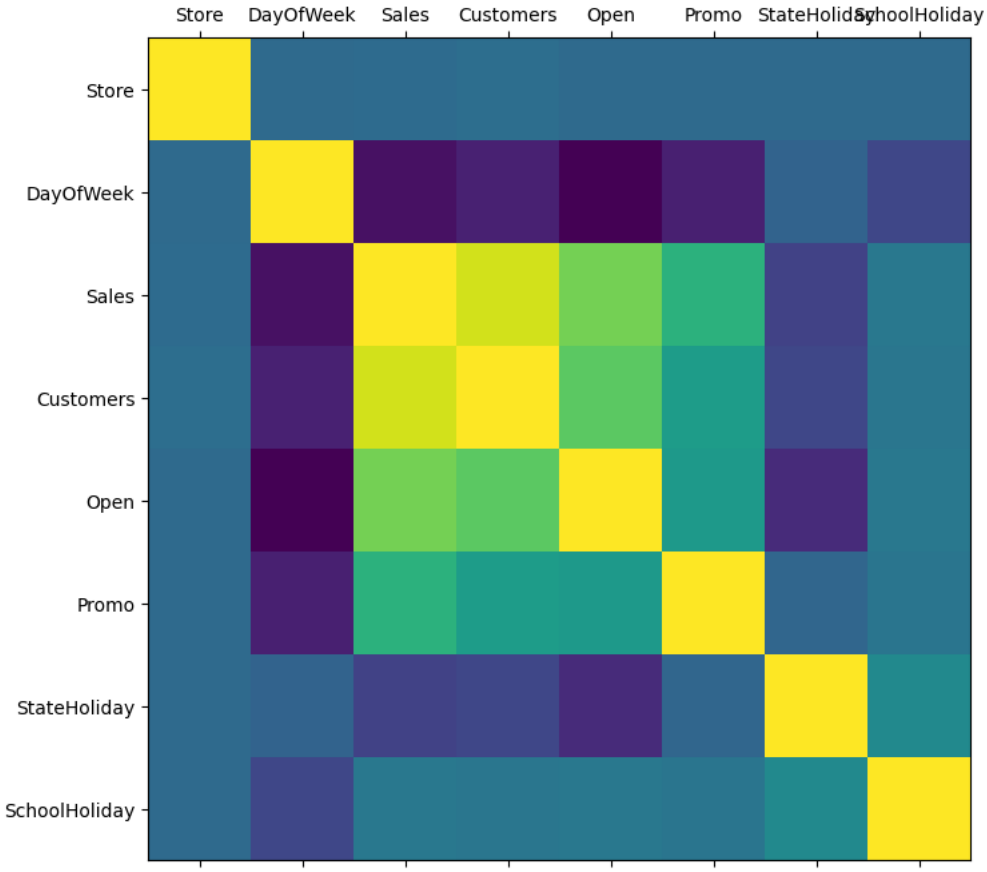

Figura 44: Matriz de correlación entre todos los atributos

Para el algoritmo de **Regresión Lineal Simple**, los resultados obtenidos fueron los siguientes:

- Varianza: 0.80.

- Score: 0.79.

Para el algoritmo de **Regresión Lineal Múltiple**, los resultados obtenidos fueron los siguientes:

- Score: 0.85.

El presente conjunto de datos utilizando la regresión lineal simple con la característica Customers, presenta una varianza del 80 %, lo cuál dependiendo de qué tan rigurosos sean puede ser aceptable. Aplicando regresión lineal múltiple con las características de Customers, Open y Promo, el modelo se comporta de mejor manera ya que el score sube 5 % más quedando una presición del modelo en un 85

Referencias

[1] Trevor Hastie Gareth James, Daniela Witten. *An Introduction to Statistical Learning: With Applications in Python.* Springer, 2023.

[2] John R. Koza, Forrest H. Bennett, David Andre, and Martin A. Keane. Automated design of both the topology and sizing of analog electrical circuits using genetic programming. In *Artificial Intelligence in Design '96*, pages 151–170, 1996.

[3] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning.* MIT Press, 2012.

[4] Sebastian Rashka and Vahid Mirdzhalili. *Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2.* Packt, Birmingham, Mumbai, 2020.

Anexos

Repositorio de información

A continuación, encontrará la dirección del repositorio en GitHub donde se encuentra almacenado el proyecto. El nombre del repositorio es *minerva-personal-assistant* y el enlace URL al repositorio en línea es: `https://github.com/andreymch22/minerva-personal-assistant.git`