Ricardo Diaz
Intro to Data Mining – DATS 6103
12/6/2021

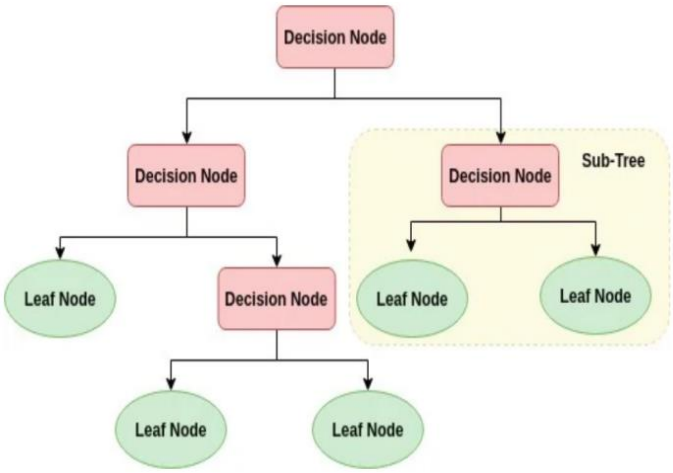
INDIVIDUAL FINAL REPORT – GROUP 4

DESCRIPTION

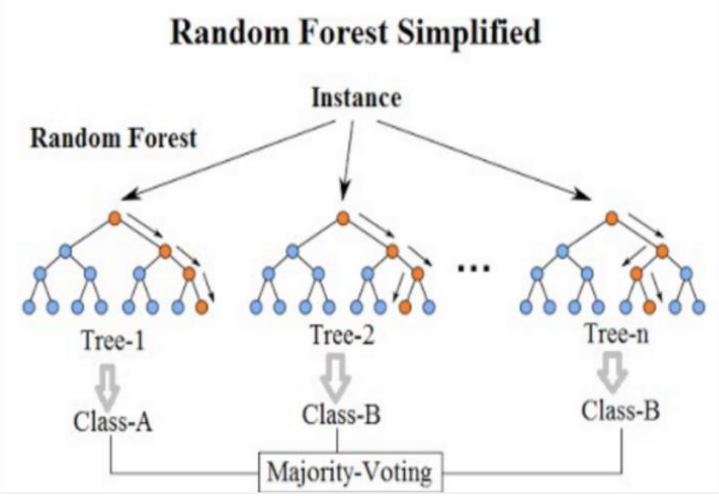
Our project examines factors that affect admission to the Masters programs in the US to see how well they predict the admission, using a dataset by UCLA graduates from Kaggle.

The workload for this project was broken into four:

- Data preprocessing
- Data visualization
- Machine learning modeling
- GUI (pyqt5)



A decision tree is a flowchart like a tree structure where a branch represents a decision rule, and the leaf node represents the outcome.



Random Forest is a ensemble of decision tree where every new data input enter each decision tree in the forest and is evaluated and classified. The most appeared classification is the result of the random forest

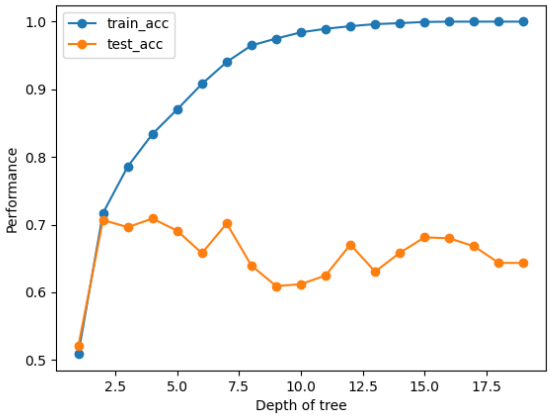
My work load for this project was broken into two:

- **Machine learning modeling**
- **GUI**

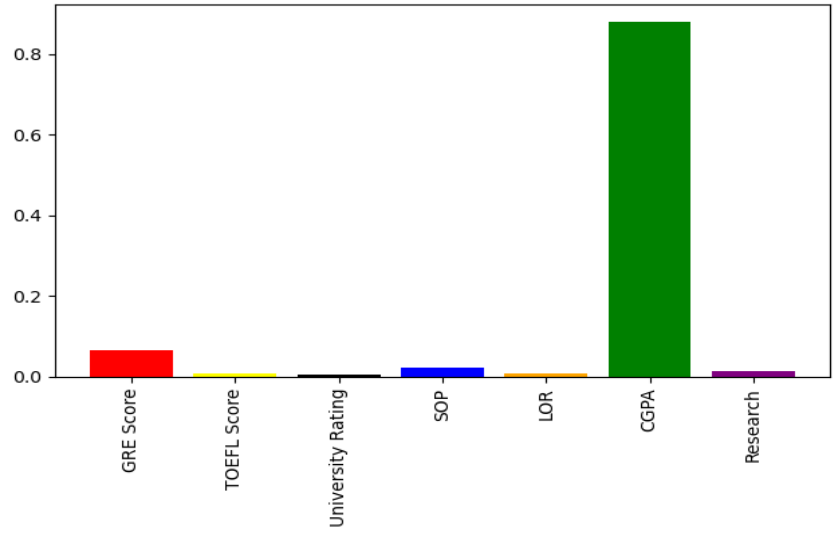
I focus on building and improving three models. Decision tree, Random Forest and Linear Regression (with the data scaled), build the different performance scores and build the feature importance's. After this I put it in PYQT5 so the user can play with the different scores and train/test size.

RESULTS:

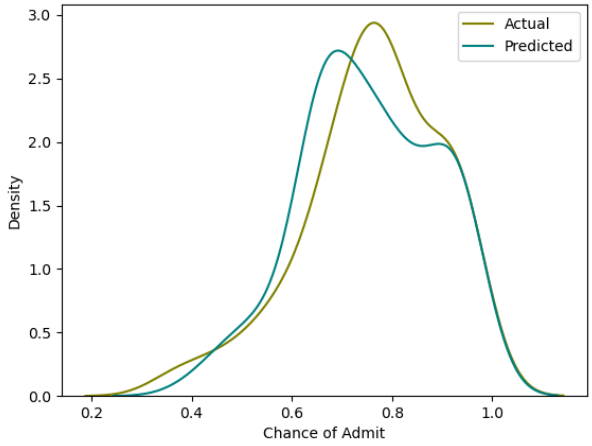
Decision Tree:
In the case of decision tree, even though our cross validation showed this was of our worst models it was the first model we learned in class and the first I built. I used this model before using scaling the data. To find the max depth in our decision tree I graph the score using different max depth.



According to this graph max depth 5 was the best for our decision. This manual method wasn't efficient to find the best paraments on our model and at this point I wasn't aware of hyperparameter.



The decision tree feature importance is not good. It only gave important to CGPA and GRE score conflicting with our EDA and other models feature importances.



The distribution graph of the actual y and decision tree predicted y variable shows is not similar. Our Decision Tree Regressor Score: 69.0% After this I decide to officially discard this model.

Random Forest:

In the case of the random forest, since the cross validation show this was one of best models. I wanted to find the appropriate parameters to have the best model possible. There were two options GridSearchCV and RandomizedSearchCv to find the best parameters. I decided to do RandomizedSearchCv for running purposes. I really took in consideration the running time in the selection of the searchcv and the different parameters I wanted to be tried.

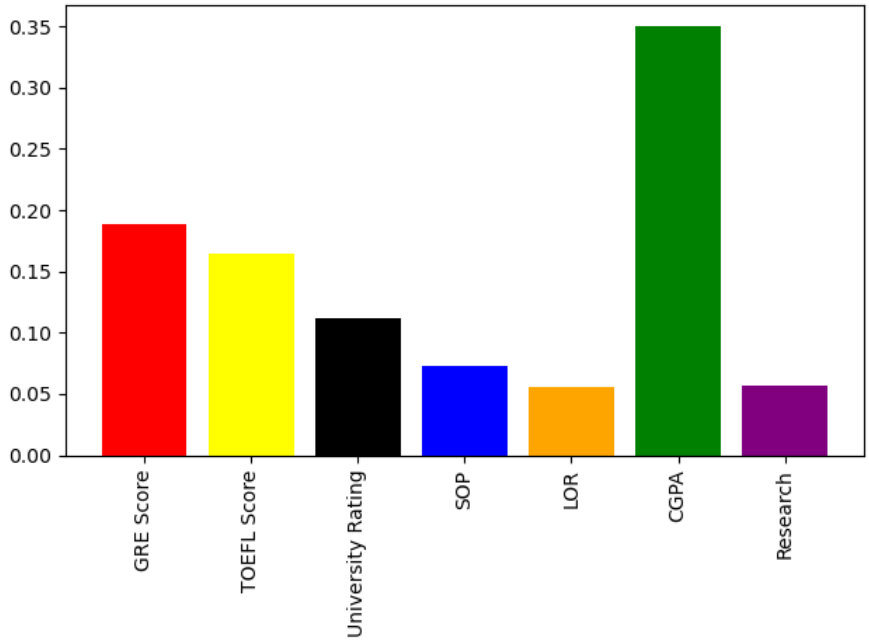
```
# Fitting Random Forest
rf = RandomForestRegressor()
# Number of trees in random forest
n_estimators = list(np.arange(10, 100, 10))
# Criteriion for the random forest
criterion = ['mse', 'mae']  ## if you put this as a parameters , takes a long
time to run and performance goes down
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = list(np.arange(4, 12))
# Minimum number of samples required to split a node
min_samples_split = [2, 5]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2]
# Method of selecting samples for training each tree
bootstrap = [True, False]
rf_param = {'n_estimators': n_estimators,
            'max_features': max_features,
            'max_depth': max_depth,
            'min_samples_split': min_samples_split,
            'min_samples_leaf': min_samples_leaf,
            'bootstrap': bootstrap}
```

This were my different parameters I wanted to tried on my random forest. I decided to opt out on the different criterion since the running time was model and my computer couldn't handle it.

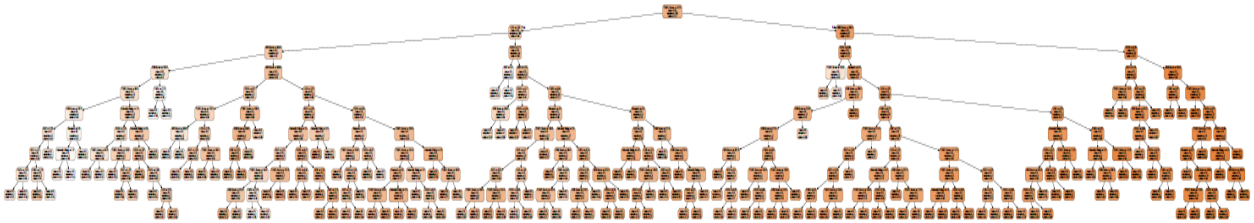
After the randomized searchcv which I iterate 100 I found the best parameters were.

```
rf = RandomForestRegressor(n_estimators=80, min_samples_split=2,
min_samples_leaf=1, max_features='sqrt', max_depth=10,
                           bootstrap=True)
```

This Random Forest gave me a score of minimum 76% and maximum 79%. This was a good and decided to good ahead with the parameters and the model and include it in my PYQT5.



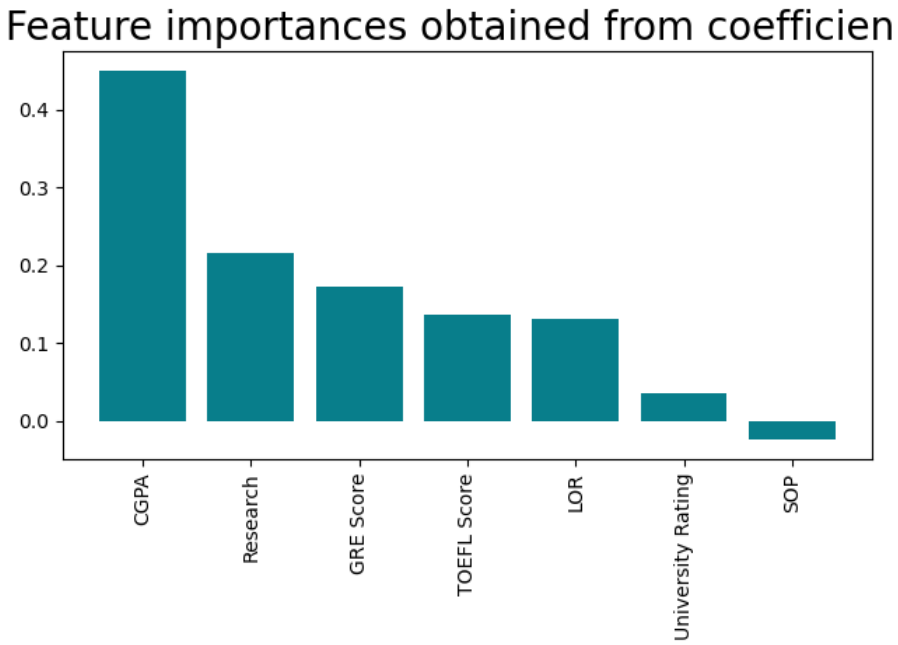
An insight I found interesting was the feature importance of the random forest. At first our EDA was confusing regarding the Research feature, in the heatmap it showed no correlation but the scatterplot it shows a very slight relationship with chance of admission. In our random forest model, it showed it was important in our prediction.



This a picture of one of our random forests, even though its large it showed no overfitting basing on the depth of tree and the scores it gave for test and train.

Linear Regression:

In the case of linear regression, I had to scale our data to do the linear regression. I tried MinMaxScaler and Standard Scaler, both gave me similar result so I decided to do Standard Scaler since our data distribution is normal.



This feature importance based on the coefficients showed research have a correlation to predicting the chance of admission which is similar to our Random Forest feature importance , also this feature importance gave very low correlation to SOP which is contrary to our EDA correlation heatmap.

The Linear Regression Scaling Regression Score was 78% which is the same score with the linear regression without scaling.

CONCLUSION:

I can conclude the best results are Random Forest and Linear Regression with scaling. This shows our cross-validation graph was correct. I selected linear regression with scaling to be the best model since the accuracy is 78% in comparison to the random forest accuracy which variates from 76-79%.

One improvement I could do is try different ensemble techniques to find a best model. Another improvement can be to find different admission graduate dataset in USA, put it together in one big dataset and then all the preprocessing and models. I suspect there is a way to use feature creation in this dataset but all the different trials we did failed, but I still have a feeling there is feature hiding somewhere.