

Assignment on Classification technique

Every year many students give the GRE exam to get admission in foreign Universities. The data set contains GRE Scores (out of 340), TOEFL Scores (out of 120), University Rating (out of 5), Statement of Purpose strength (out of 5), Letter of Recommendation strength (out of 5), Undergraduate GPA (out of 10), Research Experience (0=no, 1=yes), Admitted (0=no, 1=yes). Admitted is the target variable.

The counselor of the firm is supposed check whether the student will get an admission or not based on his/her GRE score and Academic Score. So to help the counselor to take appropriate decisions build a machine learning model classifier using Decision tree to predict whether a student will get admission or not.

1. Apply Data pre-processing (Label Encoding, Data Transformation....) techniques if necessary.
2. Perform data-preparation (Train-Test Split)
3. Apply Machine Learning Algorithm
4. Evaluate Model.

Importing required libraries

```
In [3]: import pandas as pd
```

Reading the dataset

```
In [50]: A = pd.read_csv(r'C:\Users\DELL\Downloads\Admission_Predict.csv')
```

```
In [51]: A.head()
```

```
Out[51]:
```

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	1	337	118	4	4.5	4.5	9.65	1	0.92
1	2	324	107	4	4.0	4.5	8.87	1	0.76
2	3	316	104	3	3.0	3.5	8.00	1	0.72
3	4	322	110	3	3.5	2.5	8.67	1	0.80
4	5	314	103	2	2.0	3.0	8.21	0	0.65

```
In [52]: #dropping Serial No. column as it has no significance
A.drop('Serial No.',axis=1,inplace=True)
```

```
In [12]: A.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400 entries, 0 to 399
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -
0   GRE Score              400 non-null   int64
1   TOEFL Score            400 non-null   int64
2   University Rating      400 non-null   int64
3   SOP                    400 non-null   float64
4   LOR                    400 non-null   float64
5   CGPA                   400 non-null   float64
6   Research                400 non-null   int64
7   Chance of Admit        400 non-null   float64
dtypes: float64(4), int64(4)
memory usage: 25.1 KB
```

```
In [53]: # converting chance of admit column into binary values
A['Chance of Admit '] = A['Chance of Admit '].apply(lambda x : 1 if x >= 0.50 else
```

```
In [15]: A.head()
```

```
Out[15]:
```

	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	337	118	4	4.5	4.5	9.65	1	1
1	324	107	4	4.0	4.5	8.87	1	1
2	316	104	3	3.0	3.5	8.00	1	1
3	322	110	3	3.5	2.5	8.67	1	1
4	314	103	2	2.0	3.0	8.21	0	1

```
In [16]: A['Chance of Admit '].value_counts()
```

```
Out[16]:
```

1	367
0	33

Name: Chance of Admit , dtype: int64

Train test split

```
In [17]: X = A.drop('Chance of Admit ',axis=1)
```

```
In [18]: y = A['Chance of Admit ']
```

```
In [20]: from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,y,test_size=0.20,random_state=10)
```

```
In [21]: print(X_train.shape)
print(X_test.shape)
print(Y_train.shape)
print(Y_test.shape)
```

```
(320, 7)
(80, 7)
(320,)
(80,)
```

Training the model

```
In [27]: from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()
clf.fit(X_train,Y_train)
y_pred = clf.predict(X_test)
```

```
In [28]: y_pred
```

```
Out[28]: array([0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1,
        1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
        1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
        1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1]) dtype=int64)
```

```
In [29]: df = pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
```

```
In [30]: df.head()
```

```
Out[30]:
```

	Actual	Predicted
38	1	0
387	1	1
270	1	1
181	1	1
195	1	1

```
In [31]: clf.classes_
```

```
Out[31]: array([0, 1], dtype=int64)
```

```
In [32]: from sklearn.tree import plot_tree
```

```
In [33]: features = ['GRE Score', 'TOEFL Score', 'University Rating', 'SOP', 'LOR ', 'CGPA',
        'Research']
```

```
In [39]: features
```

```
Out[39]: ['GRE Score',
        'TOEFL Score',
        'University Rating',
        'SOP',
        'LOR ',
        'CGPA',
        'Research']
```

```
In [41]: from sklearn.metrics import accuracy_score,classification_report
```

```
In [42]: accuracy_score(y_test,y_pred)
```

```
Out[42]: 0.85
```

```
In [43]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.25	0.50	0.33	6
1	0.96	0.88	0.92	74
accuracy			0.85	80
macro avg	0.60	0.69	0.62	80
weighted avg	0.90	0.85	0.87	80

Visualizing the decision tree

```
In [46]: import matplotlib.pyplot as plt
plt.figure(figsize=(20,15))
plot_tree(clf,
          feature_names = features,
          max_depth=9,
          class_names=['0','1'],
          filled = True,
          rounded = True,
          fontsize=8)

plt.savefig('decision-tree.png')
```

