Web-App for Prediction and Analysis of Graduate Admissions

Design and Analysis of Algorithms Minor Project

Ankit Grover

CCE , Manipal University Jaipur

ankit.189303158@muj.manipal.edu

I. INTRODUCTION

The following project was made to help students evaluate their chance of admission into top universities Masters admissions based on their academic profile.

II. EASE OF USE

A. *Open Source*

The following project including the project data has been open sourced at [1] with pre-trained models, and further setup instructions.

III. OBJECTIVE

Despite many universities not requiring standardized test scores as a part of their evaluation process [2] of a given candidate, many universities still require such scores. Universities often do not provide cut-offs for standardized scores and GPA as well however these are still heavily considered during evaluation. By providing an analysis and model to predict based on given data which follows the same patterns as the UCLA Graduate Admissions dataset [3] , users can use this as estimate to improve , as well shortlist universities realistically based on their profile.

IV. DATASET

The dataset has been chosen from Kaggle [5].The dataset contains several parameters which are considered important during the application for Masters Programs. These include TOFEL, GRE scores, GPA, University Rating, Statement of Purpose and Recommendation Letter strength, The dataset contains several parameters which are considered important during the application for Masters Programs. Research Experience.

- GRE Scores, TOEFL Scores : Higher is considered better
- Research Experience: Essential part of deciding admittance.
- University Ratings: Higher is better .
- GPA: Higher GPA highlights academic excellence and resilience.
- Statement of Purpose and Letter of Recommendation Strength: Highlights research ability and candidate research goals and vision.
- Research Experience: Previous research experience is often complimentary.

Parameters	**Range**
GRE Scores	0-340
TOEFL Scores	0-120
GPA	1-10(Scale)
University Rating (higher better)	0-5
Statement of Purpose and Letter of Recommendation Strength	0-5
Research Experience	1-Yes 0- No experience

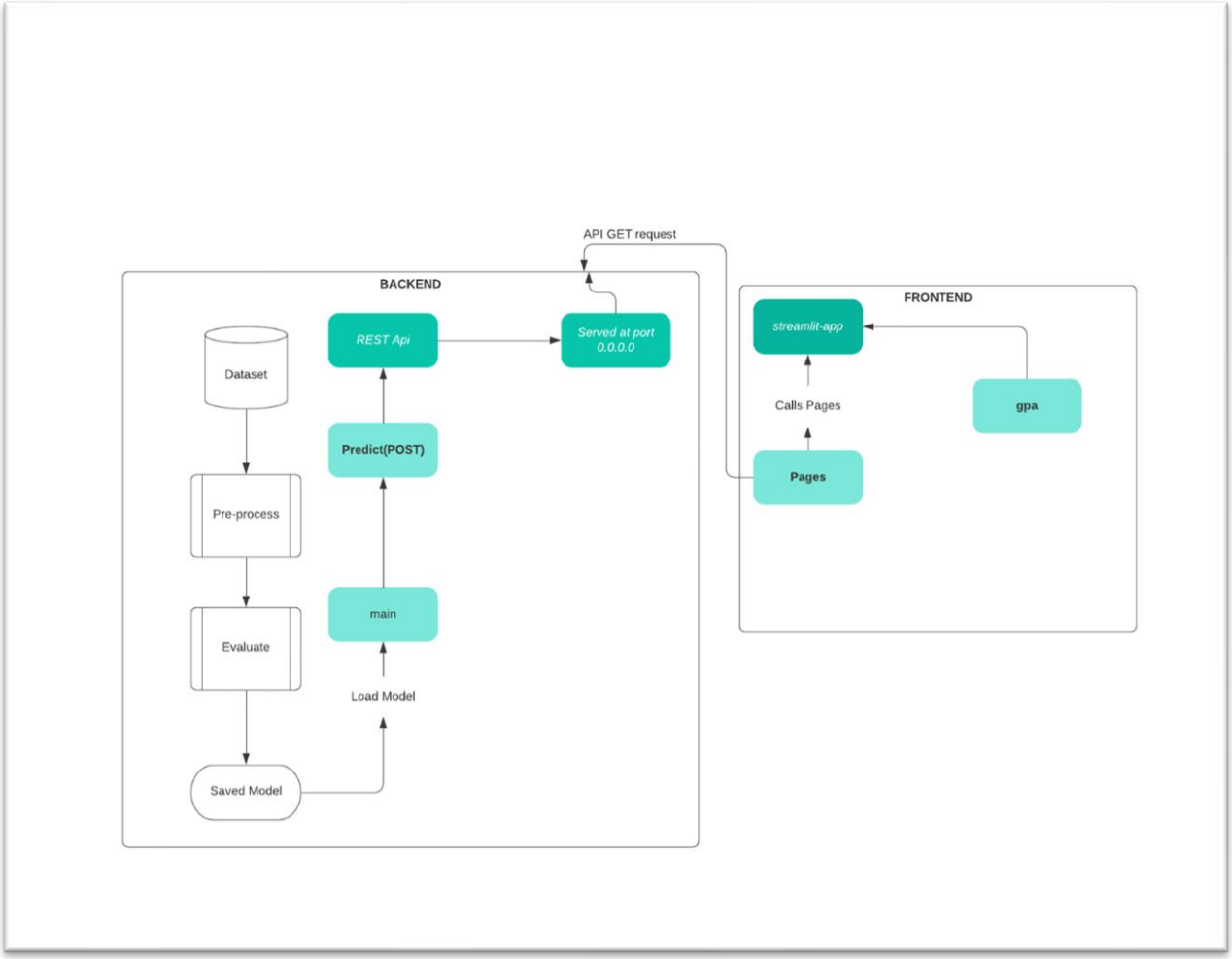
Fig 1: Parameters for evaluation of candidates

[1]https://github.com/Agrover112/Graduate_Admissions_Prediction.

V. APPLICATION ARCHTECTURE

In the following section we explain the architecture of our given web-application which has been created in the following project. Refer APPENDIX for details of the code .

The architecture of the given workflow is show below.



VI. DATA ANALYSIS

The dataset is analysed to uncover varies correlations and statistics involved. The chat below shows how chance of admission clearly increase on the increase of candidates GPA as well higher university rating in Fig 3. We also explore how important prior research experience is during admissions. We see a clearly candidates incorporate research work along with their GPA which increases their chances.55% of the candidates had prior research experience.

We further analyze which features are the most important for increasing the candidates chances. It is clear that despite many universites not putting weightage on standardized scores , they indeed play an important role . Also candidates with higher GPAs also have higher chance due to demonstrated academic ability thought not necessarily research ability of innovative ability[6][7]

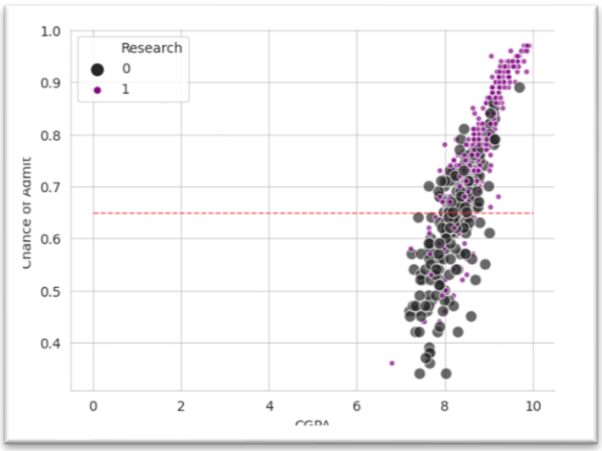


Fig 5: Research work essential for increasing chances.

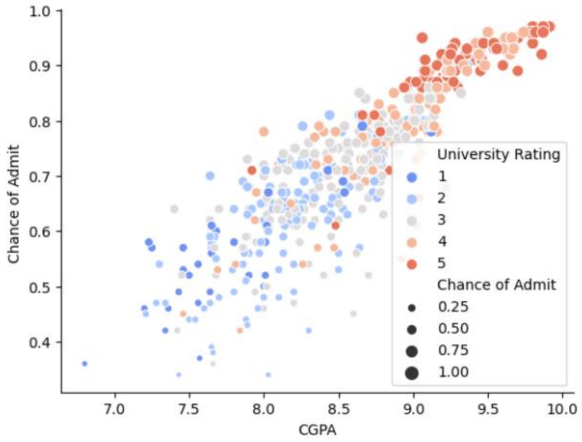


Fig. 3: Chance of Admit vs CGPA chart

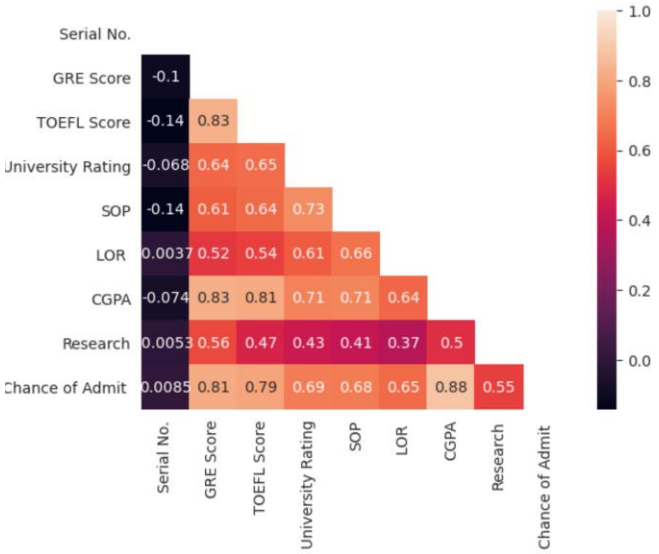


Fig. 4: Correlation between various parameters

VII. FURTHER IMPROVEMENTS

A. *Technical Improvements*

- As evident in the architectural diagram above the need for a Database for storing and loading the given model and data is essential.
- A NoSQL database such as MongoDB would be essential , especially during growth of the data.
- The data can further be encrypted to prevent data piracy and attacks using SHA or other cryptographic techniques.
- More in-depth exploration of the meaning of various parameters as not specified clearly in the dataset.[5]"*Institutions with a rank of 1 have the highest prestige, while those with a rank of 4 have the lowest*."[8]
- Enhancements to code including:
    - Comments, with descriptions
    - More exception handling.
- Speed improvements which can be done by updating the Streamlit version from 69.2.dev→71 or later.

B. *Further Analysis*

Despite the features provided in the paper more work has to be done in creating a better dataset with more features. There might be many other features thought not considered or collected could possibly affect the outcome. These include country of study, number of achievements, Resume score, names of the university of study, company affiliation, work experience.[9]

VIII. CONCLUSION

This further information would highlight more information such as hidden bias[10] or trends based on new features proposed.

References

[1] https://www.bestmastersprograms.org/universities-without-gre-requirements/#:~:text=Schools%20are%20changing%20the%20admissions,their%20preferred%20field%20of%20study.

[2] https://github.com/Agrover112/Graduate_Admissions_Prediction

[3] https://www.kaggle.com/ashishpatel26/everything-about-ucla-admission-criteria.

[4] Mohan S Acharya, Asfia Armaan, Aneeta S Antony : A Comparison of Regression Models for Prediction of Graduate Admissions, IEEE International Conference on Computational Intelligence in Data Science 2019

[5] https://www.kaggle.com/mohansacharya/graduate-admissions

[6] Matthew J. Mayhew et al. (2012). Exploring Innovative Entrepreneurship and Its Ties to Higher Educational Experiences Res High Educ 53:831–859.

[7] Jill M. Norvilitis, Howard M. Reid, "Predictors of Academic and Social Success and Psychological Well-Being in College Students", Education Research International, vol. 2012, Article ID 272030, 6 pages, 2012. https://doi.org/10.1155/2012/272030

[8] https://stats.idre.ucla.edu/r/dae/probit-regression/

[9] Raghunathan, K. (n.d.). Demystifying the American Graduate Admissions Process. https://cs.stanford.edu/people/rkarthik/DAGAP.pdf

[10] Capers, Quinn IV MD; Clinchot, Daniel MD; McDougle, Leon MD; Greenwald, Anthony G. PhD Implicit Racial Bias in Medical School Admissions, Academic Medicine: March 2017 - Volume 92 - Issue 3 - p **365-369 doi: 10.1097/ACM.0000000000001388**

Appendix next page

APPENDIX

I. Code Implementation

Fig 1 : shows the general file structure of the project. This includes Jupyter notebooks use for analysis , catboost_info containing trained model files, data containing dataset , streamlit-app containing the main app include rest API in rest folder

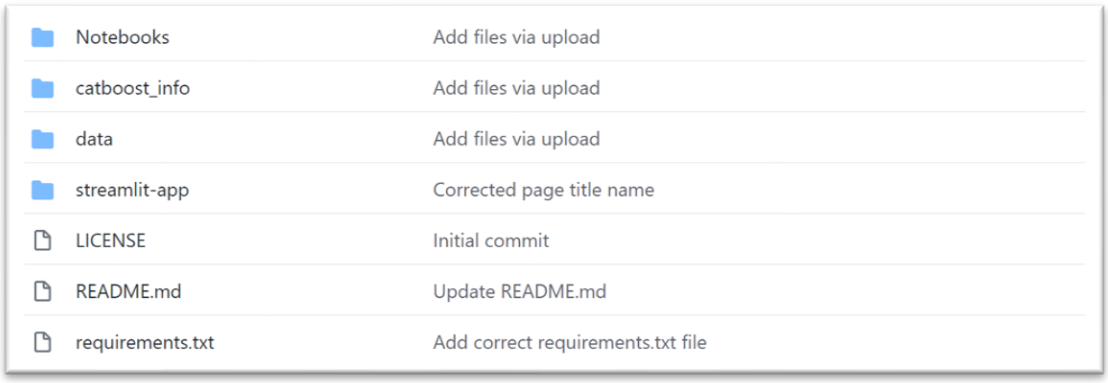


Fig. 1: File Structure

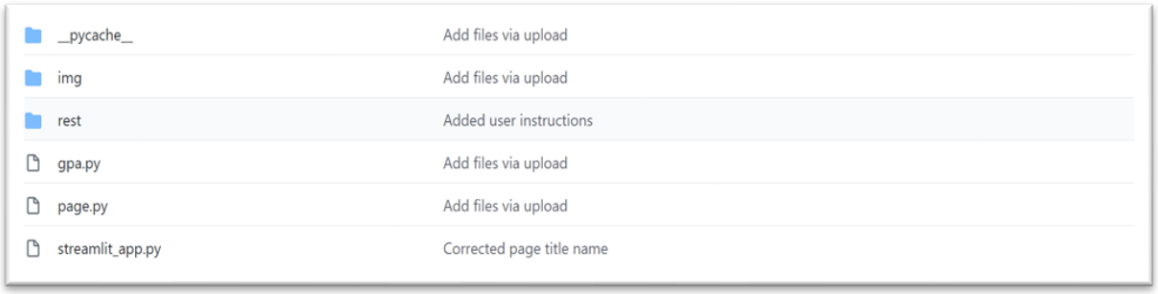


Fig. 2: Inside streamlit-app folder

II. REST API

The streamlit front-end interface the backend using a REST api which contains a pre-trained catboost model.

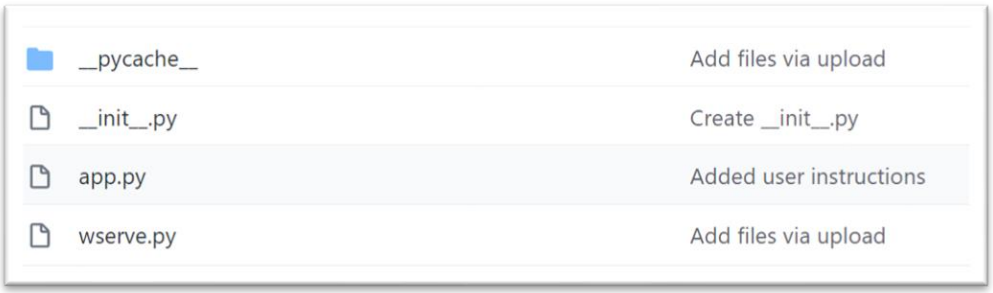


Fig. 3: Inside the rest folder.

The rest folder contains app.py which is the main API logic and tests out varies models and selects the best model based on the mean square error (MSE). A snapshot of **app.py** is shown below

```
def find_min_mse():
    """
    Applies mutiple models and prints list of models with least MSE which is our objective (i.e minimize MSE loss)
    """
    X_train,X_test, y_train, y_test=preprocess()

    models = [['DecisionTree :',DecisionTreeRegressor()],
           ['Linear Regression :', LinearRegression()],
           ['RandomForest :',RandomForestRegressor(n_estimators=150)],
           ['KNeighbours :', KNeighborsRegressor(n_neighbors = 2)],
           ['SVM :', SVR(kernel='linear')],
           ['AdaBoostClassifier :', AdaBoostRegressor(n_estimators=100)],
           ['Xgboost: ', XGBRegressor()],
           ['CatBoost: ', CatBoostRegressor(logging_level='Silent')],
           ['Lasso: ', Lasso()],
           ['Ridge: ', Ridge()],
           ['BayesianRidge: ', BayesianRidge()],
           ['ElasticNet: ', ElasticNet()],
           ]

    print("Mean Square Error...")



    res=[]
    d=[]
    for name,model in models:
        model = model
        model.fit(X_train, y_train)
        predictions = model.predict(X_test)
        res.append( np.sqrt(mean_squared_error(y_test, predictions)))
        print(name, (np.sqrt(mean_squared_error(y_test, predictions))))
        d.append([np.sqrt(mean_squared_error(y_test, predictions)),name])

    #print(sorted(res))
    for i in d[:]:
        if i[0]==sorted(res)[0]:
            print(i[1])

```

Fig. 4: Snapshot of testing process

The Catboost model shows the best MSE and is hence saved and then loaded during execution.

After training and testing our predictions, the REST api uses this saved model to serve it as a Windows background process at a given port. This is done using **wserve.py.** We use **multi-threading** to speed up performance, and serve it using **Watiress**

```
from waitress import serve
import logging
import app
from app import load_model
if __name__=="__main__":
    logging.info("** Loading Catboost model and Flask starting server...")
    logging.info("** Please wait until server has fully started")
    #print("* Loading Catboost model and Flask starting server...","please wait until server has fully started")
    load_model()
    serve(app.app,host="0.0.0.0",port="8080",threads=6)
```

III. Main Page

The main page structure is defined by the following files

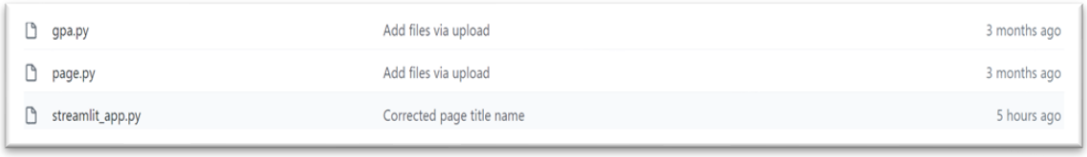
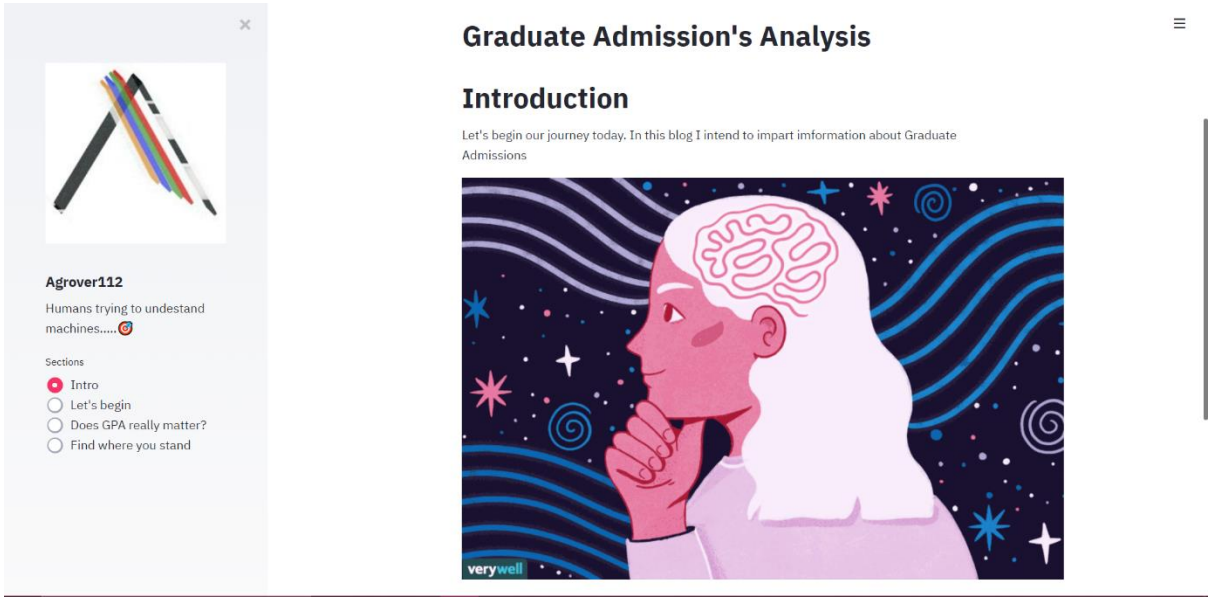
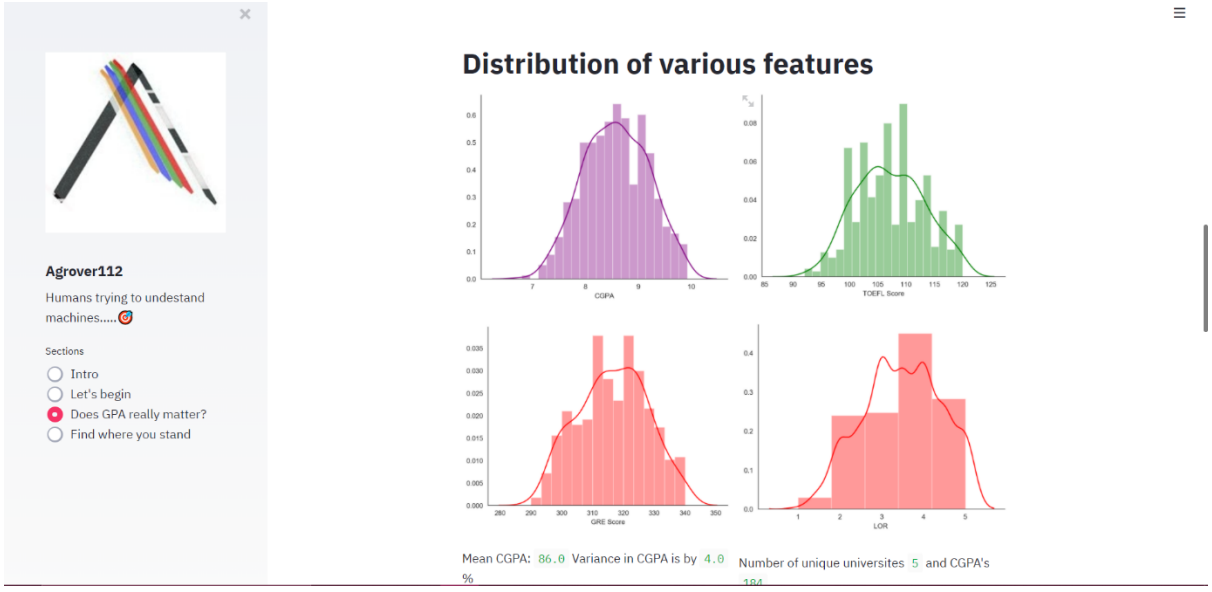


Fig. 5: Snapshot of page structure

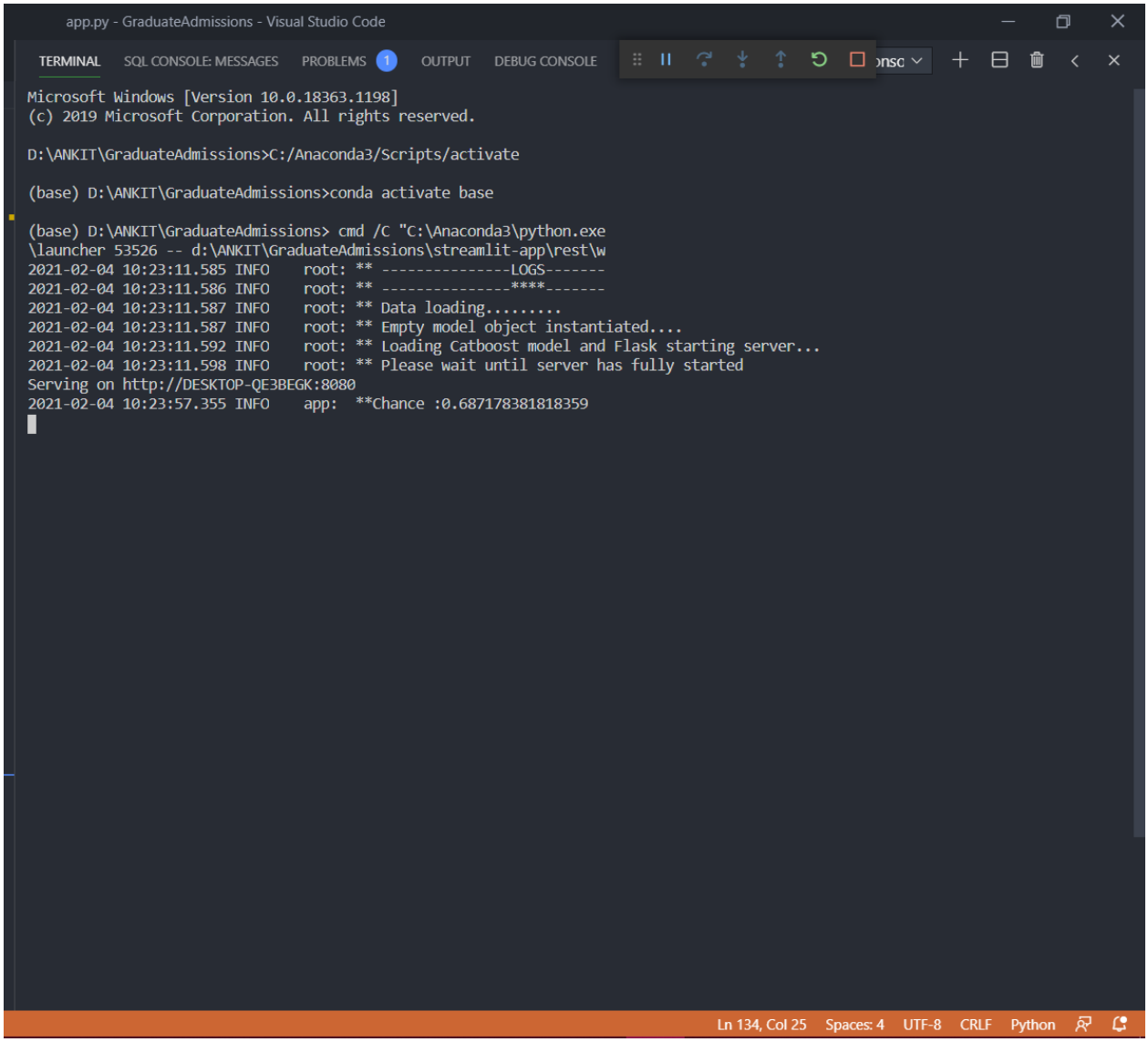
The following pages contain code for all the pages of our web-application.

OUTPUT





Logs


app.py - GraduateAdmissions - Visual Studio Code
TERMINAL
SQL CONSOLE: MESSAGES
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.
D:\ANKIT\GraduateAdmissions>C:/Anaconda3/Scripts/activate
(base) D:\ANKIT\GraduateAdmissions>conda activate base
(base) D:\ANKIT\GraduateAdmissions> cmd /C "C:\Anaconda3\python.exe
\launcher 53526 -- d:\ANKIT\GraduateAdmissions\streamlit-app\rest\w
2021-02-04 10:23:11.585 INFO root: ** ---------------LOGS-------
2021-02-04 10:23:11.586 INFO root: ** ---------------****-------
2021-02-04 10:23:11.587 INFO root: ** Data loading.....
2021-02-04 10:23:11.587 INFO root: ** Empty model object instantiated....
2021-02-04 10:23:11.592 INFO root: ** Loading Catboost model and Flask starting server...
2021-02-04 10:23:11.598 INFO root: ** Please wait until server has fully started
Serving on http://DESKTOP-QE3BEGK:8080
2021-02-04 10:23:57.355 INFO app: **Chance :0.687178381818359
Ln 134, Col 25
Spaces: 4
UTF-8
CRLF
Python