Introduction to Data Mining

Assi2

Build CART Decision Tree

```
# Loading neccesary libraries:
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
from sklearn.tree import DecisionTreeClassifier, export_graphviz, plot_tree
from sklearn.model_selection import train_test_split
```

```
import warnings
warnings.filterwarnings('ignore')
from pandas.plotting import parallel_coordinates
import matplotlib.pyplot as plt
plt.style.use('default')
```

```
%matplotlib inline
# without this the plots would be opened  in a new window (not browser)
# with this instruction plots will be included in the notebook
```

```
# Use %config InlineBackend.figure_format = 'retina'
# after %matplotlib inline to render higher resolution images
%config InlineBackend.figure_format = 'retina'
```

```
# If you wish to use Google colab, the following code will allow you to mount your Google Drive. Otherwise, comment on the following lir
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

```
# To print multiple outputs
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'
# Set it to None to display all columns in the dataframe
pd.set_option('display.max_columns', None)
```

```
#Reading the data from google drive
data=pd.read_csv('/content/gdrive/MyDrive/Colab Notebooks/Loans')
#Print that data imported successfully
print("Data imported successfully")
data.head(10)
data.columns
print('\n\n')
data.dtypes
```

```
Data imported successfully
```

	Approval	Debt-to-Income Ratio	FICO Score	Request Amount	Interest
0	F	0.0	397	1000	450.0
1	F	0.0	403	500	225.0
2	F	0.0	408	1000	450.0
3	F	0.0	408	2000	900.0
4	F	0.0	411	5000	2250.0
5	F	0.0	413	5000	2250.0
6	F	0.0	416	6000	2700.0
7	F	0.0	421	2000	900.0
8	F	0.0	422	12000	5400.0
9	F	0.0	432	10000	4500.0

```
Index(['Approval', 'Debt-to-Income Ratio', 'FICO Score', 'Request Amount',
       'Interest'],
      dtype='object')



Approval                 object
Debt-to-Income Ratio    float64
FICO Score                int64
Request Amount            int64
Interest                float64
dtype: object
```

```
# Describing the data
data.describe()
```

	Debt-to-Income Ratio	FICO Score	Request Amount	Interest
count	150302.000000	150302.000000	150302.000000	150302.000000
mean	0.183538	672.023266	13427.080145	6042.186065
std	0.137226	69.129157	9468.345958	4260.755681
min	0.000000	371.000000	500.000000	225.000000
25%	0.090000	647.000000	6000.000000	2700.000000
50%	0.160000	684.000000	11000.000000	4950.000000
75%	0.240000	714.000000	19000.000000	8550.000000
max	1.030000	869.000000	44000.000000	19800.000000

Step6. Create a CART model using the training data set that predicts *Approval* using *Debt to Income Ratio*, *FICO Score*, and *Request Amount*.

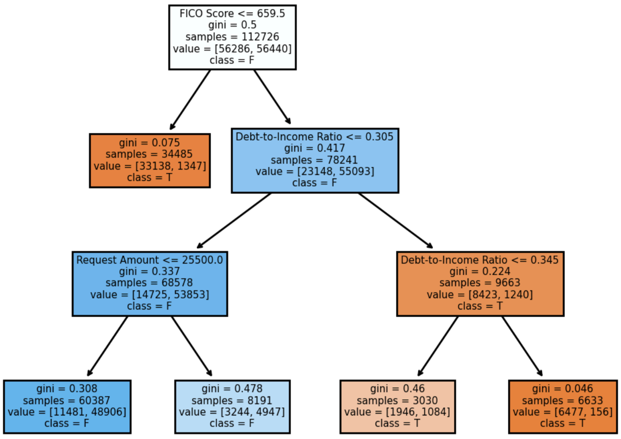
- Set the criterion to **Gini Index** and *max_leaf_nodes* to 5

```
#Write your code here
# lets split the dataset into train and test
#maintain 75%-training set and 25%-testing set
data_train,data_test=train_test_split(data, test_size=0.25,random_state=7)
#print information
print("DataSet size {}".format(data.shape))
print("Train size {}".format(data_train.shape))
print("Test size {}".format(data_test.shape))
#features = Debt to Income Ratio, FICO Score, and Request Amount.
#target=Approval.
features=['Debt-to-Income Ratio','FICO Score','Request Amount']
y=data_train['Approval']
y_name=['T','F']
x=data_train[features]
#create CART Model
cart_01=DecisionTreeClassifier(criterion='gini',max_leaf_nodes=5).fit(x,y)
```

```
DataSet size (150302, 5)
Train size (112726, 5)
Test size (37576, 5)
```

Step7. Visualize the decision tree, then save the resulting graph as "assign1_DT_1.jpg". You need to submit the " assign1_DT_1.jpg" with your code.

```
#Write your code here
_=plot_tree(cart_01,feature_names=features,class_names=y_name,filled=True)
plt.savefig("assign1_DT_1.jpg")
plt.show()
```



Step8. Describe the root split in the assign1_DT_1.jpg decision tree.

>>>>>Answer here<<<<<

The decision starts with the "FICO score" as the root node and algorithm selects the value that separates the data into two subsets.The threshold value is 659.5 which states that if the score is less than or equal to 659.5, the decision is made which is, it belongs to class "T", approval is accepted.Whereas on the other side when FICO score is greater than 659.5, there will be combination of decision (T or F), due to this we will select one more feature and proceed further, based on the gini index.

Step9. Describe the second and third splits in the assign1_DT_1.jpg decision tree.

>>>>>Answer here<<<<<

Among the features selected while FICO Score had low gini index, which made it root node, after the first split on the right side, the algorithm selected the feature "Debt to Income Ratio" the condition was,ratio is less than or equal to 0.305, based on this condition the tree split further, on the left side still the decision was not made and hence the tree split further based on feature "Request Amount" and based on that the tree split left and right on the condition (If the requested amount is less than or equal to 25500), the loan approval will not be accepted. Whereas on the right hand side if ratio is greater it will split further based on condition (ratio less than or equal to 0.345) falls under the class True and splits further hence the decision is made i.e, loan will be approved.

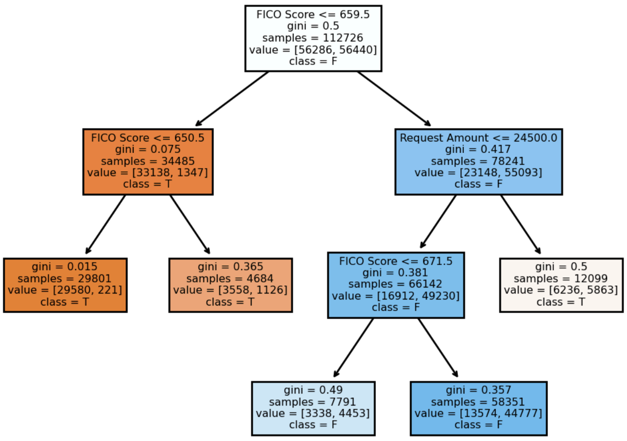
Step10. Create a CART model using the training data set that predicts *Approval* using *FICO Score*, and *Request Amount*.

- Set the criterion to *Gini Index* and *max_leaf_nodes* to 5

```
#Write your code here
#features=FICO Score, and Request Amount.
features=['FICO Score','Request Amount']
#target=Approval
y=data_train['Approval']
y_name=['T','F']
x=data_train[features]
#create cart model
cart_02=DecisionTreeClassifier(criterion='gini',max_leaf_nodes=5).fit(x,y)
```

Step11. Visualize the decision tree, then save the resulting graph as "assign1_DT_2.jpg". You need to submit the "assign1_DT_2.jpg" with your code.

```
#Write your code here
_=plot_tree(cart_02,feature_names=features,class_names=y_name,filled=True)
plt.savefig("assign1_DT_2.jpg")
plt.show()
```



Step12. Describe the root split in the assign1_DT_2.jpg decision tree.

>>>>>Answer here<<<<<

In this step, features are reduced i.e, FICO Score and Request Amount are the selected features, among the following features FICO score was selected had the best threshold value to create branch and maximize the separation and making it root node and based on this condition (FICO score is less than or equal to 659.5), if FICO score is less than the mentioned conditioned then the algorithm will move to left branch and if greater than mentioned condition move towards right branch.

Step13. Describe the second and third splits in the assign1_DT_2.jpg decision tree.

>>>>>Answer here<<<<<

When root node was splitted based on FICO Score still the decision was not made based whether to approve loan or not, it splitted further based on condition if FICO Score less than or equal to 650.5 it falls under the class "T" and hence decision is made to approve loan. Whereas, on the right side the split is made on the condition (if Request Amount is less than or equal to 24500) if yes than, split it into further based on the condition (If FICO Score is less than or equal to 671.5) then the decision is made "F" not to approve loan. And if score is greater than 671.5 then approve loan (falls under the class "T").

Step14. How does your "assign1_DT_1.jpg" decision tree compare to your " assign1_DT_2.jpg" decision tree for the given dataset? Describe the similarities and differences.

>>>>>Answer here<<<<<

Similarities:-

- FICO Score had low gini index, so it was considered as Root Node in both the decision trees.
- The Ratio of True Leaf nodes to False leaf nodes remains same.

Differences:-

- Due to the removal of "Debt to Income Ratio" attribute, after splitting of the root node, the 'FICO Score' was checked again for different threshold.
- Unnecessary Pruning.
- The Feature Importance of 'Request Amount' rose, as a loan can now be approved or declined on the basis of the Request amount, even if FICO Score is less.

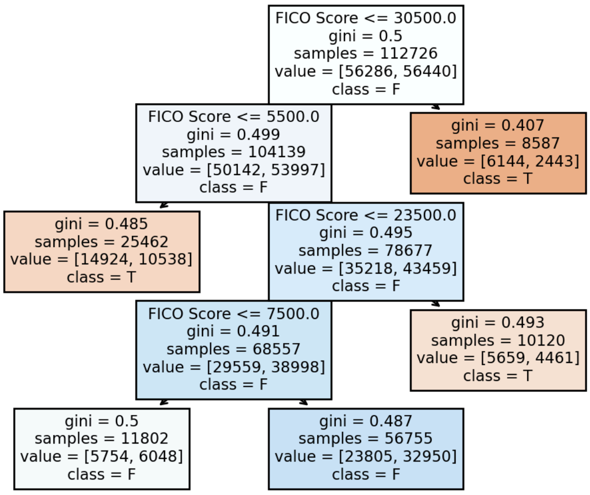
Step15. Create a CART model using the training data set that predicts *Approval* using the *Request Amount*.

- Set the criterion to *Gini Index* and *max_leaf_nodes* to 5

```
#Write your code here
#feature=Request Amount
#target=Approval
features=['Request Amount']
y=data_train['Approval']
y_name=['T','F']
x=data_train[features]
#create cart model
cart_03=DecisionTreeClassifier(criterion='gini',max_leaf_nodes=5).fit(x,y)
```

Step16. Visualize the decision tree, then save the resulting graph as "assign1_DT_3.jpg". You need to submit the "assign1_DT_3.jpg" with your code.

```
#Write your code here
_=plot_tree(cart_03,feature_names=features,class_names=y_name,filled=True)
plt.savefig("assign1_DT_3.jpg")
plt.show()
```



Step17. Describe the root split in the "assign1_DT_3.jpg" decision tree.

>>>>>Answer here<<<<<

In this step, we consider only a single feature to build our decision tree which is 'Request Amount', and thus it is selected as the Root Node, On further analysis of the dataset the loan amount was approved almost for every request with 'Request Amount' value less than 30500, which indicates the 8587 samples(or requests) that may or may not depend on other feature. The Left side of the decision tree further splits based on the request amount but with a different threshold.

Step18. Describe the second and third splits in the "assign1_DT_3.jpg" decision tree.

>>>>>Answer here<<<<<

The second and the third split indicates the range of request amount which will determine if the request will be approved or rejected. As it can be seen all the values which are less than 5500 will have high chances of approval, and those which do not will checked for further, leading to the third split, which checks for different threshold, but still it is difficult to determine as the information is too less to predict the approval or rejection based on 'Request Amount'. Hence the Accuracy will be affected.Overall, the Decision tree is made to satisfy the number of samples and thus creates different thresholds for the samples with nearly same request amount but with different result for 'Approval'.

Step19. How does your "assign1_DT_1.jpg" decision tree compare to your "assign1_DT_3.jpg" decision tree for the given dataset? Describe the similarities and differences.

>>>>>Answer here<<<<<

The "assign1_DT_1.jpg" decision tree has nearly all the features of the dataset making it more concise and interpretable, on the other hand the "assign1_DT_3.jpg" decision tree has only one feature thus to determine the outcome based on only one feature is difficult and way harder to interpret. It might oversimplify the problem, leading to poor generalization to new, unseen data.

Similarities:

- The Ratio of True Leaf nodes to False leaf nodes remains same.

Differences:

- The Root Node of 'assign1_DT_1.jpg' is 'FICO Score', whereas the root node of 'assign1_DT_3.jpg' is 'Request Amount'.
- Feature Importance of 'Request Amount' has significantly risen as it is the only feature.