Proyecto de semestre - Entrega 2

UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Materia
Introducción a la inteligencia artificial

Docente
Raul Ramos Pollan

Estudiantes
Hernán Javier Aguilar Cruz
Jhonier Raúl Jiménez
Acevedo

Universidad de Antioquia
Medellín
2023

Avances

En esta segunda entrega del proyecto de semestre, se trabaja con el dataset, de tal manera que pueda ser utilizado de manera correcta para ejecutarse con distintos modelos, y un porcentaje de predicción lo suficientemente bueno.

Primero se carga el dataset desde el link del repositorio de github:

```
[ ] url = 'https://raw.githubusercontent.com/HernanAC/TaiwaneseBankruptcyPrediction/main/data.csv'
    data = pd.read_csv(url)
```

Cargando el dataset

```
[ ] data.head()
```

	Bankrupt?	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate
0	1	0.370594	0.424389	0.405750	0.601457	0.601457	0.998969	0.796887	0.808809
1	1	0.464291	0.538214	0.516730	0.610235	0.610235	0.998946	0.797380	0.809301
2	1	0.426071	0.499019	0.472295	0.601450	0.601364	0.998857	0.796403	0.808388
3	1	0.399844	0.451265	0.457733	0.583541	0.583541	0.998700	0.796967	0.808966
4	1	0.465022	0.538432	0.522298	0.598783	0.598783	0.998973	0.797366	0.809304

5 rows × 96 columns

Una vez se tienen los datos cargados, procedemos a analizarlos.

```
[ ] data.isnull().sum()
```

```
Bankrupt?                                                  0
 ROA(C) before interest and depreciation before interest   0
 ROA(A) before interest and % after tax                    0
 ROA(B) before interest and depreciation after tax         0
 Operating Gross Margin                                    0
                                                          ..
 Liability to Equity                                       0
 Degree of Financial Leverage (DFL)                        0
 Interest Coverage Ratio (Interest expense to EBIT)        0
 Net Income Flag                                           0
 Equity to Liability                                       0
Length: 96, dtype: int64
```

```
[ ] numeric_features = data.dtypes[data.dtypes != 'int64'].index
    categorical_features = data.dtypes[data.dtypes == 'int64'].index

    data[categorical_features].columns.tolist()
```

```
['Bankrupt?', ' Liability-Assets Flag', ' Net Income Flag']
```

Podemos ver que el dataset cumple con el tamaño establecido en los estándares del proyecto, sin embargo, no cuenta con nulos y sólo tiene 3 columnas categóricas (menos del 10%, ya que son 96 en total).

Exploración de datos

El resultado a predecir es si se entra a bancarrota o no, este está dado por la columna Bankruptcy. Como podemos ver, está altamente desbalanceado por lo que se requiere de hacer unas modificaciones para que esto no sea así.

```
[ ]  data['Bankrupt?'].value_counts()

     0    6599
     1     220
     Name: Bankrupt?, dtype: int64
```

Se propone utilizar una técnica de oversampling llamada SMOTE para ayudarnos a mejorar el rendimiento de los datos al ser probados con modelos.

```
[ ]  X=data.drop(labels=['Bankrupt?'], axis=1)
     y=data['Bankrupt?']
     oversample = SMOTE()
     X,y=oversample.fit_resample(X,y)
     sns.countplot(x=y)
```

Ya que se tienen los datos que fueron pasados por el SMOTE, lo que queremos ahora es probar cómo funcionan los datos tal cual están ahora, sin cumplir los requisitos de las columnas categóricas, o el porcentaje de datos nulos.

```
[ ]  scaler=StandardScaler()
     X_scale=scaler.fit_transform(X)
     X_train, X_test, y_train, y_test = train_test_split(X_scale, y,test_size=0.3)
```

Una vez teniendo esto, entonces lo que sigue es probar distintos modelos. Para esta entrega se va a probar con regresión logística y SVM.

Para cada uno de estos, los resultados fueron buenos, y siguiendo las pautas que se habían escrito en la primera entrega, se puede decir que cumple con las expectativas de lo planteado anteriormente.

```
[ ]  log_reg.score(X_test,y_test)

     0.9093434343434343
```

```
[ ]  scaler=StandardScaler()
     X_scale=scaler.fit_transform(X)
     X_train, X_test, y_train, y_test = train_test_split(X_scale, y,test_size=0.3)
```

Ahora, estos modelos fueron desarrollados sólo de prueba y para verificar qué tan bueno son los modelos si se trabaja con los datos sin ser modificados de manera considerable. Para seguir con los requerimientos del proyecto, entonces el siguiente paso es modificar columnas para que por lo menos el 10% sean categóricas.

```
columns_to_modify = [data.columns[1], data.columns[2], data.columns[9],
                     ' Inventory Turnover Rate (times)',
                     ' Working Capital/Equity',
                     ' Contingent liabilities/Net worth']
columns_to_modify
```

```
[' ROA(C) before interest and depreciation before interest',
 ' ROA(A) before interest and % after tax',
 ' Non-industry income and expenditure/revenue',
 ' Inventory Turnover Rate (times)',
 ' Working Capital/Equity',
 ' Contingent liabilities/Net worth']
```

```
[6]  def modify_column(data, column_name):
         if column_name not in data.columns:
             print(f"{column_name} is not a valid column name.")
             return data
         else:
             col_idx = data.columns.get_loc(column_name)
             for i in range(len(data)):
                 if data.iloc[i, col_idx] < 0.33:
                     data.iloc[i, col_idx] = "LOW"
                 elif data.iloc[i, col_idx] < 0.66:
                     data.iloc[i, col_idx] = "MEDIUM"
                 else:
                     data.iloc[i, col_idx] = "HIGH"
             return data
```

```
[7]  for column_name in columns_to_modify:
       modify_column(data_copy, column_name)
     data_copy
```

En total se modifican 6 columnas, para tener un total de 9 columnas categóricas y poder cumplir con los requerimientos del proyecto. Para la entrega final entonces sería ideal tener una comparativa entre los modelos, con los datos modificados y el dataset que se utiliza al principio de esta entrega.

Para visualizar el link de la entrega 2 del proyecto, haga click aquí.