Presentación Proyecto Aplicación de la Matemática a la Ingeniería

Spaceship Titanic

Nicolás Stagnaro

Universidad Técnica Federico Santa María

Domingo 3 de Diciembre 2023

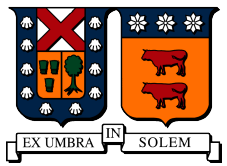


Tabla de Contenidos

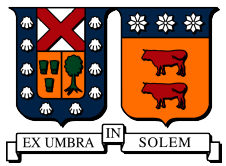
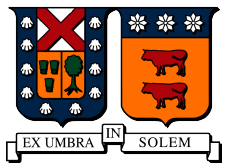
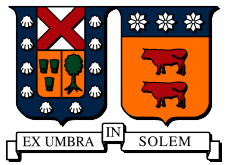


Tabla de Contenidos



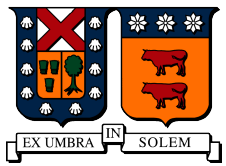
Motivación

La finalidad de este proyecto es enfrentar a los estudiantes a problemas de Machine Learning con todas las etapas (típicas) que eso implica, basándose en cada uno de los módulos aprendidos a lo largo del curso, dando pie a la investigación y a la solución de problemas operacionales del mundo real.



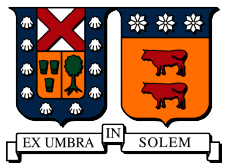
Descripción del Proyecto

La nave espacial Titanic fue un transatlántico interestelar de pasajeros lanzado hace un mes. Con casi 13.000 pasajeros a bordo, la nave emprendió su viaje inaugural transportando emigrantes de nuestro sistema solar a tres exoplanetas recientemente habitables que orbitan estrellas cercanas.



Descripción del Proyecto

Mientras rodeaba Alpha Centauri en ruta hacia su primer destino, el tórrido 55 Cancri E, la incauta nave espacial Titanic chocó con una anomalía espacio-temporal oculta, dentro de una nube de polvo. Lamentablemente, corrió un destino similar al de su homónimo de 1000 años antes. Aunque el barco permaneció intacto, casi la mitad de los pasajeros fueron transportados a una dimensión alternativa.



Descripción del Proyecto

Para ayudar a los equipos de rescate y recuperar a los pasajeros perdidos, tienes el desafío de predecir qué pasajeros fueron transportados por la anomalía utilizando registros recuperados del sistema informático dañado de la nave espacial.

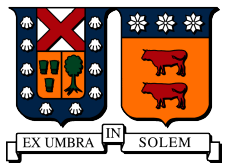
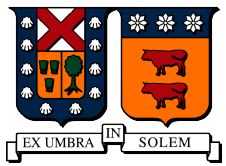
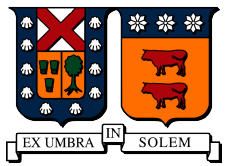


Tabla de Contenidos



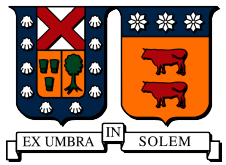
Estadística Descriptiva y Visualización

En primer lugar trabajamos con el 'train.csv', guardándolo en el dataframe 'train_df' y lo limpiamos de los valores nan, reemplazando los nan de las variables categóricas por la moda y los nan de las variables continuas con la media, como se ve en el siguiente código



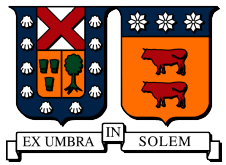
Estadística Descriptiva y Visualización

```
1 cat_predictors = ['HomePlanet', 'CryoSleep', 'Cabin', '
      Destination', 'VIP', 'Name', 'Transported']
2 for col in cat_predictors:
3     train_df[col] = train_df[col].fillna(train_df[col].mode()
      [0])
4 train_df = train_df.fillna(train_df.mean())
```

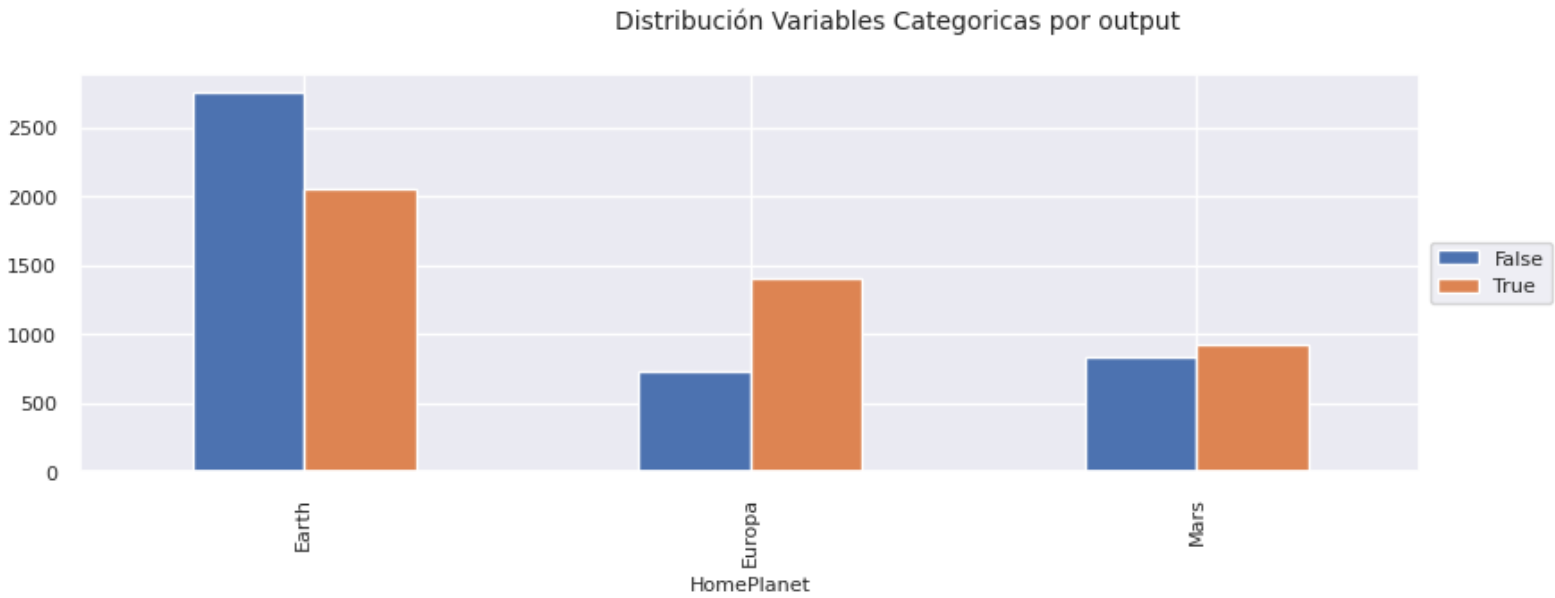


Estadística Descriptiva y Visualización

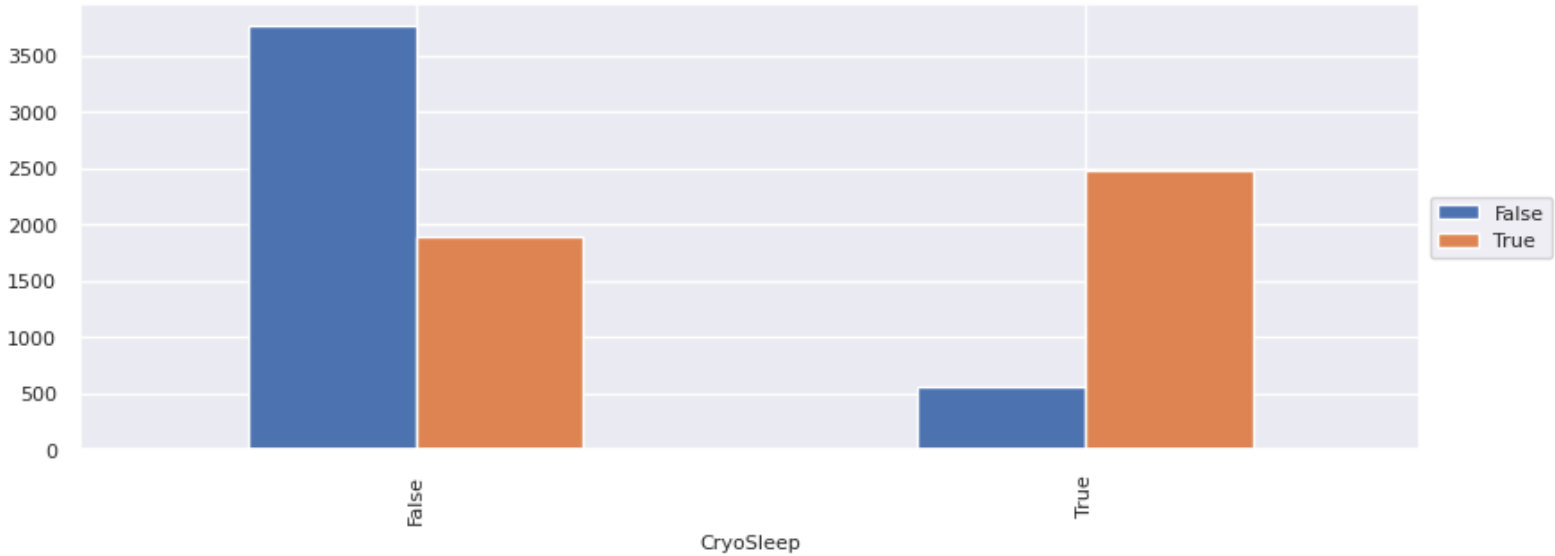
Luego se procedió a hacer boxplots para analizar si había outliers o no para las variables continuas, obteniendo que si tenían. Además de analizar la distribución, tanto de las variables continuas como de las categóricas, con respecto al output que queremos predecir.



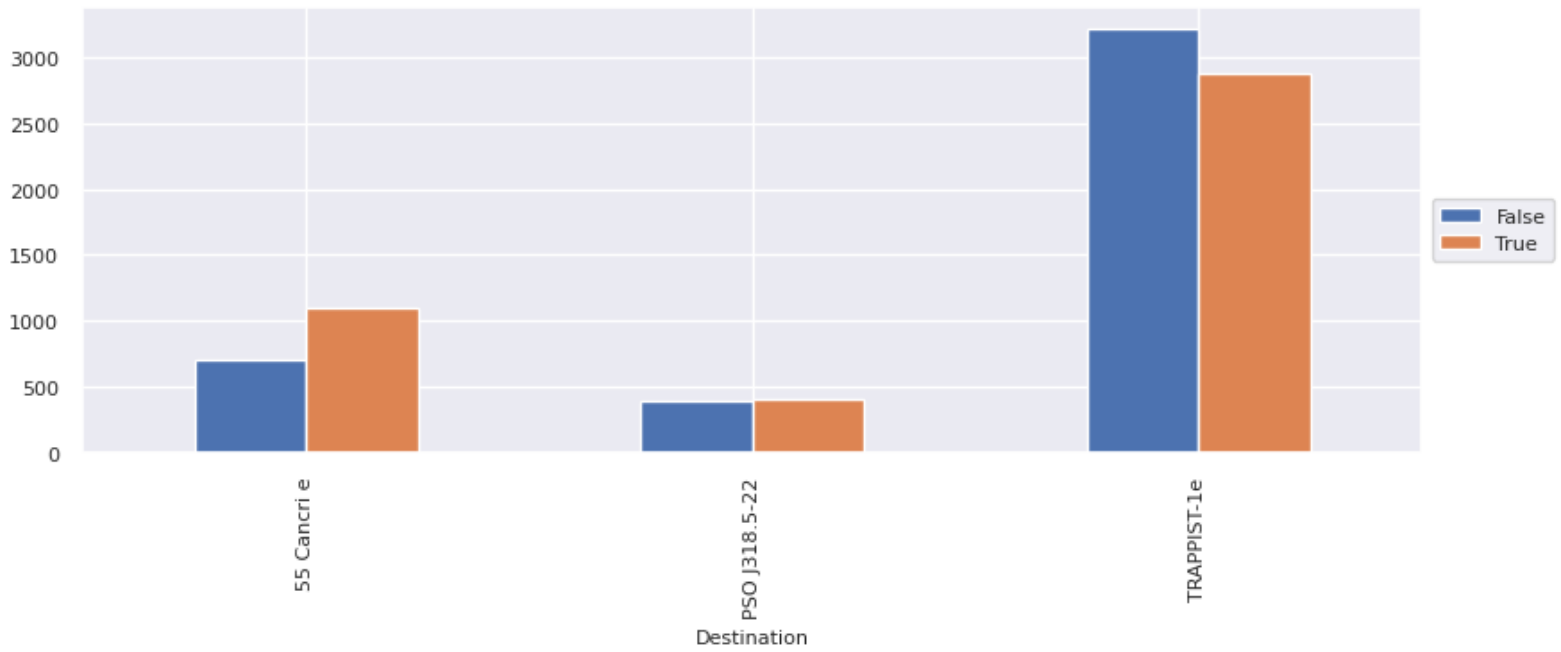
Estadística Descriptiva y Visualización



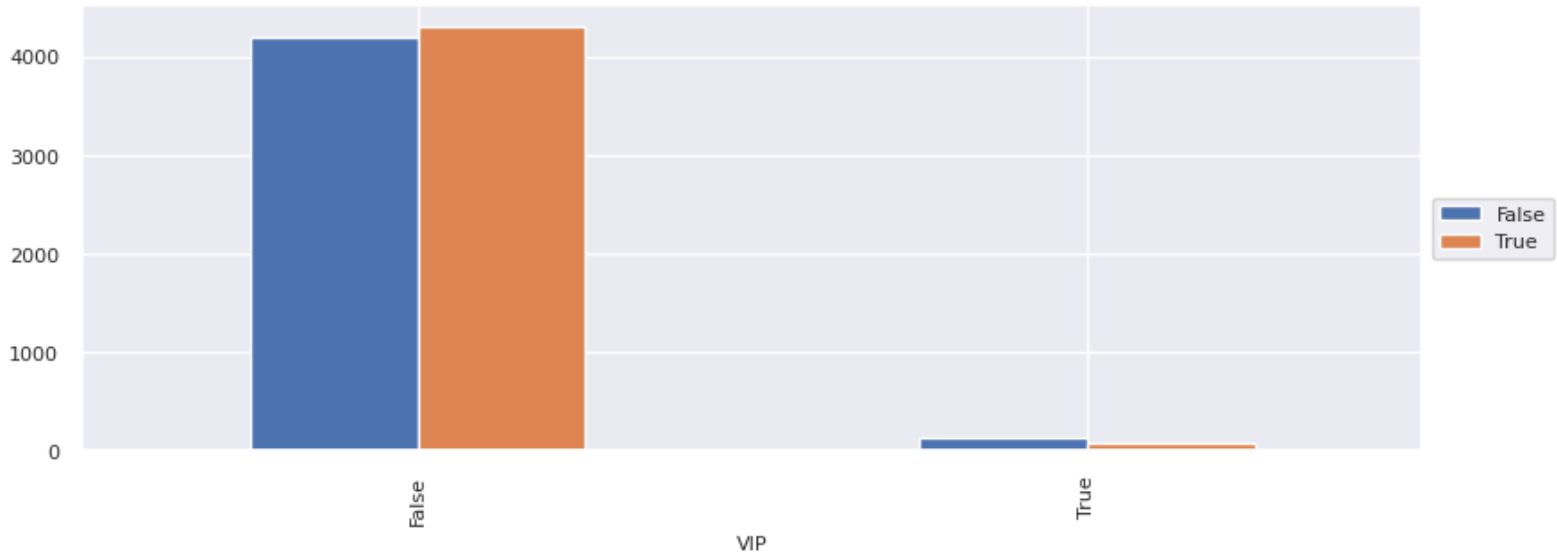
Estadística Descriptiva y Visualización



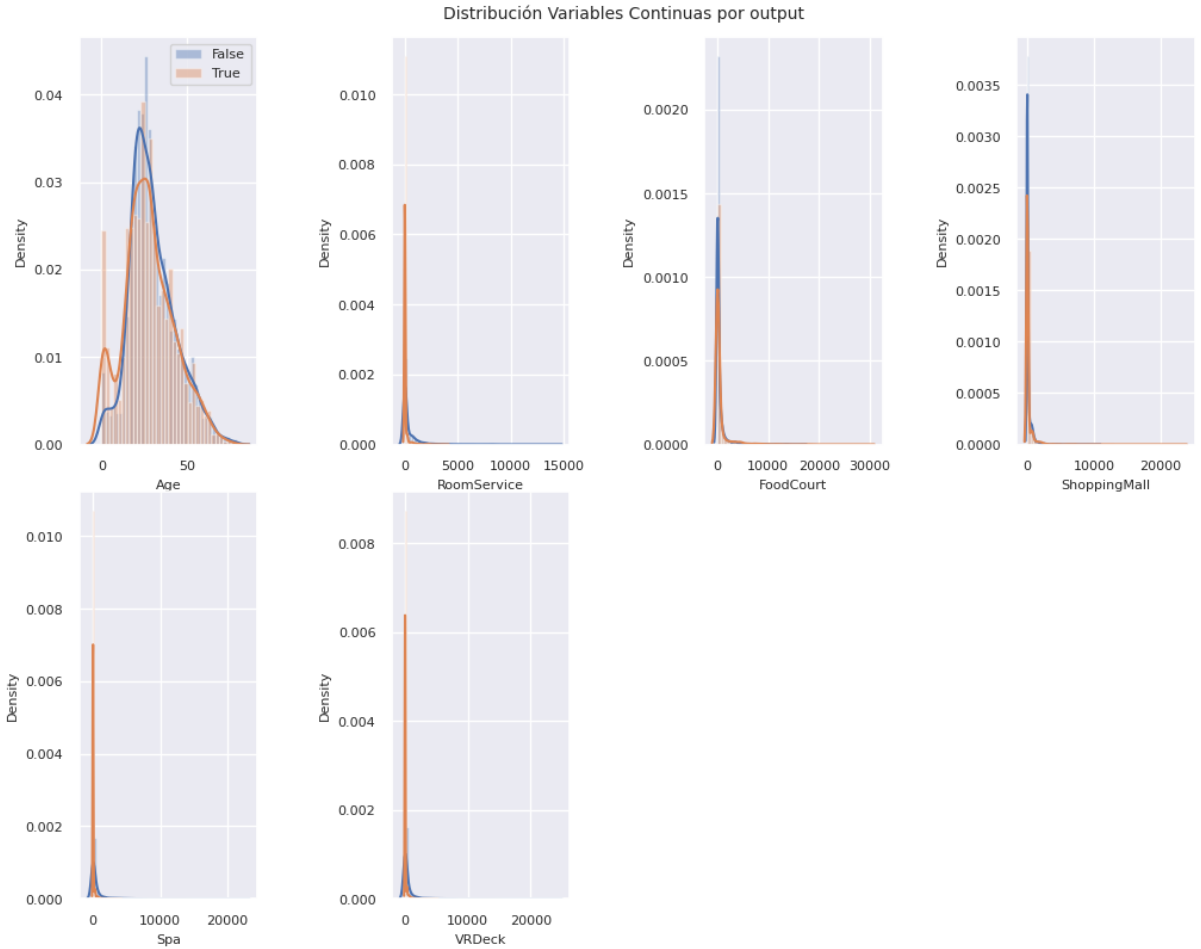
Estadística Descriptiva y Visualización



Estadística Descriptiva y Visualización

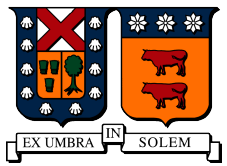


Estadística Descriptiva y Visualización



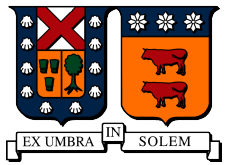
Preprocesamiento

Considerando lo realizado en la parte anterior se procede a normalizar los atributos y a separar el conjunto de entrenamiento con el de testeo como en el siguiente código



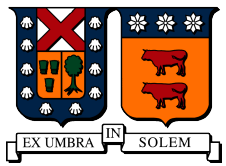
Preprocesamiento

```
from sklearn.preprocessing import RobustScaler
X1 = train_df.iloc[:, [4,6,7,8]]
scaler = RobustScaler().fit(X1)
X1_scaler = scaler.transform(X1)

from sklearn.preprocessing import OrdinalEncoder
X_2 = train_df.iloc[:, [3,5]]
ohe = OrdinalEncoder()
feature_arr = np.array(ohe.fit_transform(X_2))
X = np.concatenate((X1_scaler, feature_arr),axis=1)
data_aux = pd.DataFrame(X, columns = ["Age", "RoomService", "
    FoodCourt", "ShoppingMall", "Destination", "VIP"])
```



Preprocesamiento

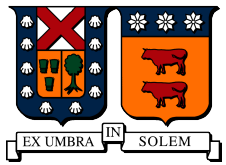
```
from sklearn.preprocessing import LabelEncoder
y = train_df.iloc[:, -1].values

le = LabelEncoder()
y = le.fit_transform(y)

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, shuffle=False)
```



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

Se seleccionaron los siguientes modelos para trabajar, donde todos poseeen hiperparámetros.

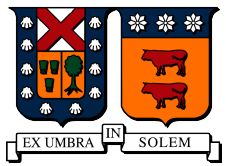
```
# Modelos
knn_model = KNeighborsClassifier()
rf_model = RandomForestClassifier()
logistic_model = LogisticRegression()
perceptron_model = Perceptron()
```



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

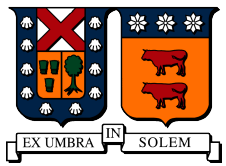
Para la optimización de los hiperparámetros se usó GridSearchCV de la siguiente manera para cada modelo:

```
# KNN
# Hiperparametros para KNN

parameters = {
        'n_neighbors':[1, 3, 5], 'weights':['uniform', 'distance
    ']}

grid_search = GridSearchCV(knn_model, parameters, cv=3, scoring=
    "accuracy", return_train_score=True, verbose=10)
grid_search.fit(X_train, y_train)
```



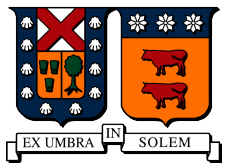
Selección de Modelos, Métricas, Análisis de Resultados y Visualización

```
# Random Forest
# Hiperparametros para Random Forest

parameters = {
        'n_estimators':[10, 15, 20], 'max_features':['log2', '
    sqrt']}

grid_search = GridSearchCV(rf_model, parameters, cv=3, scoring="
    accuracy", return_train_score=True, verbose=10)
grid_search.fit(X_train, y_train)
```



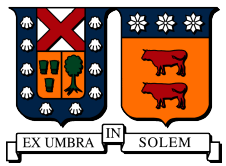
Selección de Modelos, Métricas, Análisis de Resultados y Visualización

```
# Regresor Logistico
# Hiperparametros para Regresor Logistico

parameters = {
        'C':[1.0, 10, 100], 'penalty':['l1', 'l2']}

grid_search = GridSearchCV(logistic_model, parameters, cv=3,
    scoring="accuracy", return_train_score=True, verbose=10)
grid_search.fit(X_train, y_train)
```



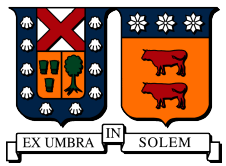
Selección de Modelos, Métricas, Análisis de Resultados y Visualización

```
# Perceptron
# Hiperparametros para Perceptron

parameters = {
        'eta0':[0.01, 0.1, 1.0], 'penalty':['l1', 'l2']}

grid_search = GridSearchCV(perceptron_model, parameters, cv=3,
    scoring="accuracy", return_train_score=True, verbose=10)
grid_search.fit(X_train, y_train)
```



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

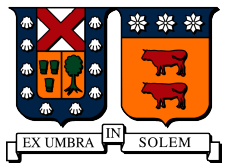
Mediante el comando 'grid_search.best_estimator_' obtuvimos el mejor modelo de cada uno de los modelos anteriores. Analizando un poco más en profundidad cada uno de estos modelos con sus respectivas métricas obtuvimos los siguientes classification reports



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

Para el KNN

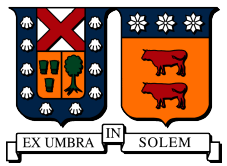
	precision	recall	f1-score	support
0	0.72	0.65	0.68	911
1	0.65	0.73	0.69	828
accuracy			0.69	1739
macro avg	0.69	0.69	0.69	1739
weighted avg	0.69	0.96	0.69	1739



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

Para el Random Forest

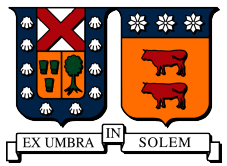
	precision	recall	f1-score	support
0	0.76	0.67	0.71	911
1	0.68	0.77	0.72	828
accuracy			0.72	1739
macro avg	0.72	0.72	0.72	1739
weighted avg	0.72	0.72	0.72	1739



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

Para el Regresor Logístico

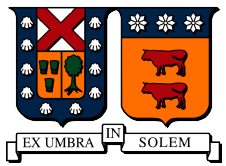
	precision	recall	f1-score	support
0	0.76	0.38	0.50	911
1	0.56	0.87	0.68	828
accuracy			0.61	1739
macro avg	0.66	0.62	0.59	1739
weighted avg	0.67	0.61	0.59	1739



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

Para el Perceptron

	precision	recall	f1-score	support
0	0.49	0.70	0.57	911
1	0.36	0.19	0.25	828
accuracy			0.46	1739
macro avg	0.42	0.44	0.41	1739
weighted avg	0.43	0.46	0.42	1739



Selección de Modelos, Métricas, Análisis de Resultados y Visualización

De donde concluímos que el mejor modelo de los cuatro es el de Random Forest, que es el que se usará para hacer las predicciones de los datos de 'test.csv'.

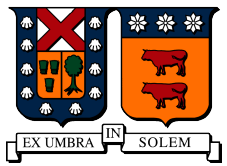
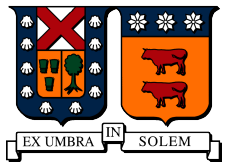


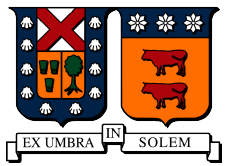
Tabla de Contenidos



Conclusiones

Se concluye finalmente que resolver problemas de machine learning como este requiere de un análisis profundo de las variables tanto continuas como categóricas, sus relaciones con el output a predecir (tanto en problemas de clasificación como regresión), además de un buen preprocesamiento para usar de buena manera los modelos.

Agregar también la importancia de la optimización de hiperparámetros, la cual sirve para discriminar entre modelos y así escoger el que realice las mejores predicciones posibles.



Gracias por su Atención