Data Science Engineering Methods and Tools

Graduate Admission Predictor for Student and University

Akshara Singh lnu.akshara@husky.neu.edu
Anurag Dhar dhar.a@husky.neu.edu
Naveen Jami jami.n@husky.neu.edu
Satwik Kashyap kashyap.sa@husky.neu.edu

Introduction

A global and cultural awareness and respect is today considered a critical component of an engineering degree. The United Sates is the home to the world's most prestigious university with each year, thousands of applications for admission from student with diverse, unique backgrounds.

Even the most qualified and confident applicants worry about getting into grad school in their dream universities. There are several online portals that provide admission related statistics, but do not provide information regarding individual profile data, thus, leaving the applicant with the only option of guessing and hoping for the admit or reject.

Universities on the other hand, also receive thousand of application each year. Handling this admission process manually could be tedious and could also lead to errors. Since the number of applicants have significantly increased in the previous years, there is a need to deploy a more effective and efficient automated method to handle the admission process.

In this project, we are focused on modelling graduate admission to Computer Science in United states. The project provides answers to some of the questions for students and University:

- To which university should I go to for computer science in USA?
- What are my Top 5 colleges for computer science to apply to get an admit in USA?
- Which candidate should be selected from vast range of students?

Our goal for the models is to solve the issue faced by students and Universities by developing machine learning models using the previous year Admits/Rejects of the students.

- ***Assisting Student Model:*** The model addressed the problems faced by students by developing a machine learning approach which make an applicant make informed decision by evaluation their chances of admission using a probability-based method across 29 Universities.
- ***Assisting University Model:*** The model is built on Northestern university data set to help make better choices of applicants from diverse group of student's profiles.

Data Set

The graduate admissions application contains student's private information and so it is kept confidential by the respective Universities. For the project, we have scraped data from various online portals like Yocket, Edulix, MSinUS where students self-report their profile information. The dataset was based on the assumption that the profile information which were reported on the portals were correct.

The dataset prepared with taking into consideration **29 universities** from Rank 1 – Rank 130 for **Computer Science** Department. The dataset consists of profiles for Indian students. For the ranking of university, we went through http://csrankings.org for reference.

The dataset for divided into 2 parts:

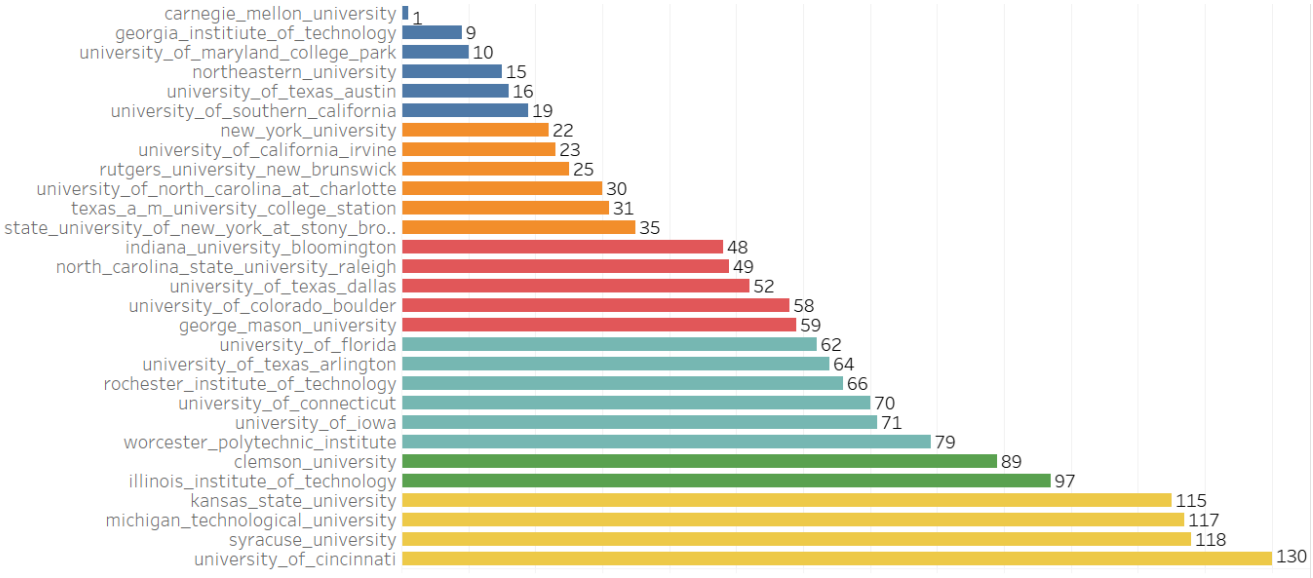
Student Assisting Model:

The dataset has a total of **9350 rows and 9 columns** across 29 Universities. Each row in the dataset represents a student profile with admits and rejects. In the dataset we had **53.5% rejects** profile and **46.5% admit** profile.

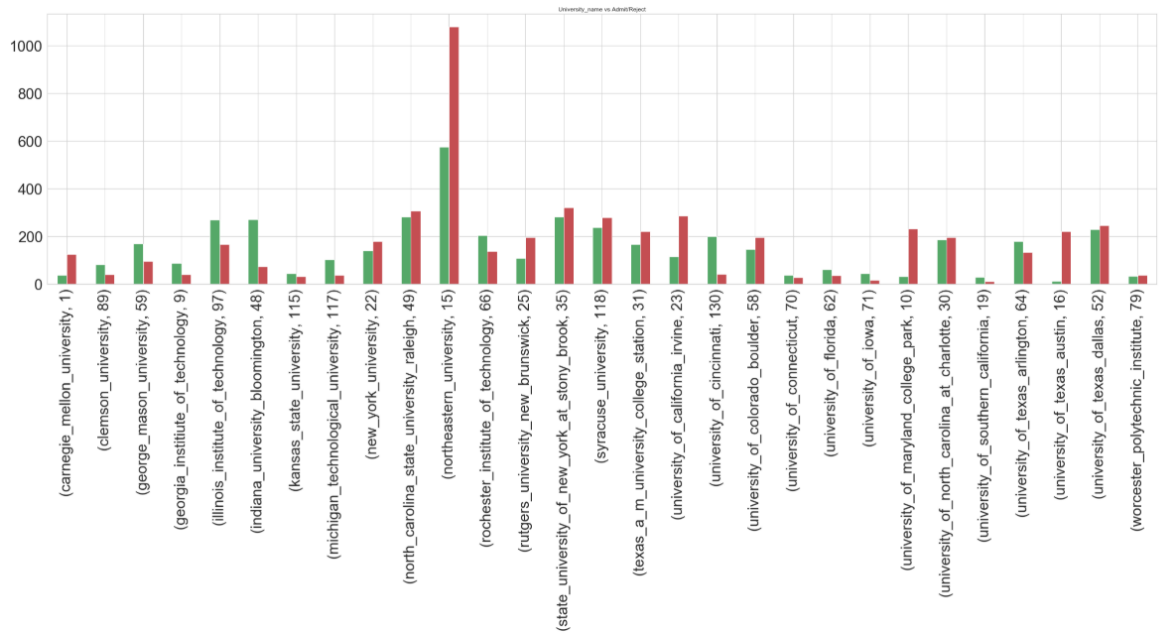
University Assisting Model:

For the data set we have only taken into consideration only Northeastern University data to build University model. The dataset has a total of **1654 rows and 9 columns.** In the dataset we had **65.2% rejects** profile and **35.8 % admit** profile.

Below charts show the colleges and the distribution of Admits and Rejects across 29 colleges.



Fg.1 Universities with their corresponding ranking for Computer Science Program



Fg.2 Each University count for admits/rejects

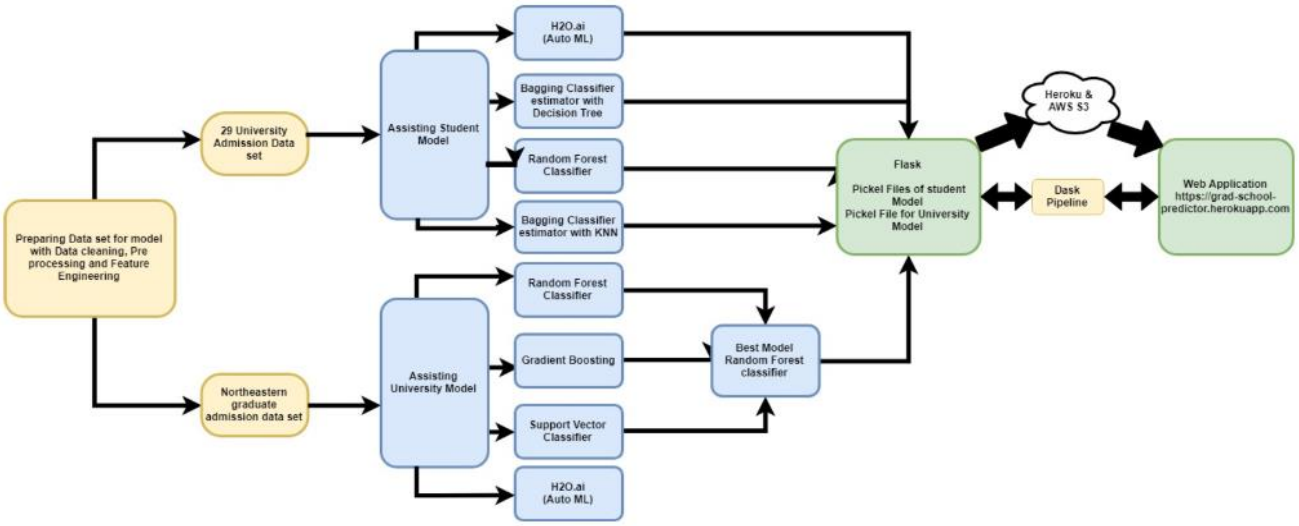
We considered the below features for the dataset

Features								
English test score	Gre Score	Gre Score Quant	Gre Score Verbal	Paper Published	Ranking	Undergraduation Score	University Name	Work experience

Table 1 Features in the dataset

Implementation

To understand the impact of prediction System and to use the power machine learning, we have designed a complete application that will help an applicant in predicting whether he/she will get an admit from university of his/her choice. To build a machine learning model that will predict the chance of getting admission, we had to train the model with different scenarios and with various historical data.



Fg.3 End to End Implementation Workflow

Data Cleaning, Pre-processing and Feature Engineering of Data

Data that we have scraped from Yocket and MSinUS, had the following issues that we addressed.

- Noisy Data
- Unformatted Text
- Inconsistent Data

In order to address the data issue, we followed some of these steps:

- Dropping null values if it cannot be replaced statically
- Filling the numerical values with median values and the categorical values using mode. For example, Gre verbal, Gre Quant score
- Noisy and unstructured data was removed using regex pattern to get desired text format
- Unformatted text with incorrect datatype were changed to the required datatype

We scaled the data set to have the features in same scale. For e.g. TOEFL is graded out of 120 and IELTS out of 9, so we have to scale them into equal scale. For reference for scaling of IELTS and TOEFL score as a single entity we referred TOEFL, IELTS Score Comparison Tool on ETS website.
Likewise, for undergraduate score reported were in scale of 10(percentile) and 100(percentage). We performed the calculation and aligned the score in the scale of 4.
For the papers published field we assigned corresponding numeric values (1/2/3) to Local/National/International papers.
The ranking of the universities has been included in the dataset as it plays an important role in selecting candidate's profile.

Automated Feature Engineering

We also performed automated feature engineering using Feature Tools to get the prediction accuracy for admit or reject according to a student's profile. Feature tools is an open source library for performing automated feature engineering. Three major components of the package are Entities,

Deep Feature Synthesis (DFS), Feature primitives. The entity set was divided into University data and Admission data.

```
Entityset: admissions_data
  Entities:
    admissions_data [Rows: 9350, Columns: 9]
    university_class [Rows: 29, Columns: 2]
  Relationships:
    admissions_data.university_name -> university_class.university_name
```

Fg.4 Entity set using Feature tools

We trained and tested 2 models and the summarization of the performance with error metrices calculation as below:

Model	Train Accuracy	Test Accuracy	F1 Score - Train	F1 Score - Test	AUC-ROC Curve - Test	AUC-ROC Curve - Train
Logistic Regression	0.63	0.61	0.56	0.53	0.67	0.69
Random Forest CLassifier	0.98	0.71	0.98	0.70	0.78	1.0

Table 2 Summarization of result for automated feature engineering

Models

Machine learning classification technique, which a supervised learning methodology which we have used in our implementation. The two models that were a classification problem with a probability-based output:

- Assisting Student Mode
- Assisting University Model: Classification problem

Implementation and details for the model has been summarized in next section.

Automated Machine Learning

For an Automated Machine Learning, we used an open source software H2o.ai. Using H20.ai we were able to get the prediction for the classification problem for making

Best 4 Model have been summarized below:

- StackedEnsemble_AllModels_AutoML_20190421_184847
- StackedEnsemble_BestOfFamily_AutoML_20190421_184847
- GBM_2_AutoML_20190421_184847
- GBM_4_AutoML_20190421_184847

Best Model Summary : StackedEnsemble_AllModels_AutoML_20190421_184847

Model	Train Accuracy	Test Accuracy	F1 Score - Train	F1 Score - Test	AUC-ROC Curve - Test	AUC-ROC Curve - Train
StackedEnsemble_AllModels_AutoML_20190421_184847	0.90	0.72	0.88	0.64	0.83	0.99

Table 3 Auto ML best model summary – Assisting Student Model

Best 4 Model have been summarized below:

- StackedEnsemble_AllModels_AutoML_20190425_050507
- StackedEnsemble_BestOfFamily_AutoML_20190425_050507
- GBM_2_AutoML_20190425_050507
- GBM_5_AutoML_20190425_050507

Best Model Summary : StackedEnsemble_AllModels_AutoML_20190425_050507

Model	Train Accuracy	Test Accuracy	F1 Score - Train	F1 Score - Test	AUC-ROC Curve - Test	AUC-ROC Curve - Train
StackedEnsemble_AllModels_AutoML_20190421_184847	0.90	0.68	0.89	0.41	0.77	0.99

Table 4 Auto ML best model summary – Assisting University Model

Pickling

Once the model is built, it's time to put the model into use. So, we stored our model in pickle format, so that it could easily be used from the application.

Dockerized and Deploy

Docker Hub was used to build a docker image of our application. The docker image was deployed on an on-cloud platform Heroku. The docker image is also build to run the web application on local host.

WebApp

The application is built using flask, that uses Jinja template to render HTML.

Pipeline Design

Data Science pipelines are sequences of processing and analysis steps applied to data to save on design time and coding.

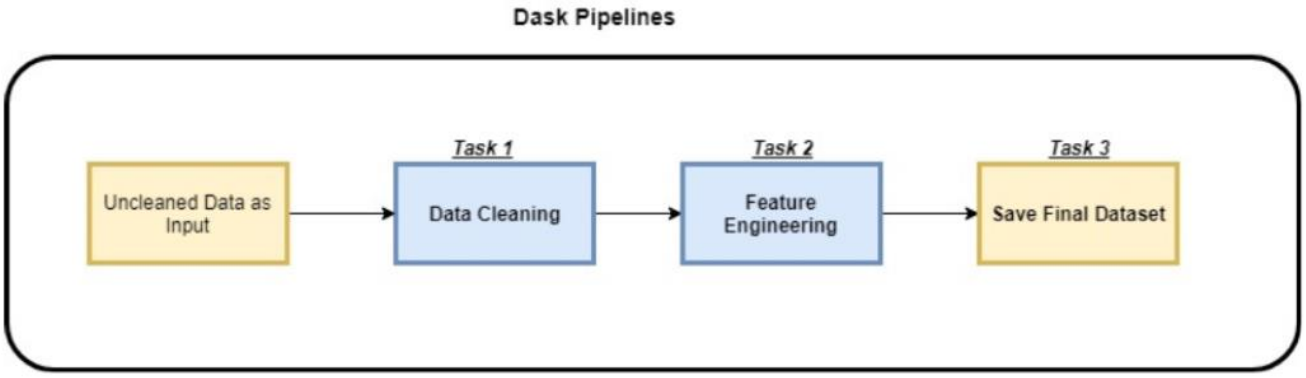
In this project we have incorporated pipeline to prepare cleaned data set from the scraped uncleaned data set, which would be used as an input to different models. Our pipelining methodology is implemented in two places:

Pipelining data for the model input:

Input to Task 1: Dataset prepared from scraping

Output after Task 2: Cleaned Dataset which could be used for modelling

- Task 1: Data cleaning with removing Inconsistent Data, Noisy Data and Unformatted Text
- Task 2: Data Pre-processing and Feature engineering with one hot encoding on multiple columns including English Score, University ranking, undergraduate score, paper published



Fg.5 Dask Pipelining

Pipelining the inputs taken from the user in WebApp

Input to Task 1: Student profile information submitted on the Web application form

Output after Task 2: Formatted and scaled data to be used for prediction

- Task 1: Scaling TOEFL and IELTS English score
- Task 2: Scaling under graduation score on a scale of 4
- Task 3: Mapping of University Name to corresponding ranking
- Task 4: Assigning corresponding values to Paper published fields

Modelling

Model Implementation for Assisting University:

The model deals with creation of classification model which could be used by Universities for selecting suitable applicants for their program. This is designed by establishing predefined requirement criteria. This model employs the Random Forest, SVM-Linear and Gradient Boosting Classifier.

The dataset considered for the model implementation has :

University – Northeastern University

Department – Computer Science

Challenge: On analysing data, we observed the dataset was not balanced, hence we resampled dataset to make it balanced.

Test train split: 20:80

Earlier: reject 1079 accept 574 Name: status	After: reject 1079 accept 1000 Name: status

Table 5 Balancing dataset summary

Labels: gre_score_quant, gre_score_verbal, test_score_toefl, undergraduation_score, work_ex, papers_published

Target: status

We have also performed cross validation and hyper tuning using GridSearch CV and summarized the results for multiple models

Random Forest Classifier: Random forests form a family of methods that consist in building an ensemble (or forest) of decision trees grown from a randomized variant of the tree induction algorithm. The forest it builds, is an ensemble of Decision Trees, most of the time trained with the "bagging" method.

Model	Test Accuracy	Train Accuracy	Grid Search - Test Accuracy	Grid Search - Test Accuracy	Test F1 Score	Train F1 Score	Best Parameter	Interpretability	Reproducability
RandomForestClassifier	0.83	0.98	0.84	0.99	0.84	0.99	{'bootstrap': True, 'criterion': 'gini', 'max_depth': 20, 'min_samples_leaf': 1, 'n_estimators': 30}	Yes	No

Table 6 Summarizing Random forest classifier – Assisting University Model

Support Vector Machine: A Support Vector Machine (SVM) is a discriminative classifier formally defined by a separating hyperplane. In other words, given labelled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two-dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

Model	Test Accuracy	Train Accuracy	Grid Search - Test Accuracy	Grid Search - Test Accuracy	Test F1 Score	Train F1 Score	Best Parameter	Interpretability	Reproducability
Model-SupportVectorMachine	0.6658	0.6844	Grid Search Test - 0.6686	Grid Search Train - 0.6944	0.66	0.65	{'C': 1, 'class_weight': 'balanced', 'degree': 3}	Yes	Non-reproducable

Table 7 Summarizing Support Vector classifier – Assisting University Model

Gradient Boosting Classifier: Gradient Boosting trains many models in a gradual, additive and sequential manner. gradient boosting performs the same by using gradients in the loss function (*y=ax+b+e , e needs a special mention as it is the error term)*. The loss function is a measure indicating how good are model's coefficients are at fitting the underlying data.

Model	Test Accuracy	Train Accuracy	Grid Search - Test Accuracy	Grid Search - Test Accuracy	Test F1 Score	Train F1 Score	Best Parameter	Interpretability	Reproducability
Gradient Boosting Classifier	0.79	0.80	0.84	0.99	0.81	0.82	{'n_estimators': 80}	Yes	No

Table 8 Summarizing Gradient Boosting – Assisting University Model

As it is evident from above table Random forest performance is better than other two models, Support vector machine and gradient boosting.

Graduate Admission model for Assisting Student:

There are 2 functionality that we covered to prepare the model:

- Students is getting an admit/reject in the selected university
- Student is shown with the list of 5 top ranked Computer science university

We have a labelled dataset (accept & reject) of MS in Computer Science admission profiles scraped from Yocket of 29 Universities in US.

Challenge: On analysing data, we observed the dataset was not balanced, hence we resampled dataset to make it balanced.

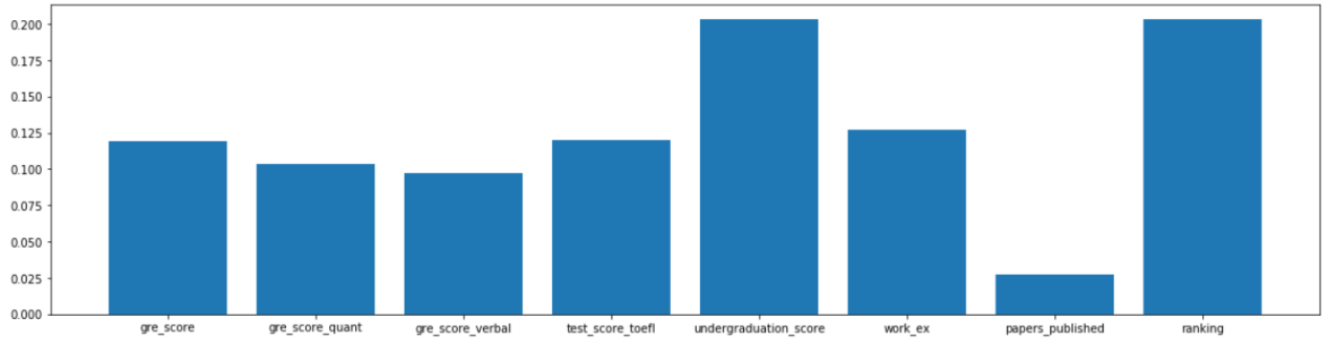
Test train split: 25:75

```
dataset.status.value_counts()

reject    5055
accept    4789
Name: status, dtype: int64
```

Table 9 Admit and Reject count for dataset

Below was the feature importance for the dataset:



Fg.6 Feature Importance

We tried Support Vector Machines, Decision Tree Classifier, Random Forest Classifier, Bagging Model with Decision Tree classifier and kNN, MLP Classifier, GNB Classifier on our data . We have also performed cross validation and hyper tuning using GridSearch CV and summarized the results for multiple models. The result highlighted that ensemble techniques like Random Forest and Bagging models are giving best results giving up to 80% test accuracy and which we pickled and deployed in our website.

Model	Test Accuracy	Train Accuracy	Grid Search - Test Accuracy	Grid Search - Train Accuracy	Test F1 Score	Train F1 Score	Test AUC ROC Curve	Train AUC ROC Curve
Bagging - Decision Tree	0.79	0.99	0.80	0.98	0.78	0.99	0.89	0.99
Random Forest Classifer	0.77	0.98	0.78	0.97	0.77	0.97	0.87	1.0
Bagging - kNN(k=3(Best))	0.72	0.87	0.74	0.94	0.73	0.94	0.82	0.99

Table 9 Summarizing Best 3 Models – Assisting Student Model

How to run the model?

The model could be accessed in two ways:

- Docker Image
- On cloud Platform: Heroku

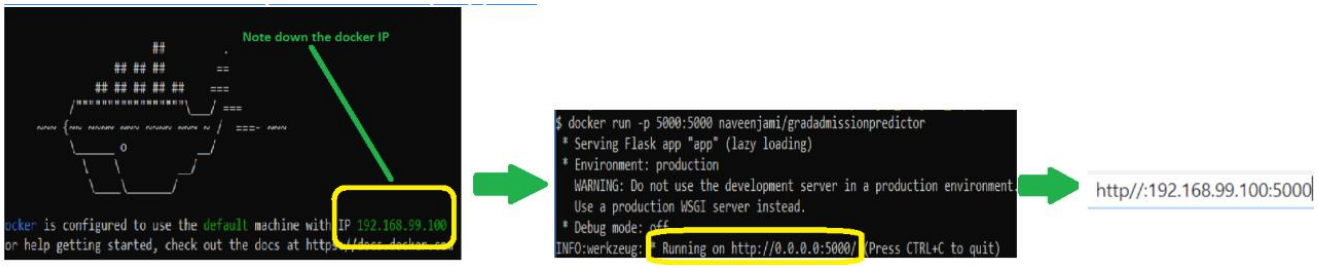
Docker Image

To use the docker image, follow the below instructions:

- Extract the docker IP
- Run the below command on docker and the output would be a URL:

  docker run -p <laptop port>:5000 naveenjami/gradadmissionpredictor

- Run on browser:
  http://dockerIP:<laptopport>

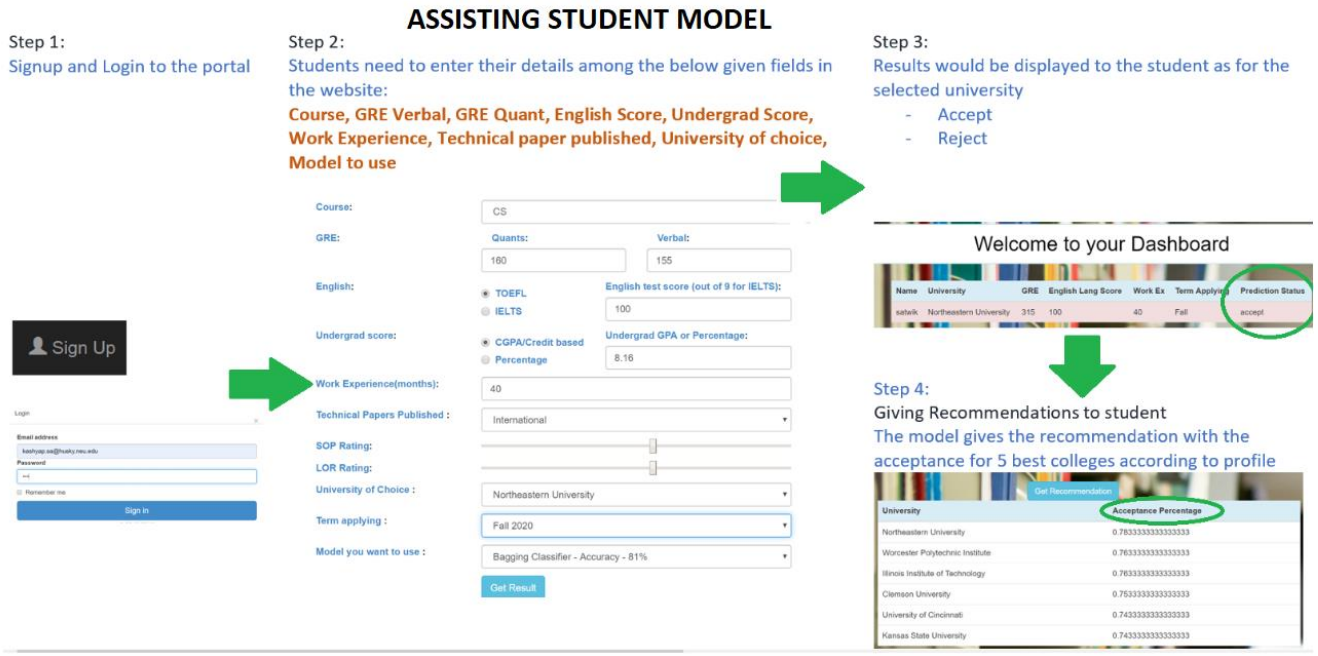


Fg.6 Docker Image to launch application

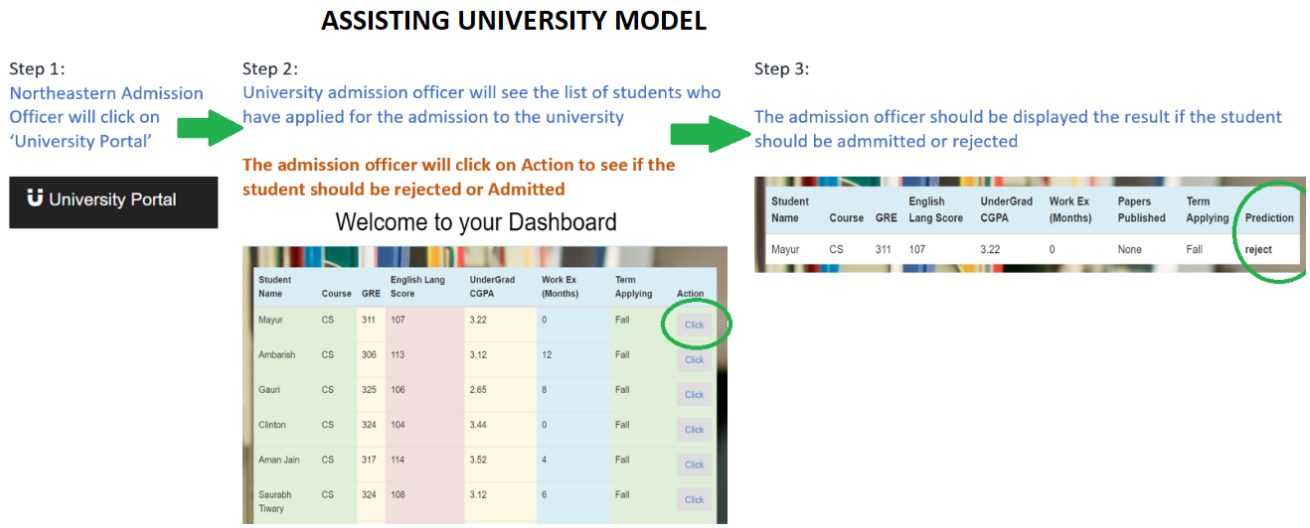
On Cloud Platform: Heroku

The model has also been pipelined, dockerized and deployed on Heroku. To launch the web application, launch the URL: https://grad-school-predictor.herokuapp.com/

Below screenshots covers the detailed steps to use the 'Student Assisting Model' and 'University Assisting Model'



Fg.7 Student Assisting Model - steps to reproduce



Fg.8 University assisting Model - steps to reproduce

Conclusion

This project has been a great learning experience covering end to end data science workflow covering aspects from data scraping to model deployment.

We can conclude that for the Assisting Student Model, our conclusion could be divided into 2 parts:

- For the chances of admit/reject of the student in a selected university we found the best model as Bagging classifier with decision tree
- For the recommendation system to provide the top 5 university according to the student profile, we found the best model as Bagging classifier with decision tree with the prediction accuracy of 80% on test and F1 score of 0.78

Similarly, for the Assisting University Model, the Random Forest Classifier was the best model with an accuracy of 84% on test and F1 score of 0.84

Future Work

The Model could be used by students and University for Graduate Admission decision process. The model was performing with good Accuracy on the available data set and features. The performance of the model could be further improved if we have additional and more diverse data including such as Letter of Recommendation rating, statement of purpose rating, count of Local/National/International Paper Published etc. We could also enhance the model with student data from multiple geographic location and keeping in mind that the reported data is authentic.

References

University Rankings in Computer Science - http://csrankings.org

Toefl, IELTS Score Comparison Tool - https://www.ets.org/toefl/institutions/scores/compare/

Web Scraping - https://yocket.in, https://www.edulix.com/

Wikipedia.com