Tecnológico de Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos (Gpo 102)

Curso:
TC3006C.102

Campus:
Monterrey

Limpieza del Conjunto de Datos

Equipo 2:

Rafhael Eduardo Chavez Ramirez | A00832228

Jose David de la Garza Salas | A00834760

Pablo Andrés Martínez Sánchez | A01252489

Andre Sebastian Galindo Posadas | A00833376

Daniel Sánchez Villarreal | A01197699

Lugar y Fecha:

Monterrey, Nuevo León

19 de Agosto del 2024

Índice

Introducción

Extracción de Datos

```
print(df_gender.shape)
df_gender.head()
```

```
(418, 2)
```

	PassengerId	Survived
0	892	0
1	893	1
2	894	0
3	895	0
4	896	1

Tabla 1: En esta es la tabla del archivo gender_submission.csv y se enseña por cuantas columnas y filas está hecha

```
print(df_train.shape)
df_train.head()
```

```
(891, 12)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

Tabla 2: En esta es la tabla del archivo train.csv y se enseña por cuantas columnas y filas está hecha

```
print(df_test.shape)
df_test.head()
```

```
(418, 11)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

Tabla 3: En esta es la tabla del archivo test.csv y se enseña por cuantas columnas y filas está hecha

Lo que se hizo primero fue leer los archivos que se descargaron del reto del Titanic y luego de entender porque información estaba compuesta, se empezó a hacer el análisis correspondiente para saber qué datos nos podría servir para el modelo. La primera forma en la que limpiamos los datos fue ver cuáles eran las columnas que más datos vacíos tenían (NaN).

```
count_na = df_train.isna().sum()
print('Cantidad de valores NaN en datos de entrenamiento:\n', count_na)
```

```
Cantidad de valores NaN en datos de entrenamiento:
 PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

Tabla 4: Se muestran todos los datos vacíos (NaN), que tiene cada columna del archivo train.csv

En la tabla se observó que los datos de la cabina y de la edad son los que presentan más información faltante. En el caso de la cabina son 687 datos faltantes de 891 datos, luego de realizar una sencilla regla de tres, nos percatamos de que cerca del 77% de los datos estaban ausentes. Debido a esta alta proporción de datos faltantes, decidimos eliminar esa columna.

En la columna de edad, hay 177 valores faltantes de un total de 891, lo que equivale al 19.86% de los datos en esa columna. Se consideraron dos cursos de acción para

abordar este problema: imputación de datos o eliminación. Tras evaluar el modelo en ambos escenarios, se escogerá el que tenga mejor precisión. Para la imputación, utilizamos la mediana, agrupando los datos por clase y sexo, y así obtener la mediana de cada grupo. Esta se usó para llenar los datos faltantes, con cerca de seis combinaciones diferentes. Este método se consideró adecuado debido a la menor proporción de información faltante en esta columna.

Además, identificamos que hay información que no aporta mucho al modelo, como el PassengerID, que solo indica cómo se almacenó al pasajero en la base de datos, pero no ofrece información relevante para el modelo, ya que los pasajeros ya están ordenados con un número respectivo. Otro dato eliminado fue el Ticket, ya que no proporciona información sobre la supervivencia de los pasajeros. El último dato que se eliminó fue el Fare, dado que tampoco brinda información relevante sobre los pasajeros que sobrevivieron o no al accidente del Titanic.

```
[ ] df_train = df_train.drop(['Ticket', 'Cabin', 'PassengerId', 'Fare'], axis=1)
```

```
print(df_train.shape)
df_train.head()
```

```
(891, 8)
```

	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Embarked
0	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	S
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	C
2	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	S
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	S
4	0	3	Allen, Mr. William Henry	male	35.0	0	0	S

Tabla 5: Se muestran los datos luego de la limpieza en el archivo train.cvs y test.csv

Una pequeña modificación que se hizo al archivo fue el hecho de cambiar los géneros por valores booleanos, esto para que sea más sencillo de analizar para el modelo que queremos crear, por lo que hicimos una nueva asignación donde los valores de 0 es para el sexo masculino, mientras que los valores de 1 son para las mujeres y eliminamos la columna de sexo normal, de esta manera la mayoría de datos que analiza el sistema son númericos. Luego de todas las modificaciones la tabla final queda de la siguiente manera

```
df_train.head()
```

	Survived	Pclass	Name	Age	SibSp	Parch	Embarked	Sex bool
0	0	3	Braund, Mr. Owen Harris	22.0	1	0	S	0
1	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	38.0	1	0	C	1
2	1	3	Heikkinen, Miss. Laina	26.0	0	0	S	1
3	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	35.0	1	0	S	1
4	0	3	Allen, Mr. William Henry	35.0	0	0	S	0

Tabla 6: En esta tabla se ven los datos finales y sin nada redundante en los archivos de train.csv y test.csv