STUDYBOT: AI COUNSELLOR 2.0

Project Report submitted by

VAISHNAVI HP
(4NM20IS169)

VAISHNAVI K BHAT
(4NM20IS170)

VIJHISHA V BHANDARY
(4NM20IS177)

Under the Guidance of

Dr. Ashwini B
Associate Professor and Head of Department,
Dept. of Information Science and Engineering, NMAMIT

In partial fulfilment of the requirements for the award of the Degree

of

Bachelor of Engineering in Information Science and Engineering

from

Visvesvaraya Technological University, Belagavi

Department of Information Science and Engineering
NMAM Institute of Technology, Nitte - 574110
(An Autonomous Institution affiliated to VTU, Belagavi)

APRIL 2024

NITTE EDUCATION TRUST
N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled

"StudyBot: AI Counsellor 2.0"

is a bonafide work carried out by

Vaishnavi H P (4NM20IS169), Vaishnavi K Bhat (4NM20IS170)

and Vijhisha V Bhandary (4NM20IS177)

in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in Information Science and Engineering

prescribed by Visvesvaraya Technological University, Belagavi

during the year 2023-2024.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

_____	_____	_____
Signature of the Guide	***Signature of the HOD***	***Signature of the Principal***

Semester End Viva Voce Examination

Name of the Examiners	**Signature with Date**
1. _____	_____
2. _____	_____

ACKNOWLEDGEMENT

The satisfaction that accompanies the completion of any task would only be complete with the mention of all the people, without whom this endeavour would have been a difficult one to achieve. Their constant blessings, encouragement, guidance, and suggestions have been a constant source of inspiration.

First and foremost, my gratitude to my project guide **Dr. Ashwini B** for the constant guidance throughout this Project and for the valuable suggestions.

I also take this opportunity to express a deep sense of gratitude to the project coordinators for their valuable guidance and support.

My thanks to our beloved HOD, **Dr. Ashwini B,** for extending support in carrying out this project in the department and providing us with all the necessary facilities.

My sincere thanks to our beloved principal, **Dr. Niranjan N Chiplunkar** for permitting us to carry out this project at our college and providing us with all the necessary facilities.

Finally, thanks to the staff members of the Department of Information Science and Engineering and our friends for their honest opinions and suggestions throughout our project.

Vaishnavi H P (4NM20IS169)
Vaishnavi K Bhat (4NM20IS170)
Vijhisha V Bhandary (4NM20IS177)

ABSTRACT

The StudyBot project addresses the complexities faced by students navigating the process of studying abroad. In response to the lack of personalised guidance available to aspiring international students, StudyBot leverages AI and machine learning technology to revolutionise decision-making in university selection and admission probabilities.

Through an interface, StudyBot collects comprehensive data from students, including academic history, extracurricular involvement, exam scores, and study preferences. This data forms the basis for a platform that provides personalised recommendations and predicts admission probabilities for various universities. Utilising Python's machine learning libraries and historical admission data, StudyBot offers tailored advice to enhance users' chances of acceptance, particularly for ambitious university choices. The frontend of the StudyBot application will be developed using Flask to provide a seamless and user-friendly experience. The trained model will process user inputs to generate predictions. Additionally, StudyBot incorporates a Natural Language Processing feature enabling users to receive personalised feedback on their resumes, further enhancing their preparedness for the application process.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In an increasingly interconnected world, the allure of pursuing education overseas has become a compelling aspiration for many students. However, the path to studying abroad is riddled with uncertainties, especially when it comes to making informed decisions about universities and navigating the intricate admission processes. The StudyBot project emerges as a beacon of guidance, aiming to alleviate the challenges faced by students seeking to study abroad. It seeks to bridge the gap in personalised counselling and information dissemination, particularly in the realm of university selection and enhancing admission probabilities.

StudyBot stands as a response to the evident gap in tailored support for students embarking on their international academic journeys. The lack of accessible, personalised guidance often leaves students grappling with decisions about where to study and how to bolster their chances of admission. This project acknowledges the imperative need for a robust, AI-driven solution that harnesses user-specific data to provide individualised recommendations. By leveraging technology and data-driven insights, StudyBot aims to empower students with information crucial to their academic and professional futures, thereby simplifying the complex process of studying abroad.

At its core, StudyBot revolves around empowering students by amalgamating their academic, extracurricular, and aspirational details. Through an interactive bot interface, StudyBot will engage users in an intuitive manner to gather pertinent information. This data will be the cornerstone for the platform's ability to furnish tailored suggestions and advice. By leveraging technology to process vast amounts of information, StudyBot endeavours to offer students a clearer roadmap for their educational pursuits abroad, bridging the gap between aspiration and realization.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY ON ADMISSION PREDICTION ALGORITHMS

This topic discusses the work done by various authors, students, and researchers in a brief discussion, which is the Admission Prediction algorithms. The purpose of this section is to critically summarise the current knowledge in the field.

Table 2.1: Literature survey on Admission Prediction algorithm

SL. NO.	PAPER	AUTHORS	YEAR	OBJECTIVE
[1]	Student Admission Predictor	Himanshu Sonawane	2018	The objective of this research is to develop a Student Admission Predictor (SAP) system to assist international students in predicting their chances of admission into American universities based on their profiles, aiming to provide reliable recommendations and reduce the need for costly consultancy services, ultimately saving students time and money in the application process..

[2]	University Recommendation System for Abroad Studies using Machine Learning	Yash Kadam, Yash Kadulkar, Vishwas Moolya, Shruti Agrawal	2023	The objective of this paper is to address the challenge of finding suitable postgraduate education institutions by developing a recommendation system, leveraging historical data and machine learning algorithms, to assist graduate admission applicants in selecting universities where they have a high likelihood of acceptance based on their profiles. This system aims to save applicants time and money spent on applying to institutions beyond their reach and provides personalized recommendations to streamline the application process.
[3]	A recommender system for predicting students' admission to a graduate program using machine learning algorithms	El Guabassi, Inssaf, Zakaria Bousalem, Rim Marah, Aimad Qazdar	2021	The objective of this paper is to develop a recommender system utilizing machine learning algorithms to predict university admission early, addressing the challenge of uncertain admission requirements, with experimental findings favoring Random Forest Regression as the most effective algorithm and highlighting Cumulative Grade Point Average (CGPA) as a critical determinant for admission likelihood.
[4]	Graduate admission prediction	Aljasmi, Sara, Ali Bou Nassif, Ismail	2020	The objective of this paper is to employ machine learning models, including

	using machine learning	Shahin, Ashraf Elnagar		multiple linear regression, knn, random forest, and Multilayer Perceptron, to predict the likelihood of student admission to a master's program, providing prospective students with early insights into their acceptance chances, with experimental results highlighting the superiority of the Multilayer Perceptron model.
[5]	Predicting Student University Admission Using Logistic Regression	Paratala Rajagopal Sharan Kumar	2020	The paper aims to employ logistic regression and data analytics to predict student admission to Master's degree programs by identifying key contributing factors, facilitating better prioritization in the applicant screening process, ultimately ensuring admission for the most suitable candidates.
[6]	College Admission Prediction using Machine Learning	Monu Narnaware, Siddhesh Surve, Samarth Tandale, Saurabh Tekade, Pooja Kohok	2023	The objective of this paper is to propose a computer-aided method, to automate and simplify the college selection process for engineering students. By leveraging historical college cut-off data and analyzing students' academic performance and preferences, the system aims to predict the most suitable colleges and generate a personalized list, thereby aiding students in making informed decisions and accurately completing their admissions applications.

[7]	Prediction for University Admission using Machine Learning	Chithra Apoorva D.A, Malepati Chandu Nath,Peta Rohith, Bindushree. S	2020	The objective of this research is to develop a predictive model, named UAP, to accurately assess students' chances of admission into American universities for Masters programs based on their profiles, considering crucial factors such as GRE, TOEFL/IELTS scores, SOP, and LOR, aiming to provide reliable guidance without the need for costly consultancy services or inaccurate online applications.
[8]	Prediction Probability of Getting an Admission into a University using Machine Learning	A.Sivasangari, V. Shivani, Y. Bindhu, D. Deepa, R. Vignesh	2021	To develop a model that accurately predicts the likelihood of admission into universities based on students' profiles, leveraging linear regression, random forest, and cat boost algorithms, to provide reliable guidance and analysis of scores versus admission probability without costly consultancy services applications.

2.2 ONSITE SURVEY

Incorporating an on-site survey involving consultations with two prominent study abroad counsellors, Yocket and WiZdom Ed, yielded valuable observations crucial for guiding aspiring students. The survey revealed a notable trend: students guided by these counsellors exhibited enhanced admission prospects at universities. Additionally, these guided students showcased improved profile ratings, signifying the pivotal role of expert counselling in shaping a compelling academic profile. Moreover, the survey unveiled nuanced insights into the varying weightage assigned to different factors by distinct countries during the admission process.

Understanding these differential preferences among countries offers invaluable guidance for aspiring students, emphasizing the significance of tailored approaches when applying to diverse international institutions.

2.3 PRE-EXISTING TOOLS

In examining various pre-existing tools designed to assist students in their pursuit of studying abroad, distinct characteristics and functionalities emerged. Gyandhan, tailored specifically for the USA, notably focuses solely on exam scores and CGPA for prediction purposes, potentially limiting its predictive scope. Conversely, Yocket implements a unique approach, requiring users to rate their own resumes, possibly enabling personalized insights into application readiness. Similarly, YMGrad, akin to Gyandhan, primarily relies on exam scores and CGPA for predictions, potentially restricting the depth of analysis. An intriguing observation across these tools is the utilization of data from previously admitted students to train their algorithms, hinting at the reliance on historical success data for predictive modelling. These findings underscore the diverse functionalities and data approaches employed by existing tools, offering insights into both their strengths and potential limitations for aspiring students seeking guidance in their study abroad endeavours.

CHAPTER 3

OBJECTIVES

The objectives that are required to be completed to proceed with this new, modern way of analysing admission predictions are:

1. To Process the collected data and apply necessary feature engineering techniques to prepare it for machine learning analysis. This step involves data cleaning, transformation, and handling missing values.
2. To Utilize machine learning techniques to analyse the pre-processed data and assign weightage to each feature based on the preferences and criteria of different countries. To Create a model to predict the probability of admission for each user to specific universities.
3. To Develop a user-friendly recommendation system to collect essential details from students seeking to study abroad, including academic performance, extracurricular activities, and exam scores.
4. To Develop a Natural Language Processing Model to rate the user's resume and also provide suggestions on country selection.
5. To Integrate the frontend, backend, and machine learning components to create a seamless web application. Test the system thoroughly to ensure accuracy and reliability.

CHAPTER 4

PROBLEM DEFINITION

To address the challenges faced by students navigating the process of studying abroad due to the lack of personalized guidance, The StudyBot project aims to revolutionize decision-making in university selection and admission probabilities by leveraging machine learning technology. It seeks to bridge the gap in personalized counselling and information dissemination, empowering students with data-driven support to simplify the complex process of studying abroad and enhance their chances of admission to their preferred universities.

CHAPTER 5

SYSTEM REQUIREMENTS SPECIFICATION

5.1 SOFTWARE REQUIREMENTS

- Python
- Flask
- Pandas
- Scikit-Learn
- Pickle
- Spacy
- JSON
- PyMuPDF

5.2 HARDWARE REQUIREMENTS

- Processor: Intel Core i5 or above
- Storage: 256GB HDD or above
- RAM: 4GB RAM or above
- GPU: 4 GB or above with CUDA technology

CHAPTER 6

SYSTEM DESIGN

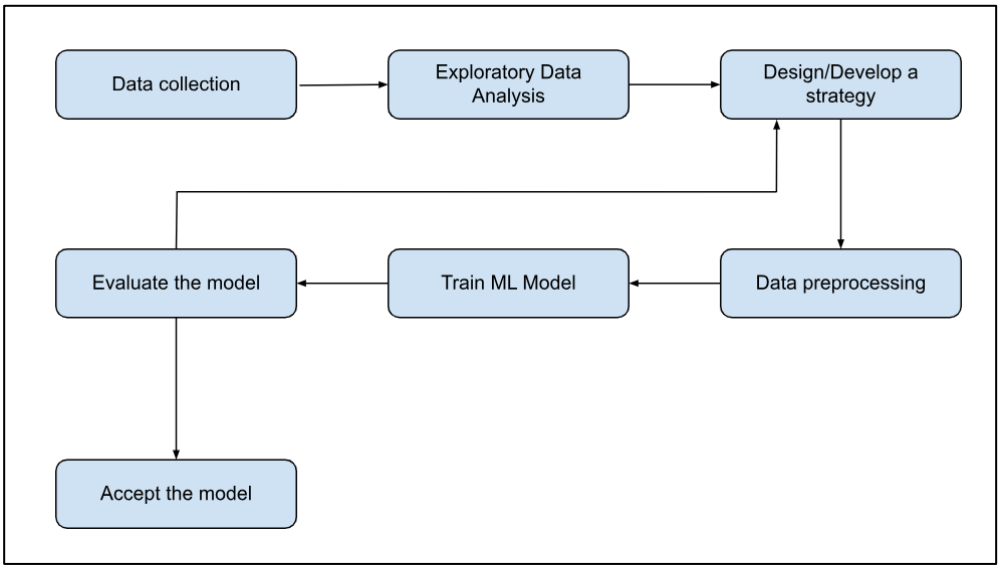

Fig 6.1: System design

Data Collection:

- Web scraping techniques were employed to extract relevant data from the Yocket website which is a platform providing information and assistance to students aspiring for higher education abroad.
- Collected data includes the following fields: Country, University, CGPA, Experience (in months), GRE score, IELTS score, TOEFL score, backlogs (yes/no), research papers, profile rating (excellent, very good, good and average), admit/reject

Exploratory Data Analysis:

- Analysed summary statistics and distributions for key variables such as CGPA, experience, GRE score, IELTS score, and TOEFL score.
- Conducted correlation analysis to explore relationships between numerical variables, identifying potential multicollinearity.
- Investigated the distribution of profile ratings and admission statuses, examining their prevalence and associations with other factors like country and university.

Design/Develop a Strategy:

- Define specific objectives and metrics for the recommendation system, informed by EDA finding.
- Outline a comprehensive plan for algorithm selection, feature engineering, and data preprocessing, ensuring alignment with the project's goals and constraints.

Data Pre-processing:

- Address missing values in fields such as Experience, GRE score, IELTS score, TOEFL score, research papers by filling with zeros indicating either lack of experience, examination not applicable for the applicant or no papers published.
- Encode categorical variables – backlogs and profile rating, into numerical representations, selecting methods such label encoding based on computational efficiency and model interpretability.

Train ML Model:

- 3 decision tree models were trained, one for each country, using the pre-processed data to predict profile ratings.

Evaluate Model Performance:

- Assess the model's performance on the testing set using evaluation metrics such as the accuracy, etc.
- Fine-tune the model based on testing set performance if necessary, adjusting hyperparameters or incorporating additional training data.

CHAPTER 7

IMPLEMENTATION

7.1 UNIVERSITY ADMISSION PREDICTOR

The project utilizes decision tree classifiers to predict profile ratings for students applying to universities in different countries, such as the United States, the United Kingdom, and Canada. The dataset contains information about applicants' academic performance (e.g., CGPA, GRE score, TOEFL score), research experience, and other relevant factors.

Data preprocessing is an essential step in preparing the dataset for model training. In this project, missing values in crucial columns, including CGPA, experience, GRE score, IELTS score, TOEFL score, and research papers, are filled with zeros to ensure data consistency and completeness. Additionally, categorical data in the 'backlogs' column is encoded into numerical labels using the LabelEncoder from the scikit-learn library.

```
[3]: df['cgpa'].fillna(0, inplace=True)
     df['experience'].fillna(0, inplace=True)
     df['gre score'].fillna(0, inplace=True)
     df['ielts score'].fillna(0, inplace=True)
     df['toefl score'].fillna(0, inplace=True)
     df['papers'].fillna(0, inplace=True)

[4]: from sklearn.preprocessing import LabelEncoder

     le = LabelEncoder()
     df['backlogs'] = le.fit_transform(df['backlogs'])
```

Fig 7.1.1: Data pre-processing

Splits the dataset into training and testing sets for each country (USA, UK, Canada) using the train_test_split function from scikit-learn. The test size is set to 20%, and random_state ensures reproducibility.

```
df_usa = df[df['Country'] == 'United States of America']
df_uk = df[df['Country'] == 'United Kingdom']
df_canada = df[df['Country'] == 'Canada']
```

Fig 7.1.2: Create Dataframes for each Country

```
from sklearn.model_selection import train_test_split

usa_train, usa_test = train_test_split(df_usa, test_size=0.2, random_state=42)
uk_train, uk_test = train_test_split(df_uk, test_size=0.2, random_state=42)
canada_train, canada_test = train_test_split(df_canada, test_size=0.2, random_state=42)
```

Fig 7.1.3: Split the dataset to training and testing data

Following train_test_split, separate Random Forest classifiers are trained for each country of interest, namely the United States, the United Kingdom, and Canada. These models are trained using relevant features such as CGPA, experience, standardized test scores (GRE, IELTS, TOEFL), backlogs, and research papers. The fit() method from scikit-learn is employed to train each decision tree classifier, associating feature data with the target variable 'profile rating' to learn the relationships between applicant attributes and admission outcomes.

```
from sklearn.ensemble import RandomForestClassifier

model_usa = RandomForestClassifier()
model_canada = RandomForestClassifier()
model_uk = RandomForestClassifier()
```

Fig 7.1.4: Random Forest Classifier models for each country

```
model_usa.fit(df_usa[['cgpa', 'experience', 'gre score', 'ielts score', 'toefl score', 'backlogs', 'papers']],
              df_usa['profile rating'])
model_uk.fit(df_uk[['cgpa', 'experience', 'gre score', 'ielts score', 'toefl score', 'backlogs', 'papers']],
             df_uk['profile rating'])
model_canada.fit(df_canada[['cgpa', 'experience', 'gre score', 'ielts score', 'toefl score', 'backlogs', 'papers']],
                 df_canada['profile rating'])
```

Fig 7.1.5: Training all the models with respective dataframes

Once trained, the decision tree models are serialized using the pickle library and saved as binary files ('model_usa.pkl', 'model_uk.pkl', 'model_canada.pkl'). Serialization enables the models to be stored efficiently and reused for making predictions on new applicant data without the need for retraining.

```
import pickle

with open('model_usa.pkl', 'wb') as f:
    pickle.dump(model_usa, f)
with open('model_uk.pkl', 'wb') as f:
    pickle.dump(model_uk, f)
with open('model_canada.pkl', 'wb') as f:
    pickle.dump(model_canada, f)
```

Fig 7.1.6: Saving the model as pickle files

7.2 RESUME CORRECTNESS CHECK

The project involves the parsing and analysis of resumes using the spaCy library in Python. The code begins by loading resume data from a JSON file named 'dataset.json'. The loaded data is stored in the 'cv_data' variable.

The 'get_spacy_doc' function processes each resume document and prepares it for training. It iterates over each document and its corresponding annotations, creating spaCy 'Doc' objects and adding entity annotations to them.

```
def get_spacy_doc(file, data):
  nlp = spacy.blank("en")
  db = DocBin()

  for text, annot in tqdm(data):
    doc = nlp.make_doc(text)
    annot = annot["entities"]

    ents = []
    entity_indices = []

    for start, end, label in annot:
      skip_entity = False
      for idx in range(start, end):
        if idx in entity_indices:
          skip_entity = True
          break
      if skip_entity == True:
        continue

      entity_indices = entity_indices + list(range(start, end))
```

```
      try:
          span = doc.char_span(start, end, label=label, alignment_mode='strict')
      except:
        continue
      if span is None:
        err_data = str([start, end]) + "    " + str(text) + "\n"
        file.write(err_data)
      else:
          ents.append(span)

    try:
      doc.ents = ents
      db.add(doc)
    except:
        pass
  return db
```

Fig 7.2.1: “get_spacy_doc” Function

The loaded resume data is split into training and testing sets using the 'train_test_split' function from scikit-learn. The training set comprises 70% of the data, while the testing set comprises the remaining 30%.

The 'get_spacy_doc' function is called for both the training and testing datasets to create spaCy 'DocBin' objects. These objects contain processed resume documents suitable for model training and evaluation. The processed training and testing data are saved to disk in spaCy format ('.spacy') at the specified paths.

The spaCy library's training command is invoked to train a named entity recognition (NER) model using the configuration file ('config.cfg') and the training and testing data. The training is performed on a GPU (specified by 'gpu-id 0') using the training data from the 'train_data.spacy' file and validation data from the 'test_data.spacy' file. The trained model is saved to the specified output directory ('./output').

```
!python -m spacy train /content/drive/MyDrive/resume_parser/config/config.cfg
--output ./output --paths.train /content/drive/MyDrive/resume_parser/model_data/train_data.spacy
--paths.dev /content/drive/MyDrive/resume_parser/model_data/test_data.spacy --gpu-id 0
```

Fig 7.2.2: Invoking spaCy training command

The trained model is loaded from the saved using the 'spacy.load' function. This model is ready to be used for parsing and analysing resume documents.

The PyMuPDF library is used to extract text from the PDF document, which is then processed by the spaCy model to identify named entities (e.g., person names, organizations) and their corresponding labels.

7.3 FLASK WEB APPLICATION

Flask web application is used for the project involving resume parsing, admission prediction, and profile rating. Here's an explanation of the main components and functionalities of the code:

The Flask routes define different endpoints of the web application, such as "/", "/resume_rating", and "/admission_predictor", and Each route corresponds to a specific HTML template file that will be rendered when the corresponding URL is accessed.

```python
@app.route("/")
def home():
    return render_template("index.html")


@app.route("/resume_rating")
def resume_rating():
    return render_template("resume_rating.html")


@app.route("/admission_predictor")
def admission_predictor():
    return render_template("admission_predictor.html")
```

Fig 7.3.1: Flask routes

7.3.1 Resume Correctness Endpoint

The "/files-upload" route handles file uploads, processes the uploaded resume file using spaCy for named entity recognition (NER), and calculates a rating based on the identified entities in the resume. The rating is determined by the presence of specific entities such as NAME, SKILLS, COLLEGE NAME, etc.

```
uploaded_file = request.files['uploaded_file']
uploaded_file.save("uploads/" + uploaded_file.filename)
fname = "uploads/" + uploaded_file.filename
nlp = spacy.load("model/model-last")
doc = fitz.open(fname)
text = "  "
for page in doc:
    text = text + str(page.get_text())

doc = nlp(text)

resume_prop_list = [ent.label_ for ent in doc.ents]
resume_prop_new_list = []
[resume_prop_new_list.append(item) for item in resume_prop_list if item not in resume_prop_new_list]
rating = 0
for i in range(len(resume_prop_new_list)):
    if (resume_prop_new_list[i] == "NAME"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "SKILLS"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "COLLEGE NAME"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "UNIVERSITY"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "DEGREE"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "YEARS OF EXPERIENCE"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "CERTIFICATION"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "EMAIL ADDRESS"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "LINKEDIN LINK"):
        rating = rating + 10
    elif (resume_prop_new_list[i] == "COMPANIES WORKED AT"):
        rating = rating + 10
```

Fig 7.3.1.1: Resume Correctness Check

7.3.2 Admission Prediction Endpoint

The "/admission-predict" route handles admission predictions based on user inputs such as country, CGPA, GRE score, etc. It loads a pre-trained model based on the selected country and predicts the admission status (ADMIT or REJECT) for a list of universities in that country.

```
i = 0
table = ""
cgpa = request.form.get("cgpa")
gre = request.form.get("gre")
ielts = request.form.get("ielts")
toefl = request.form.get("toefl")
papers = request.form.get("papers")
backlogs = request.form.get("backlogs")
experience = request.form.get("experience")

new_data = pd.DataFrame({
    'cgpa': [cgpa],
    'experience': [experience],
    'gre score': [gre],
    'ielts score': [ielts],
    'toefl score': [toefl],
    'backlogs': [1] if backlogs == 'Yes' else [0],
    'papers': [papers]
})

predictions = model.predict(new_data)[0]
```

Fig 7.3.2.1: Collecting and predicting user input

CHAPTER 8

RESULTS

8.1 EXPLORATORY DATA ANALYSIS

Visual analysis follows, utilizing Matplotlib and Seaborn libraries to generate histograms for each feature, revealing their distributions and patterns. Additionally, a correlation heatmap is constructed to visualize the relationships between variables, with annotations providing correlation coefficients. This holistic approach to data analysis aims to provide insights into the dataset's characteristics, identifying potential trends, dependencies, and areas for further investigation.

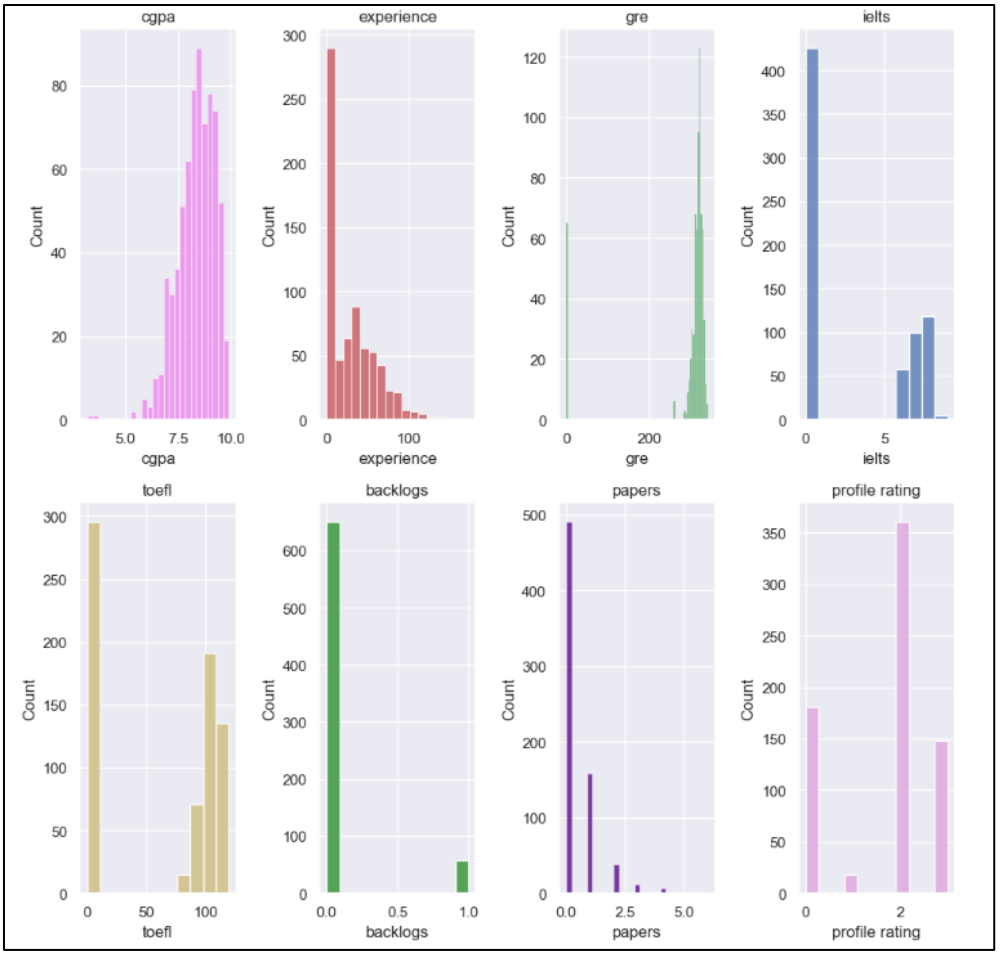

Fig 8.1.1: Distribution of data for USA dataset

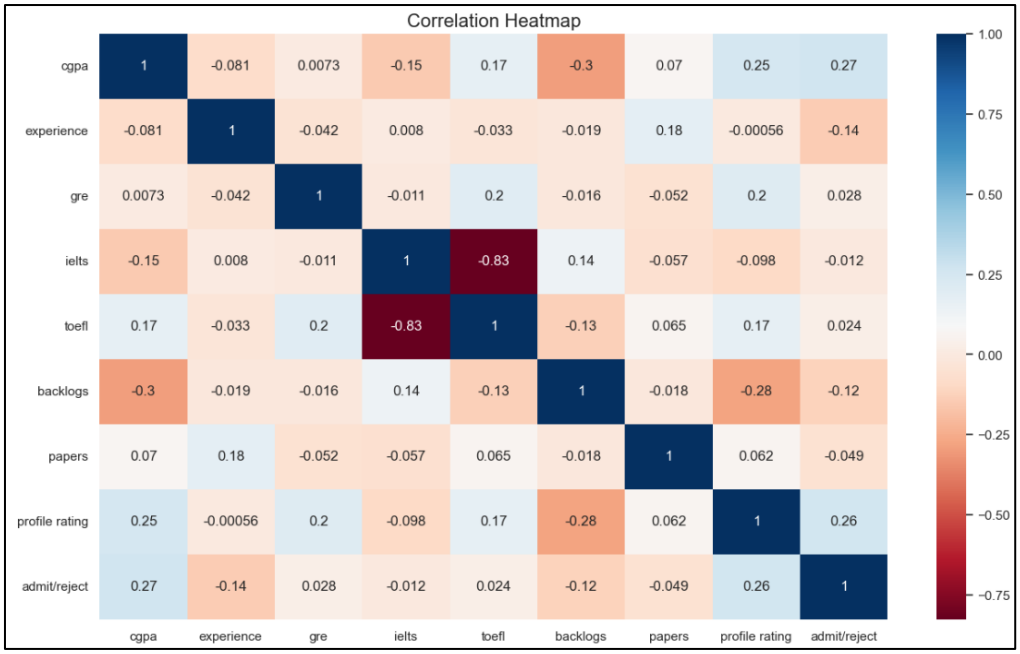

Fig 8.1.2: Correlation matrix for USA dataset

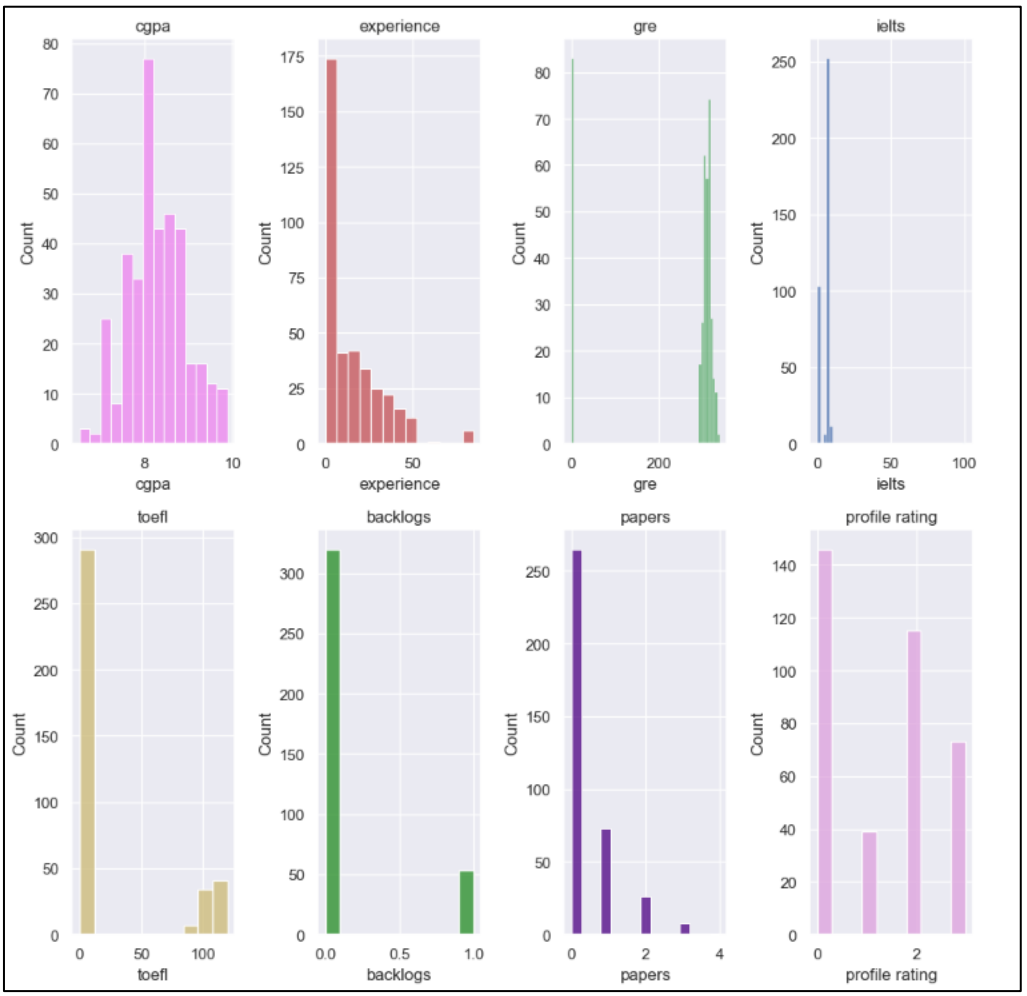

Fig 8.1.3: Distribution of data for Canada dataset

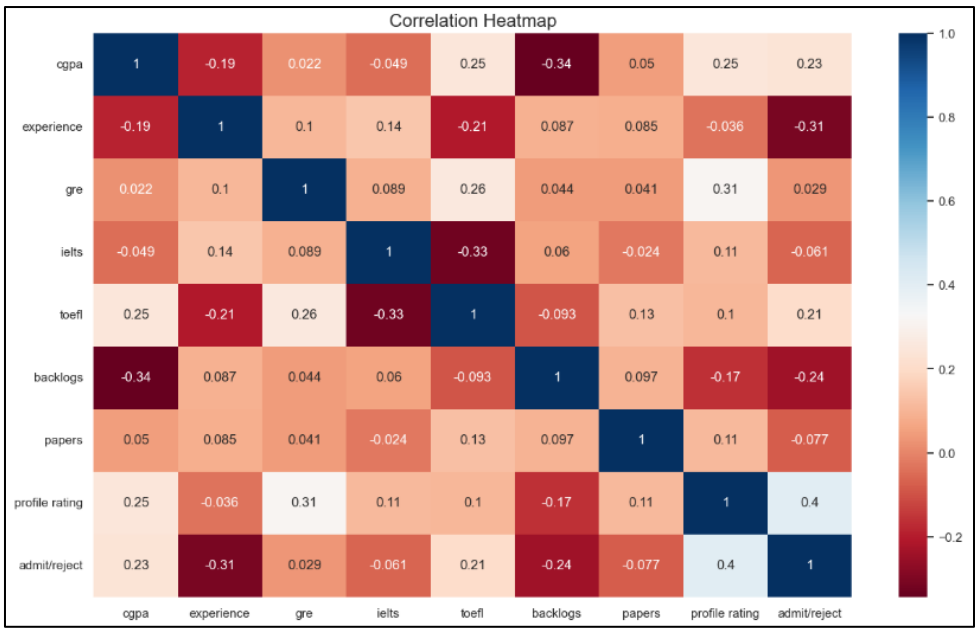

Fig 8.1.4: Correlation matrix for Canada dataset

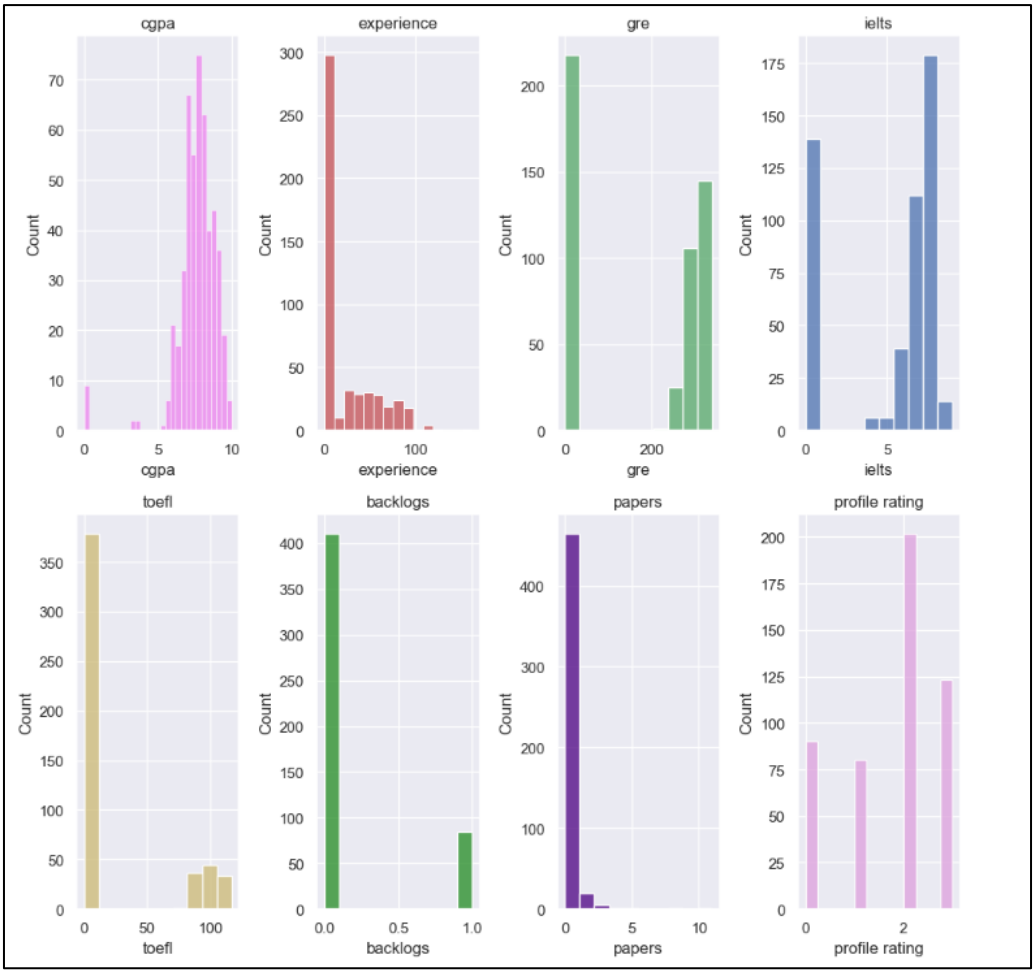

Fig 8.1.5: Distribution of data for UK dataset

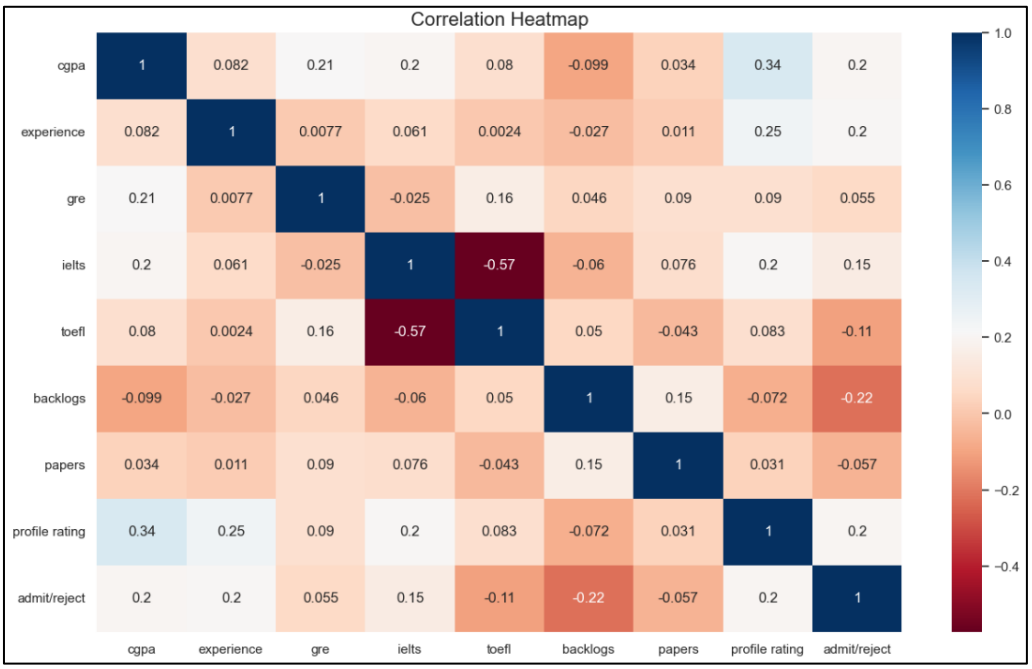

Fig 8.1.6: Correlation matrix for UK dataset

8.2 ADMISSION PREDICTION MODELS

The admission prediction models achieved varying levels of accuracy across different datasets:

- For the USA dataset, the model achieved an accuracy of approximately 52.82%.
- For the UK dataset, the model demonstrated higher accuracy, achieving around 72.73%.
- Similarly, for the Canada dataset, the model performed well, with an accuracy of approximately 74.67%.

```
Accuracy for USA dataset: 0.528169014084507
Accuracy for UK dataset: 0.7272727272727273
Accuracy for Canada dataset: 0.7466666666666667
```

Fig 8.2.1: Accuracies of the different models

These accuracy scores indicate the model's ability to predict admission outcomes based on input features such as CGPA, standardized test scores, experience, backlogs, and papers. The relatively higher accuracy for the UK and Canada datasets suggests that the model may generalize better to admissions processes in those countries compared to the USA.

8.3 WEB APPLICATION USING FLASK AND HTML

8.3.1 Index

The index page of the Flask application serves as the central hub from which users can access various functionalities, including the admission predictor and resume correctness checker.

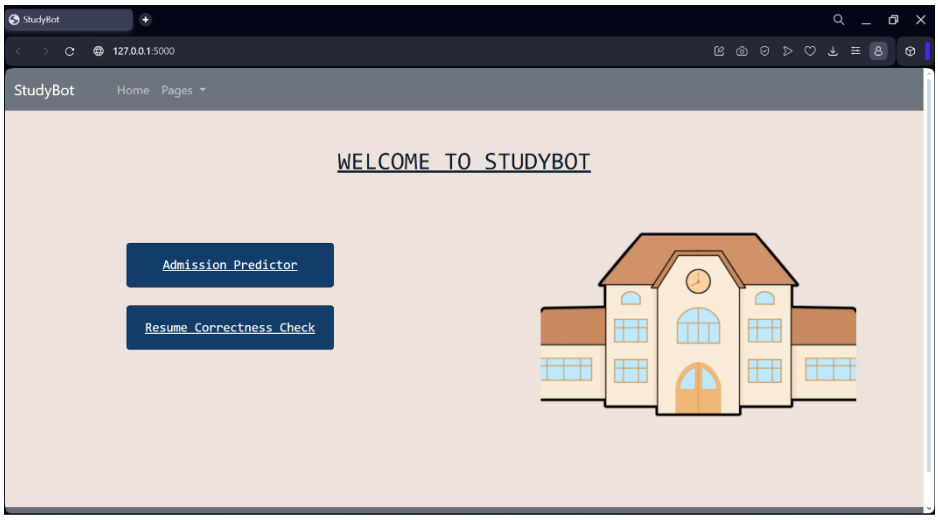

Fig 8.3.1.1: Index

8.3.2 Admission predictor page

The “Admission Predictor” link directs users to a dedicated page where they can input their academic credentials, such as CGPA, standardized test scores, and other relevant details, along with their desired country for higher education. After submitting the necessary information, the application leverages pre-trained machine learning models to predict the admission status for a selection of universities within the chosen country.

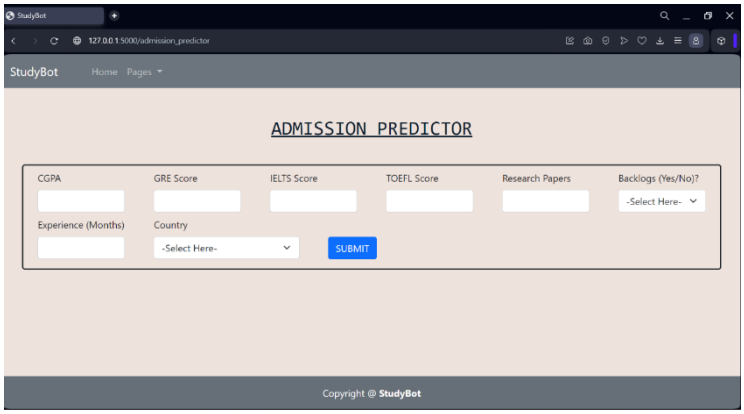

Fig 8.3.2.1: Admission predictor requiring input

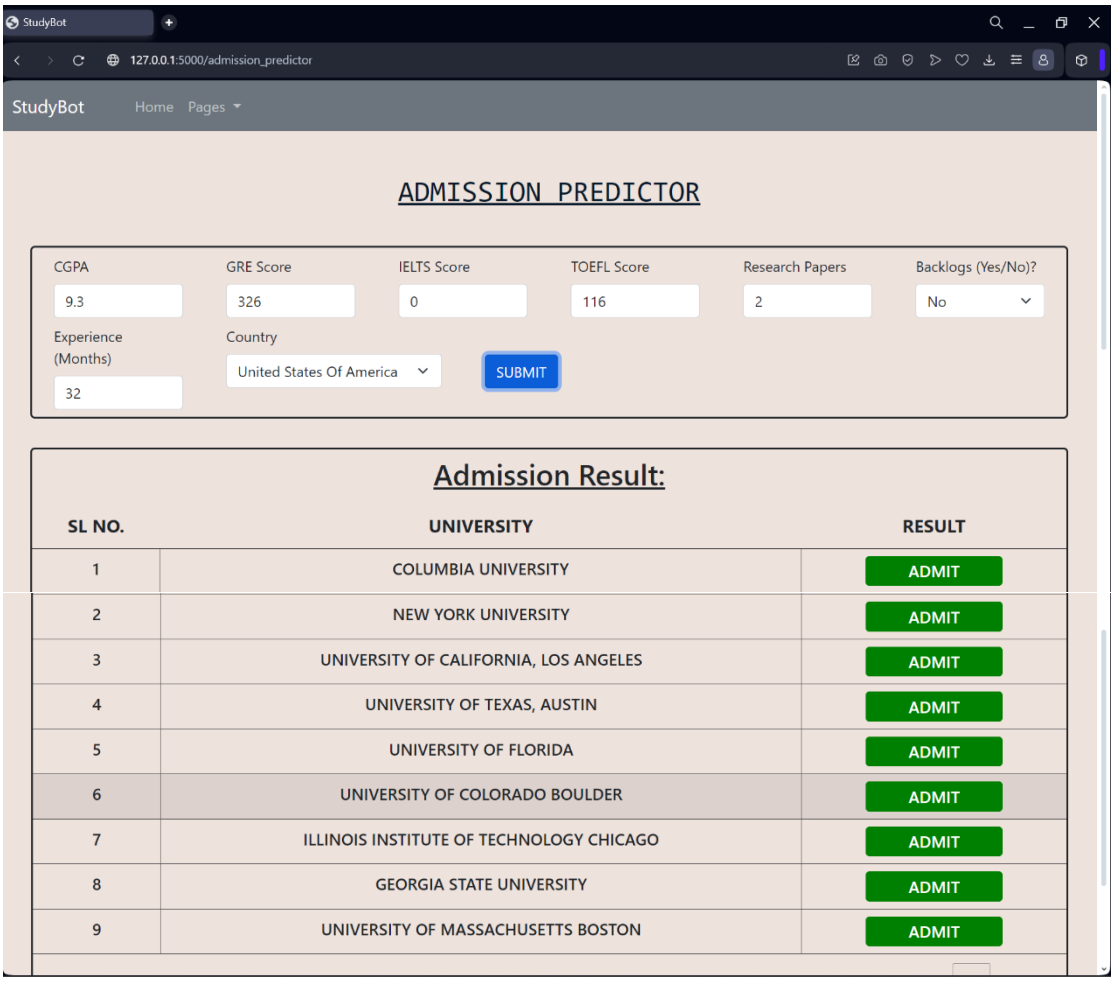

Fig 8.3.2.2: Admission predictor results

8.3.3 Resume correctness page

The Resume Correctness Checker link leads users to a separate page where they can upload their resumes for analysis. The application utilizes natural language processing techniques to parse the contents of the resume, identifying key entities such as the applicant's name, skills, educational background, and work experience. Based on this analysis, the application computes a rating reflecting the completeness and quality of the resume.

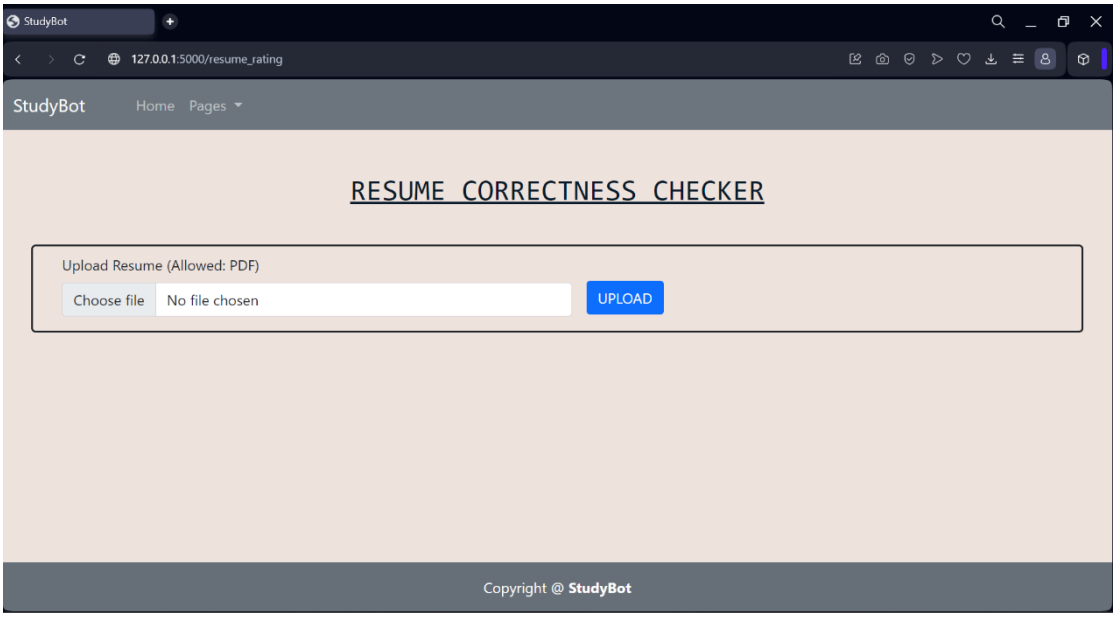

Fig 8.3.3.1: Resume Correctness Checker requiring input

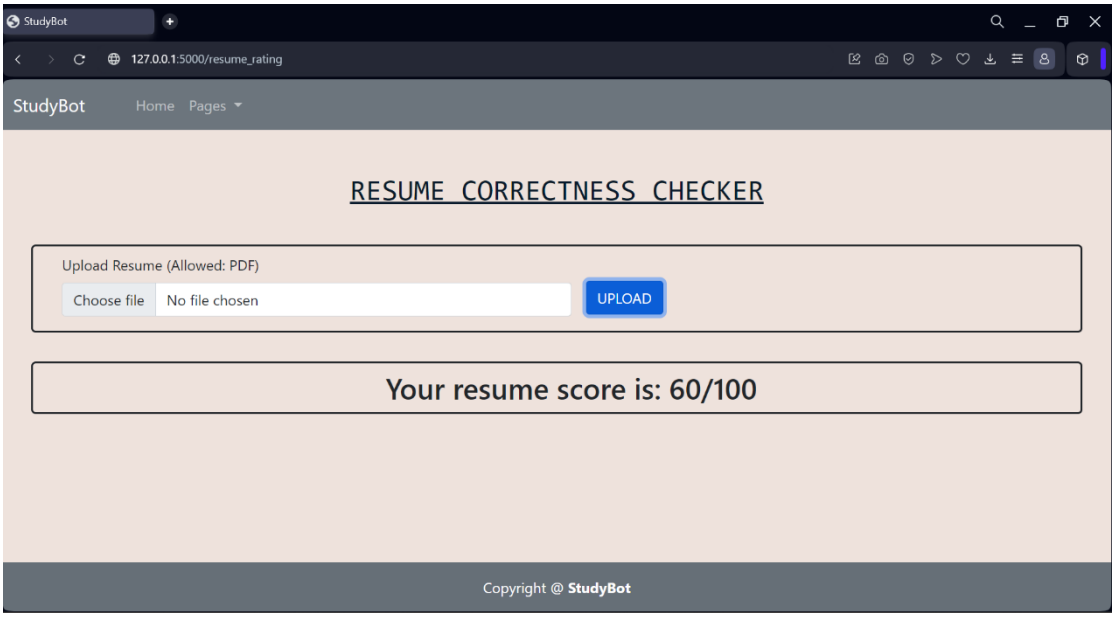

Fig 8.3.3.2: Resume Correctness Checker output

CONCLUSION

The StudyBot project presents a significant step forward in addressing the complexities and challenges encountered by students seeking to study abroad. By harnessing the power of machine learning technology, StudyBot offers personalized guidance and predictive insights to aid students in university selection and admission processes.

Through its interface, StudyBot collects and analyses comprehensive data from students, including academic history, extracurricular involvement, and study preferences. This wealth of information serves as the foundation for the platform to generate tailored recommendations and predict admission probabilities for various universities.

The implementation of Python's machine learning libraries and historical admission data enables StudyBot to provide users with chance to enhance their chances of acceptance, particularly for ambitious university choices. Furthermore, the development of the StudyBot frontend using Flask ensures a seamless and user-friendly experience for students navigating the platform.

Incorporating a Natural Language Processing feature further enhances StudyBot's capabilities, allowing users to work on their resumes and bolstering their preparedness for the application process.

In conclusion, StudyBot emerges as a valuable tool for students embarking on their international academic journeys, empowering them with data-driven support and bridging the gap between aspiration and realization. By simplifying the complex process of studying abroad and providing personalized guidance, StudyBot contributes to making the dream of international education more accessible and achievable for aspiring students .

SCOPE FOR FUTURE WORK

Expansion of Geographic Coverage and University Database

To further enhance StudyBot's utility and global relevance, future developments will prioritize expanding the platform's geographic coverage to include a broader spectrum of study destinations beyond the current focus areas of the USA, UK, and Canada. This expansion will entail the incorporation of comprehensive databases comprising universities and educational institutions from diverse regions worldwide. By broadening the platform's reach, StudyBot aims to cater to the evolving preferences and aspirations of a more diverse international student population, facilitating informed decision-making regarding higher education options across various continents.

Integration of Advanced Resume Enhancement Tools

In future iterations, StudyBot can integrate advanced resume enhancement tools powered by artificial intelligence (AI) and natural language processing (NLP) technologies. These tools will go beyond basic resume analysis to provide users with personalized recommendations for improving their resumes. By leveraging sophisticated algorithms, StudyBot will offer insights into content optimization, formatting, and keyword utilization, helping users create compelling resumes that effectively showcase their qualifications and experiences. This enhancement will further enhance StudyBot's utility as a comprehensive resource for students seeking to enhance their academic and professional profiles

REFERENCES

[1] **Sonawane, Himanshu (2017)** "Student admission predictor." Unpublished doctoral thesis, Dublin, National College of Ireland.

[2] **Kadam, Y., Kadulkar, Y., Moolya, V. and Agrawal, Shruti (2023)** "University Recommendation System for Abroad Studies using Machine Learning." International Journal for research in Applied sciences and engineering technology, IJRASET50835

[3] **Guabassi, E., Zakaria, I., Rim Marah, B., and Qazdar, A. (2021)** "A recommender system for predicting students' admission to a graduate program using machine learning algorithms." International Journal of Online and Biomedical Engineering (iJOE). 17. 135. 10.3991/ijoe.v17i02.20049.

[4] **Aljasmi, Sara, Nassif, Ali, Shahin, Ismail and Elnagar, Ashraf. (2020)** “Graduate Admission Prediction Using Machine Learning”. Unpublished doctoral thesis, 14. 10.46300/91013.2020.14.13.

[5] **Rajagopal, Sharan Kumar (2020) “**Predicting Student University Admission Using Logistic Regression”, European Journal of Computer Science and Information Technology, Vol.8, No.3, pp.46-56

[6] **Narnaware, M., Surve, S., Tandale, S., Tekade, S., and Kohok, Pooja (2023) “**College Admission Prediction using Machine Learning” International Journal for research in Applied science and engineering Technology, IJRASET56539.

[7] **D.A, Chithra, Nath, Malepati, Rohith, Peta, S, Bindushree, S, Swaroop. (2020). "**Prediction for University Admission using Machine Learning". International Journal of Recent Technology and Engineering (IJRTE). 8. 4922-4926. 10.35940/ijrte.F9043.038620.

[8] **Sivasangari, A., Shivani, V., Bindhu, Y., Deepa, D., and Vignesh, R. (2021)** "Prediction Probability of Getting an Admission into a University using Machine Learning," 5th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2021, pp. 1706-1709, doi: 10.1109/ICCMC51019.2021.9418279.