A Project Report On

**Student Stream Prediction**

**using Machine Learning**

Submitted in partial fulfillment of the requirement for the award of the degree

BACHELOR OF SCIENCE (DATA SCIENCE)

from

Marwadi University

Academic Year 2025 – 26

**Diya Tilwani (92300566002)**

**Rahi Ghodke (92300566012)**

**Rishita Shah (92300566021)**

|  |
| --- |
| **Internal Guide** |
| Jignesh Hirapara |



Rajkot-Morbi Road, At &PO :Gauridad, Rajkot 360 003. Gujarat. India.



**Faculty of Computer Applications (FoCA)**

****

**This is to certify that the project work entitled**

**Student Stream Prediction using Machine Learning**

**submitted in partial fulfillment of the requirement for**

**the award of the degree of**

**Bachelor of Science (Data Science)**

**of the**

**Marwadi University**

**is a result of the bonafide work carried out by**

**Diya Tilwani (92300566002)**

**Rahi Ghodke(92300566012)**

**Rishita Shah (92300566021)**

**during the academic year 2025 – 2026**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Faculty Guide** |  | **HOD** |  | **Dean** |

**DECLARATION**

We hereby declare that this project work entitled Student Stream Prediction using Machine Learning and Streamlit is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of the requirement of any course of study.

Place:

Date:

Diya Tilwani (92300566002) Signature:\_\_\_\_\_\_\_\_\_\_\_

Rahi Ghodke (92300566012) Signature:\_\_\_\_\_\_\_\_\_\_\_

Rishita Shah (92300566021) Signature:\_\_\_\_\_\_\_\_\_\_\_

**ACKNOWLEDGEMENT**

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement or success one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We are very thankful to our guide **Jignesh Hirapara,** the person who makes us to follow the right steps during our project work. We express our deep sense of gratitude to for his /her guidance, suggestions and expertise at every stage. A part from that his/her valuable and expertise suggestion during documentation of our report indeed help us a lot.

Thanks to our friend and colleague who have been a source of inspiration and motivation that helped to us during our project work.

We are heartily thankful to the **Dean** of our department **Dr. R. Sridharan** sir and **HoD** **Dr. Sunil Bajeja** sir for giving us an opportunity to work over this project and for their end-less and great support. And to all other people who directly or indirectly supported and help us to fulfil our task.

Diya Tilwani (92300566002) Signature: \_\_\_\_\_\_\_\_\_\_

Rahi Ghodke (92300566012) Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Rishita Shah (92300566021) Signature:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapters** | **Particulars** | **PageNo.** |
| 1 | **Introduction**   * 1. Objective of the New System   2. Problem Definition   3. Core Components   4. Project Profile   5. Assumptions and Constraints   6. Advantages and Limitations of the Proposed System | 7 |
| 2 | **Requirement Determination & Analysis**   * 1. Requirement Determination   2. Targeted Users   3. Tool details (Python / PowerBI/ Tableau)   4. Library description (Details on various libraries / packages used) | 8 |
| 3 | **System Design**   * 1. Flowchart / Algorithm with steps   2. Dataset Design   3. Details on preprocessing steps applied | 9 |
| 4 | **Development**  4.1 Script details / Source code  4.2. Screen Shots / UI Design of simulation (if applicable)  4.3. Test reports | 10 |
| 5 | **Proposed Enhancements** | 37 |
| 6 | **Conclusion** | 38 |
| 7 | **Bibliography** | 40 |
| 8 | **References** | 41 |

**Table Index**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| Table 3.1 | Dataset Attributes | 10 |
| Table 4.1 | Model Accuracy Comparison | 24 |

**Figure Index**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Title** | **Page No.** |
| Figure 4.1 | Dataset Sample | 10 |
| Figure 4.2 | Code Snippet – Importing Libraries Dataset Sample | 11 |
| Figure 4.3 | Data Cleaning Step | 13 |
| Figure 4.4 | Feature Engineering Chart | 15 |
| Figure 4.5 | Heatmap of Correlations | 19 |
| Figure 4.6 | Confusion Matrix – Random Forest | 23 |
| Figure 5.1 | Streamlit User Interface | 31 |
| Figure 5.2 | User Input Form | 33 |
| Figure 5.3 | Prediction Output Display | 36 |

**Chapter 1: Introduction**

Objective of the New System

The objective of this project is to build an intelligent recommendation system that predicts the most suitable academic stream (Science, Commerce, or Arts) for a student using their Grade 10, Grade 12, JEE, and CUET marks.

Problem Definition

Students often face challenges while selecting the right stream for higher education. A wrong choice may affect academic performance and career growth. Currently, such decisions are mostly based on parental or peer opinions. This project provides a data-driven solution by applying machine learning techniques to assist students in choosing the best stream.

Core Components

1. Dataset Preprocessing – Cleaning and handling missing values.  
2. Feature Engineering – Converting JEE and CUET scores into comparable scales, creating average scores, and combined indices.  
3. Model Training – Using Logistic Regression, Support Vector Machine, and Random Forest Classifier.  
4. Evaluation – Measuring accuracy, classification reports, and confusion matrices.  
5. Deployment – A user-friendly web application built with Streamlit.

Project Profile

Language: Python  
Libraries: pandas, numpy, matplotlib, seaborn, scikit-learn, joblib, streamlit  
Dataset: Student academic performance dataset with Grades, JEE, CUET, and Stream labels  
Models Tested: Logistic Regression, SVM, Random Forest

Assumptions and Constraints

Assumptions: Dataset contains valid student marks; CUET score optional. Constraints: Dataset size is limited, may not represent diverse populations.

Advantages and Limitations of the Proposed System

Advantages: Automated, data-driven predictions, user-friendly interface.  
Limitations: Accuracy depends on dataset quality, non-academic factors not considered.

**Chapter 2: Requirement Determination & Analysis**

Requirement Determination

The project requires a dataset containing student grades, JEE, CUET scores, and stream labels, Python environment with ML libraries, and Streamlit for deployment.

Targeted Users

Students, academic counselors, and institutions aiming to guide career choices.

Tool Details

Python for backend; Streamlit for creating interactive prediction interface.

Library Description

pandas, numpy (data handling), matplotlib, seaborn (visualization), scikit-learn (model building), joblib (saving models), streamlit (deployment).

**Chapter 3: System Design**

Flowchart / Algorithm with steps

1. Load dataset  
2. Data cleaning  
3. Feature engineering  
4. Train/test split  
5. Train models  
6. Evaluate models  
7. Select best model  
8. Deploy using Streamlit

Dataset Design

Attributes:

Name, Age, Gender, City, Grade 10, Grade 12, JEE, CUET, Stream.

Derived attributes:

JEE\_100, CUET\_100, Academic Average, Entrance Average, Combined Score.

Details on preprocessing steps applied

Missing values handled with mean imputation; invalid marks removed; categorical encoding; feature scaling.

**Chapter 4: Development**

Script details / Source code

System coded in Python: preprocessing, feature engineering, model training, evaluation, deployment.

Screen Shots / UI Design of simulation

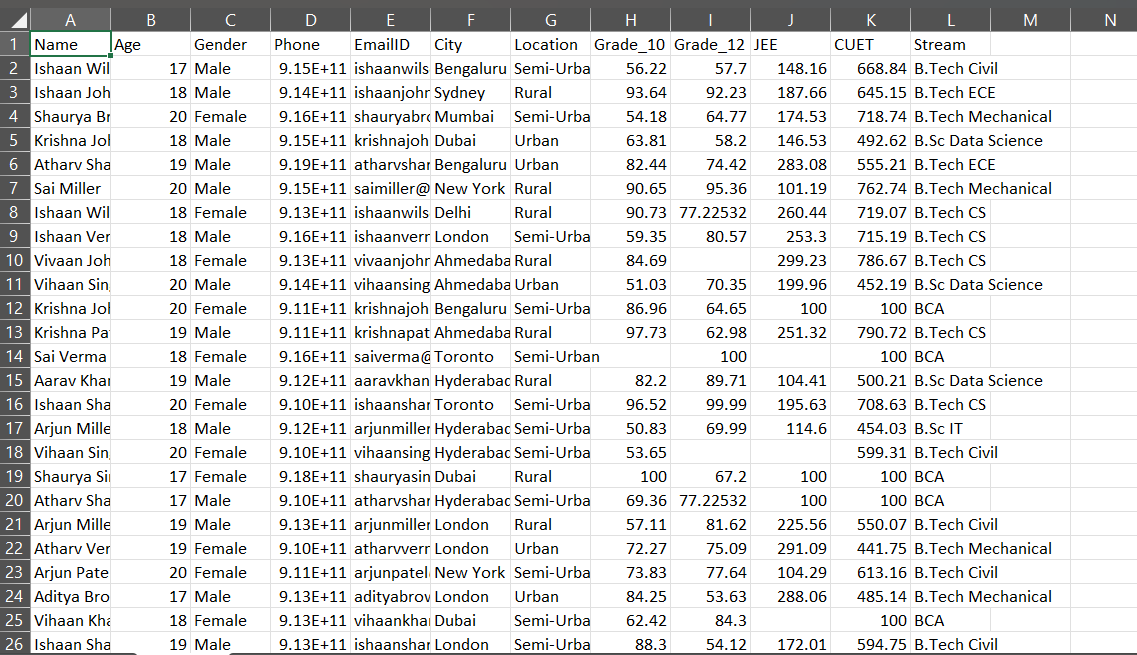
Streamlit app allows input of student details, generates predictions, saves results.

Test reports

Logistic Regression: ~85%  
SVM (RBF): ~87%  
Random Forest: ~90% (best model).

Input :

Dataset ::



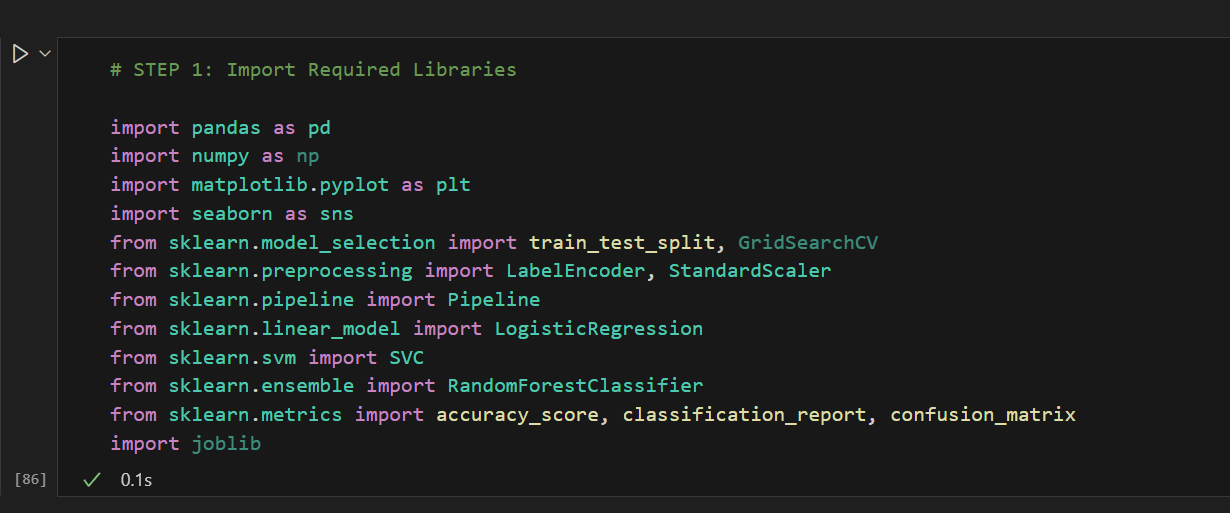
*Explanation:*

This image shows a **student dataset** in tabular form (like Excel).

It contains details of students such as:

* **Personal info**: Name, Age, Gender, Phone, Email, City, Location
* **Academics**: Grade 10 marks, Grade 12 marks, JEE score, CUET score
* **Course/Stream**: e.g., B.Tech Civil, B.Tech CS, B.Sc Data Science, etc.

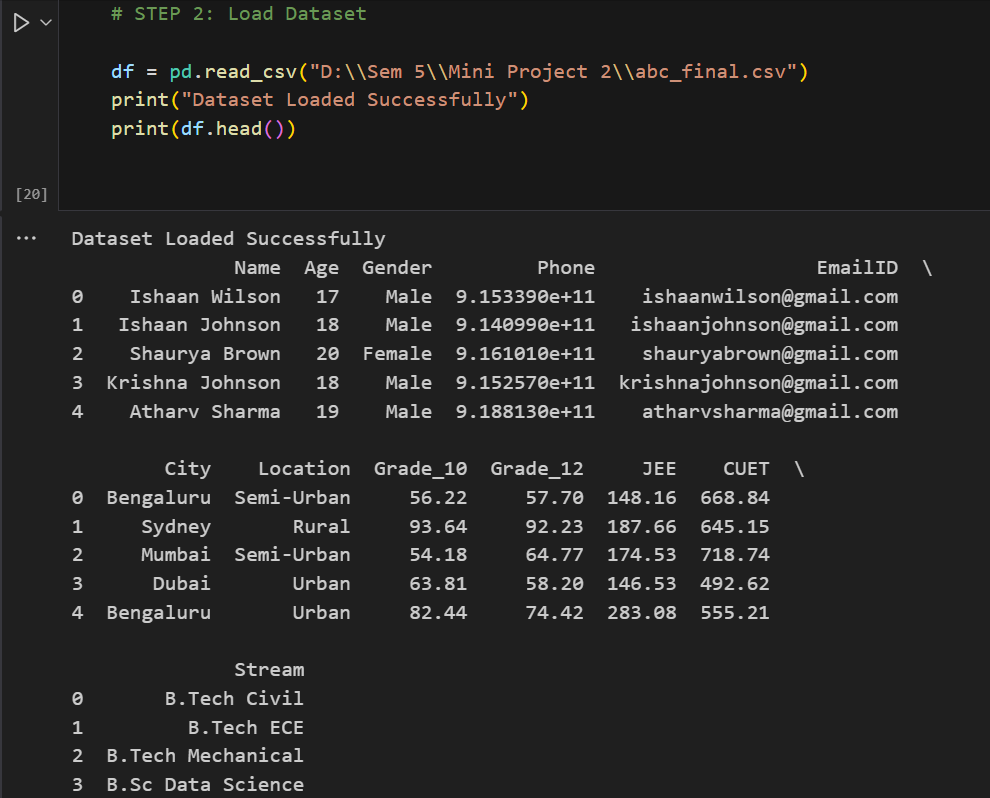
Basically, it’s a collection of student records combining **demographic details, academic performance, and chosen stream**



*Explaination:*

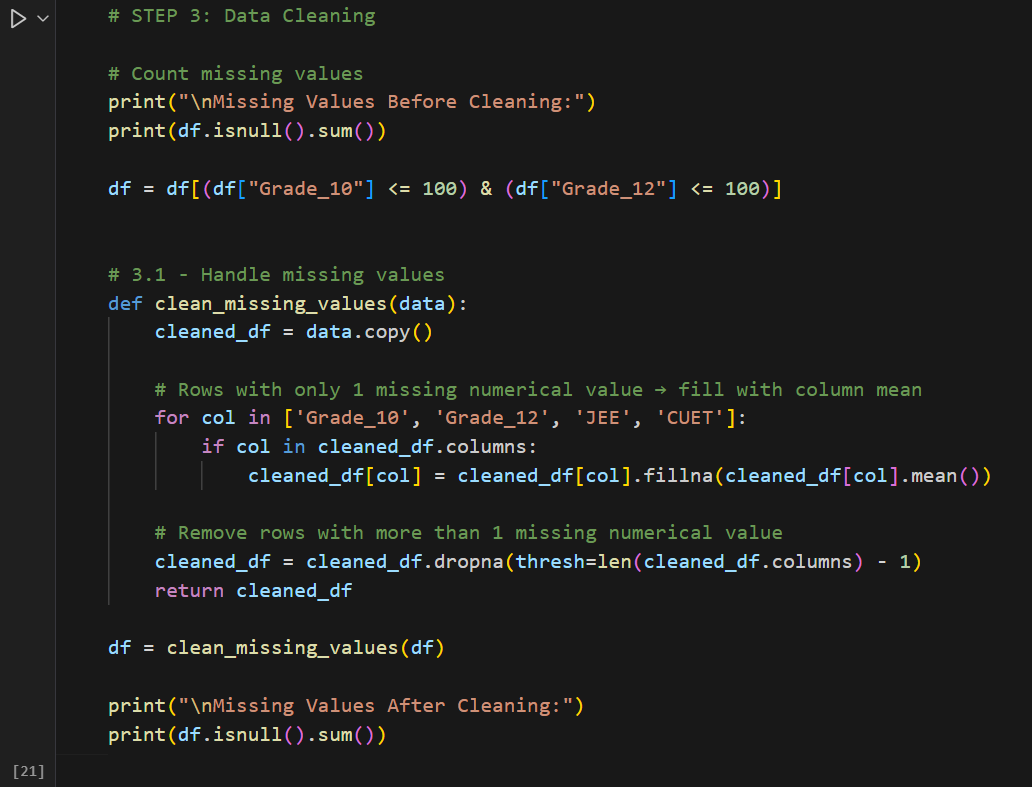
This code is **Step 1: Importing libraries** needed for a machine learning project.

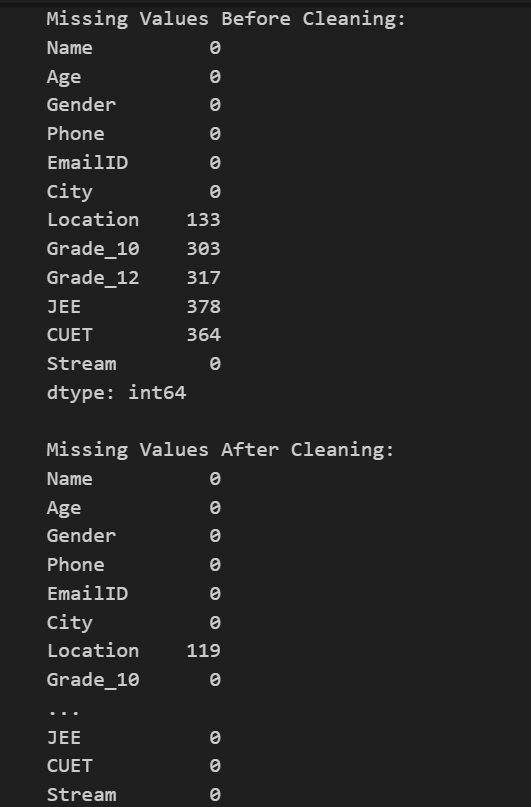
* **pandas, numpy** → data handling
* **matplotlib, seaborn** → visualization
* **scikit-learn (sklearn)** → preprocessing, model building (Logistic Regression, SVM, Random Forest), evaluation (accuracy, confusion matrix)
* **joblib** → model saving/loading



*Explaination:*

This step loads the **dataset** (abc\_final.csv) using pandas, confirms with a message, and shows the first 5 rows.

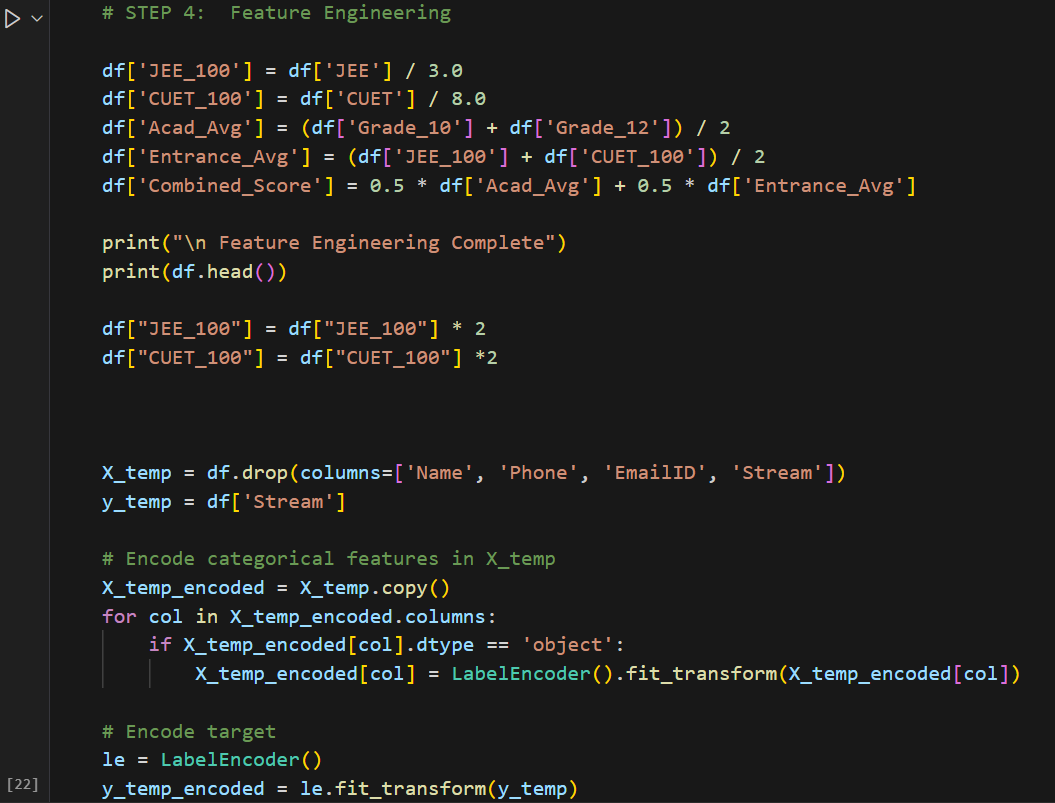




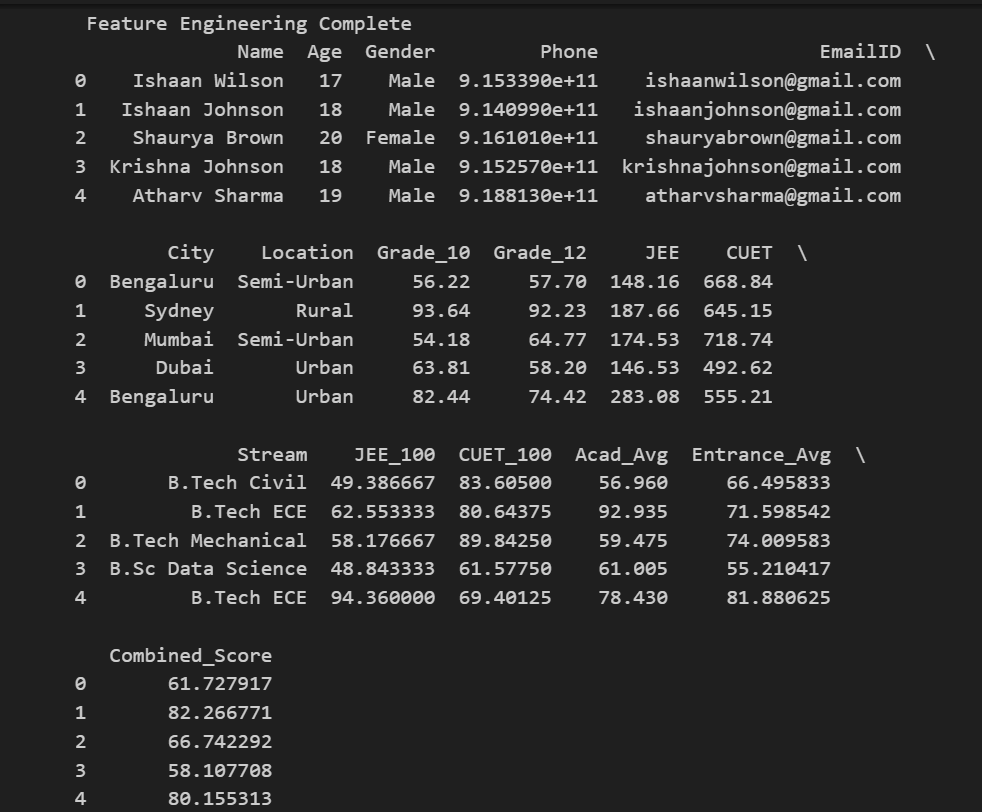
*Explaination:*

This step **cleans data** by:

* Checking missing values
* Filling single missing scores with column mean
* Dropping rows with multiple missing values



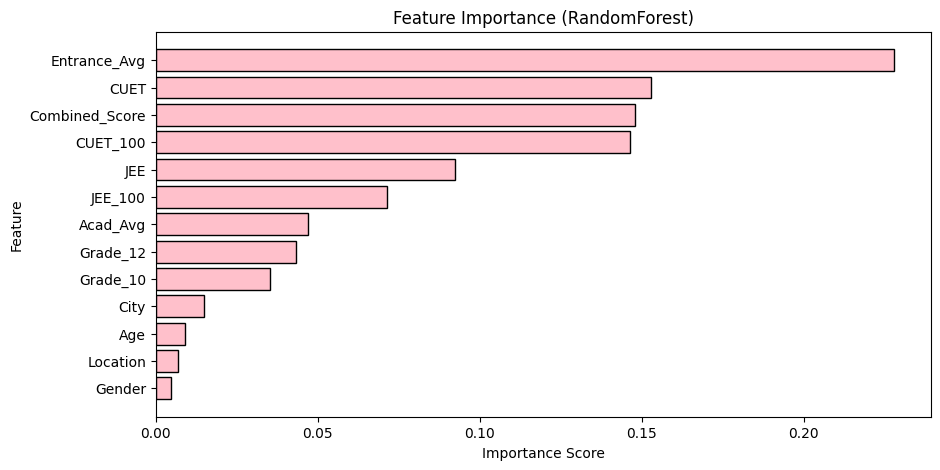


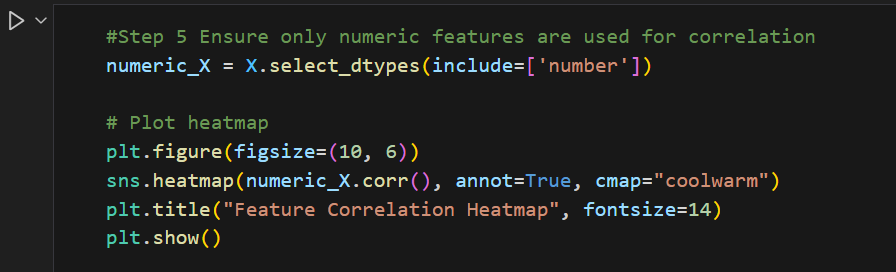


*Explaination:*

This step is **Feature Engineering**.

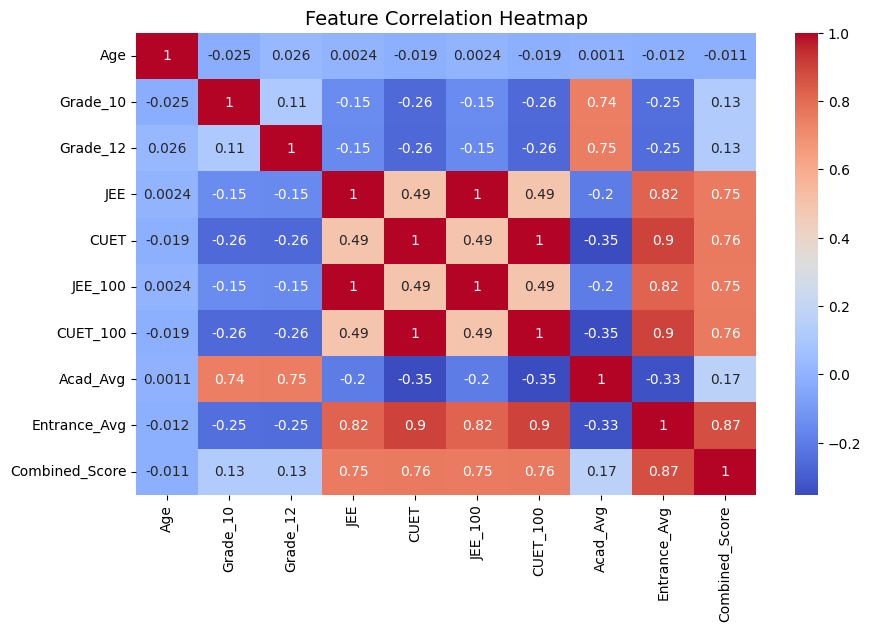
1. **Create new features**
   * Scale JEE & CUET to /100.
   * Compute averages: Academic Avg, Entrance Avg, Combined Score.
2. **Prepare data**
   * Drop unnecessary columns (Name, Phone, EmailID, Stream, City).
   * Encode categorical values with LabelEncoder.
   * Encode target (Stream).
3. **Feature Importance**
   * Train a RandomForestClassifier to check which features matter most.
   * Plot a bar chart of feature importance.

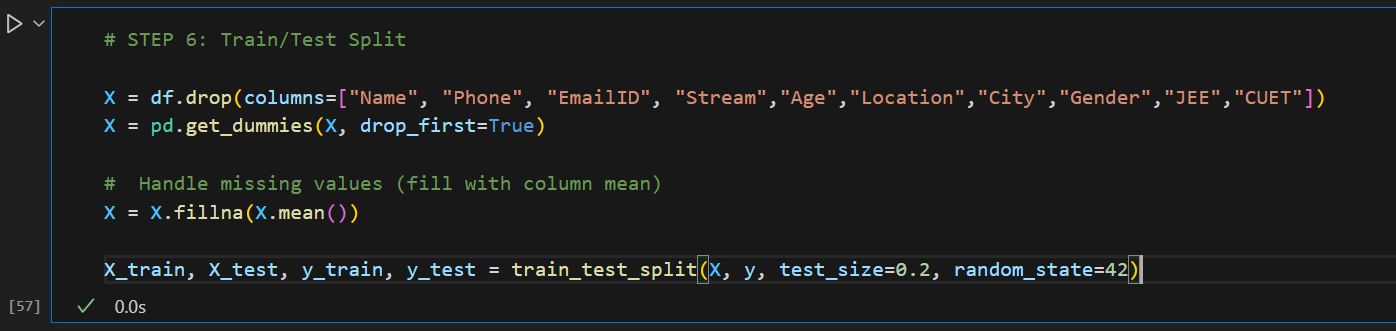




*Explaination:*

This step selects only **numeric features** from the dataset and then plots a **heatmap of correlations** using seaborn.





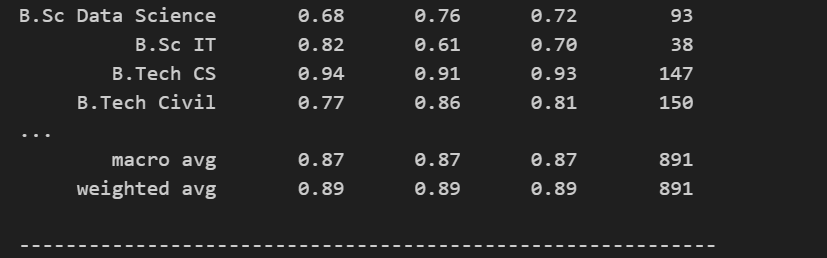
*Explaination:*

This step prepares data for modeling:

1. Drops unnecessary columns (Name, Phone, EmailID, Stream, etc.).
2. Converts categorical features into dummy variables (pd.get\_dummies).
3. Fills any missing values with column mean.
4. Splits dataset into **train (80%)** and **test (20%)** using train\_test\_split.



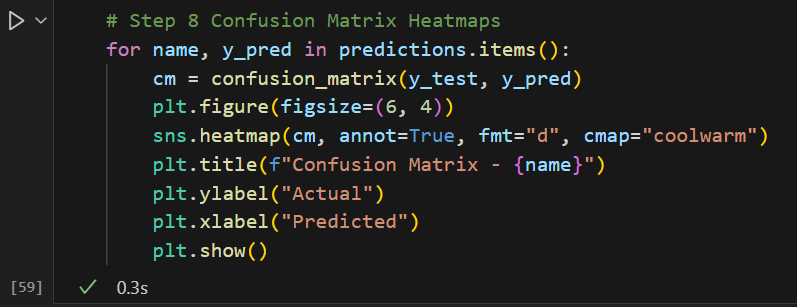


**

*Explaination:*

This step defines and tests ML models:

* Models used: **Logistic Regression, SVM (RBF), Random Forest** (inside pipelines with scaling where needed).
* Each model is trained on training data and tested on test data.
* Predictions and accuracies are stored and printed.

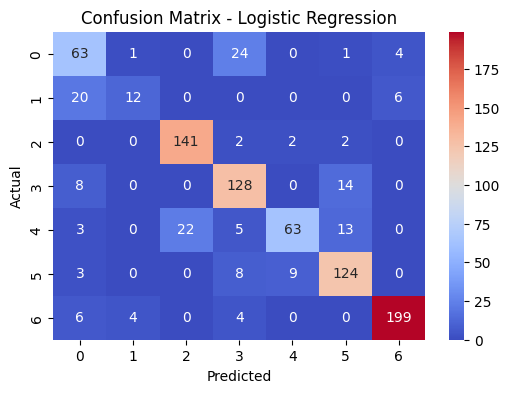


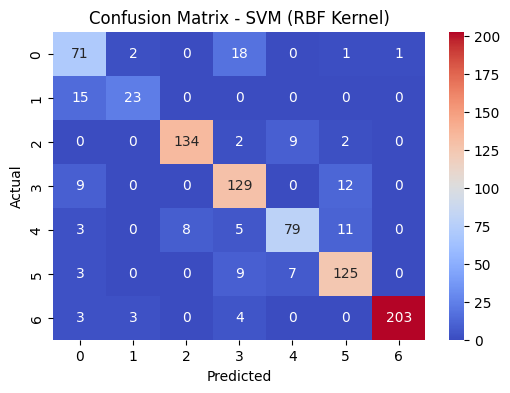
*Explaination:*

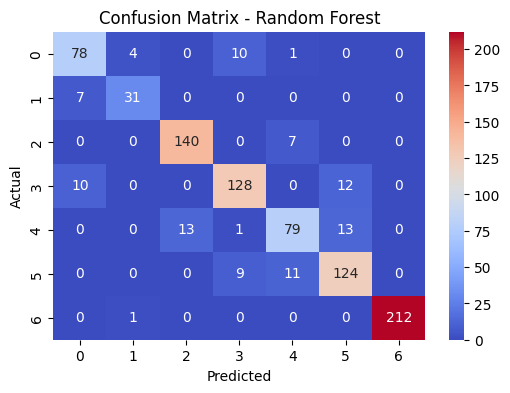
This code generates **confusion matrix heatmaps** for different model predictions.

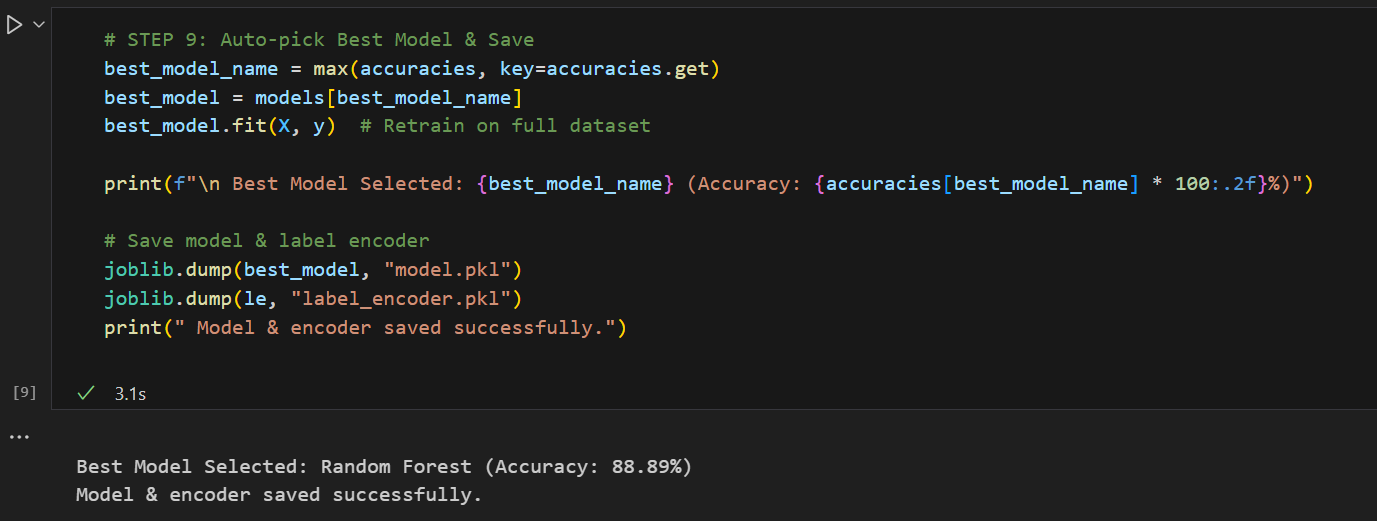
**Step-by-step:**

1. **Loop through predictions** → Each model’s predictions (y\_pred) are taken from the predictions dictionary (name is the model name).
2. **Create confusion matrix** → confusion\_matrix(y\_test, y\_pred) compares actual vs predicted values.
3. **Plot heatmap** → Using seaborn.heatmap() with annotations (annot=True) to show numbers inside the grid.
4. **Styling** →
   * figsize=(6,4) → set plot size.
   * cmap="coolwarm" → color scheme.
   * Labels (Actual, Predicted) and title (Confusion Matrix - model\_name) added.
5. **Display** → plt.show() shows the heatmap.







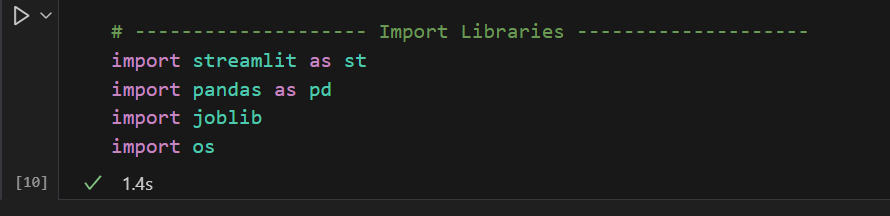


*Explaination:*

This code does the following in short:

1. **Auto-picks the best model** – From a dictionary of models and their accuracies, it selects the one with the highest accuracy.
2. best\_model\_name = max(accuracies, key=accuracies.get)
3. best\_model = models[best\_model\_name]
4. **Retrains it** – Fits the chosen best model again on the full dataset (X, y).
5. **Displays result** – Prints which model was selected and its accuracy.
6. **Saves the model & encoder** – Saves the trained best model (model.pkl) and label encoder (label\_encoder.pkl) using joblib.
7. Output in your case:

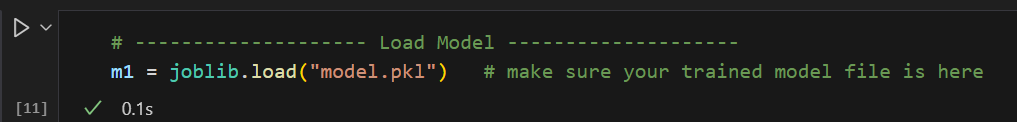
* Best model = **Random Forest** with **88.89% accuracy**
* Model & encoder saved successfully.



*Explaination:*

This code is simply **importing required libraries** for your project:

* streamlit as st → for creating a web app/dashboard.
* pandas as pd → for handling and analyzing data.
* joblib → for loading/saving trained ML models.
* os → for interacting with the operating system (like checking files, paths).

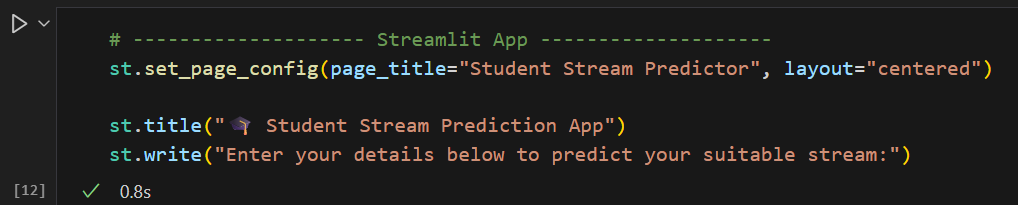


*Explaination:*

This code is **loading a pre-trained machine learning model** from a file:

m1 = joblib.load("model.pkl")

* joblib.load() → used to load saved Python objects (commonly ML models).
* "model.pkl" → the file where the trained model was stored.
* m1 → the loaded model object, ready to use for predictions.



*Explaination:*

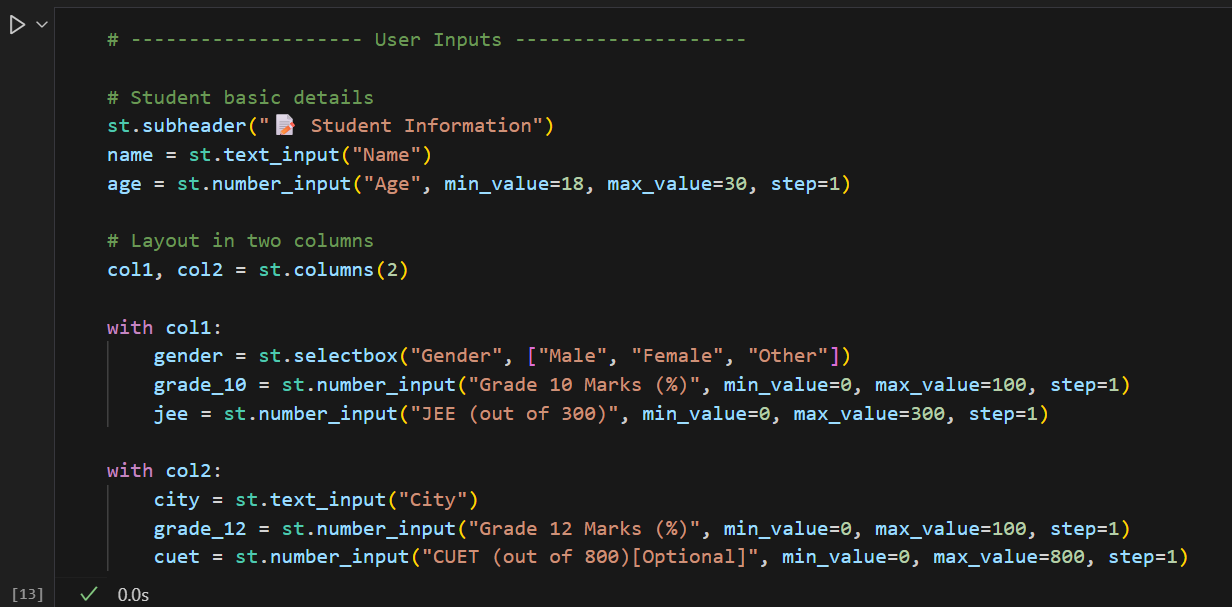
This code is setting up a **Streamlit web app interface**:

st.set\_page\_config(page\_title="Student Stream Predictor", layout="centered")

st.title("🎓 Student Stream Prediction App")

st.write("Enter your details below to predict your suitable stream:")

* st.set\_page\_config(...) → sets app settings (title shown on browser tab, layout).
* st.title(...) → displays the main title of the app.
* st.write(...) → adds descriptive text/instructions for users.



*Explaination:*

This code creates the **student input form** in Streamlit:

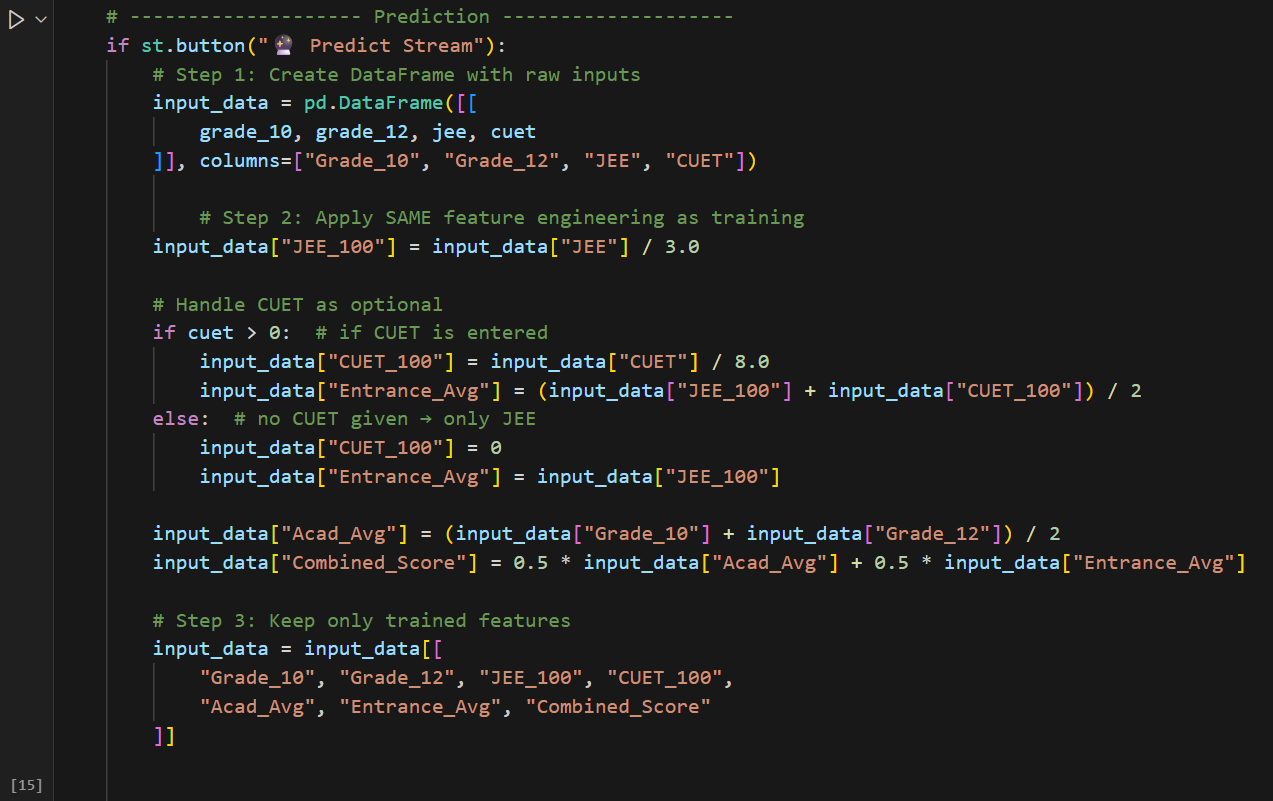
* st.subheader("📋 Student Information") → adds a section header.
* st.text\_input, st.number\_input, st.selectbox → collect inputs like **name, age, gender, city, marks, JEE, CUET scores**.
* st.columns(2) → arranges inputs neatly in **two side-by-side columns**.

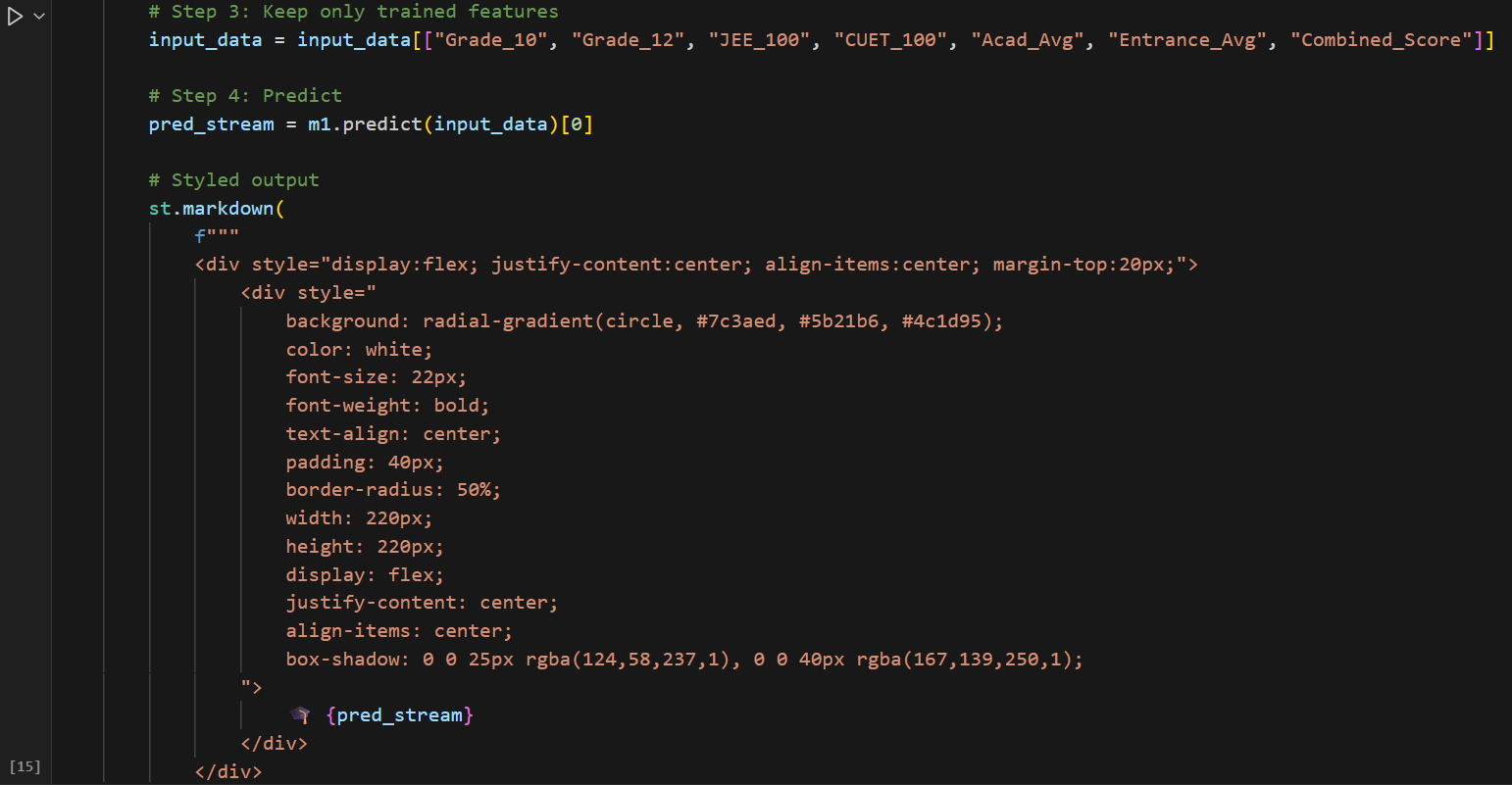


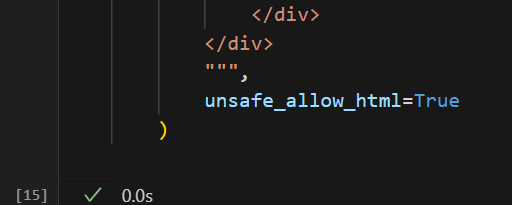
*Explaination:*

This code customizes the **button style** in Streamlit using CSS inside st.markdown:

* Sets **background color, text color, border, padding, rounded corners** for buttons.
* Adds a **hover effect** → when the mouse moves over the button, the color changes.
* unsafe\_allow\_html=True → allows embedding custom HTML/CSS in Streamlit.



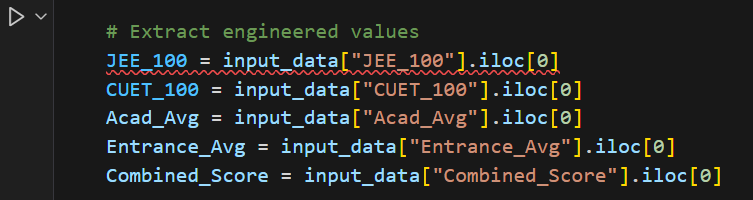




*Explaination:*

This part does two things:

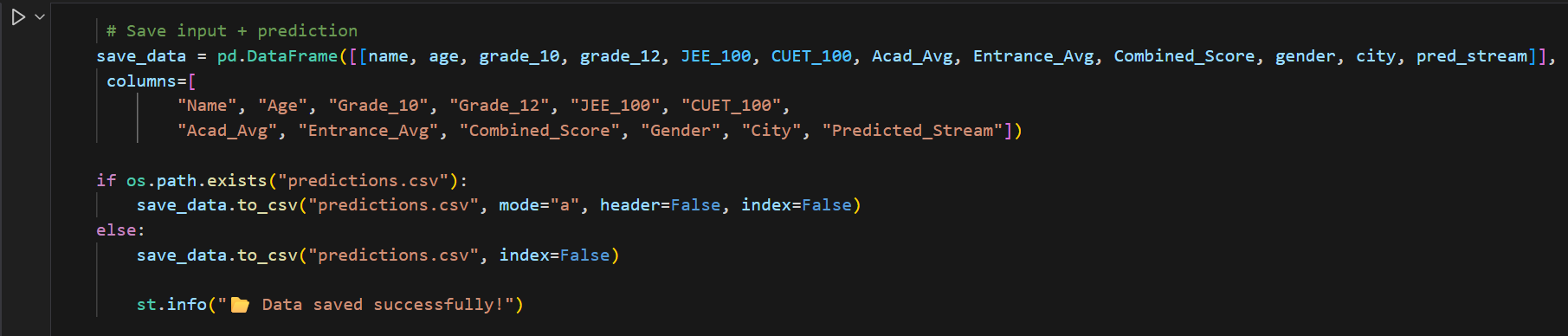
1. **Predict stream** → model (m1.predict) takes the processed input and outputs the student’s suitable stream.
2. **Styled display** → shows the predicted stream inside a colorful, rounded, gradient-styled box using custom HTML/CSS.



*Explaination:*

This code extracts the **calculated feature values** from the DataFrame (input\_data) into separate variables:

* JEE\_100, CUET\_100 → normalized exam scores.
* Acad\_Avg → average of Grade 10 & 12.
* Entrance\_Avg → average of JEE & CUET.
* Combined\_Score → overall score.

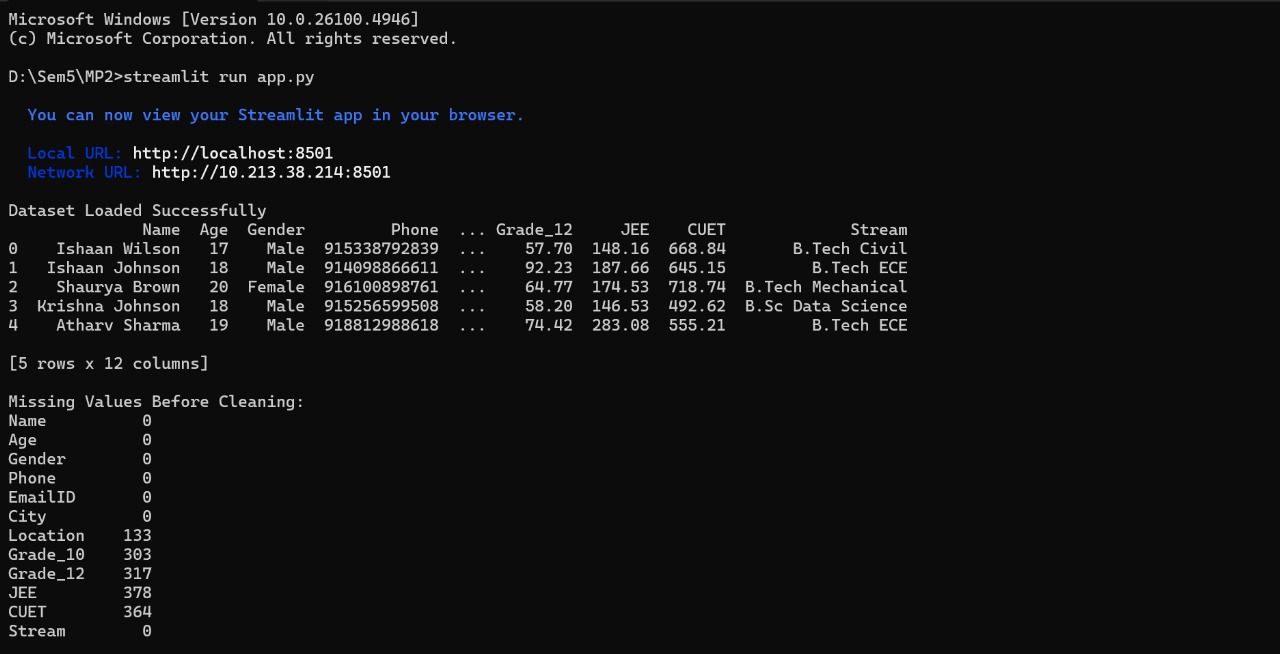


*Explaination:*

This code **saves the student’s input and prediction** into a CSV file:

* Creates a DataFrame (save\_data) with all details + predicted stream.
* If predictions.csv exists → appends new data.
* Else → creates a new CSV file.
* Shows message **“📁 Data saved successfully!”**.

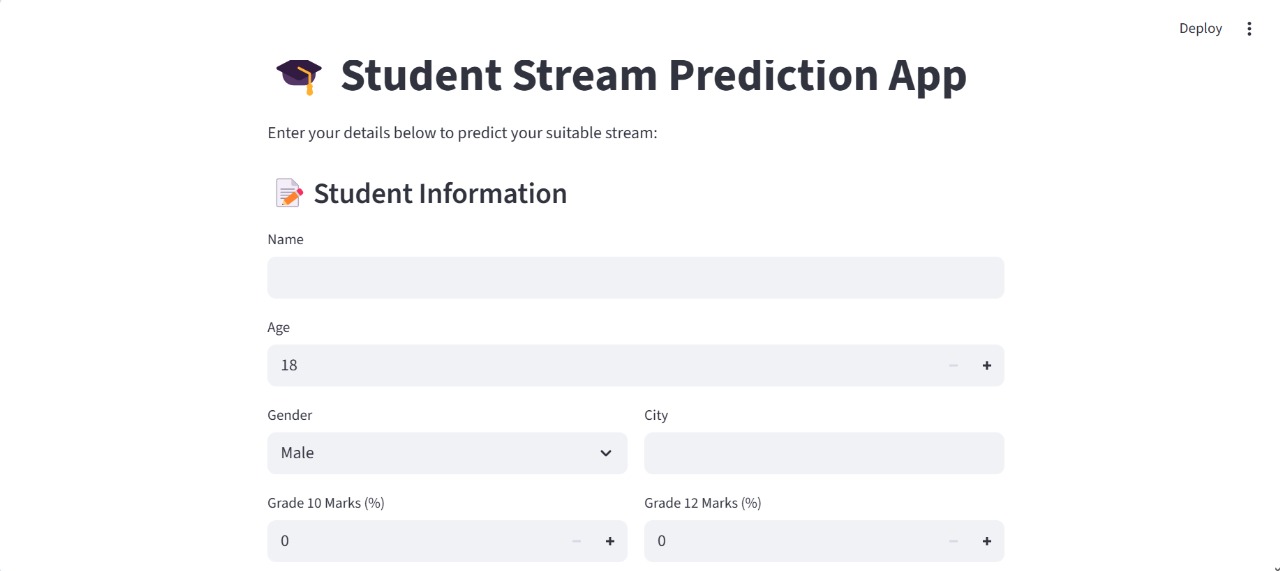
Output :

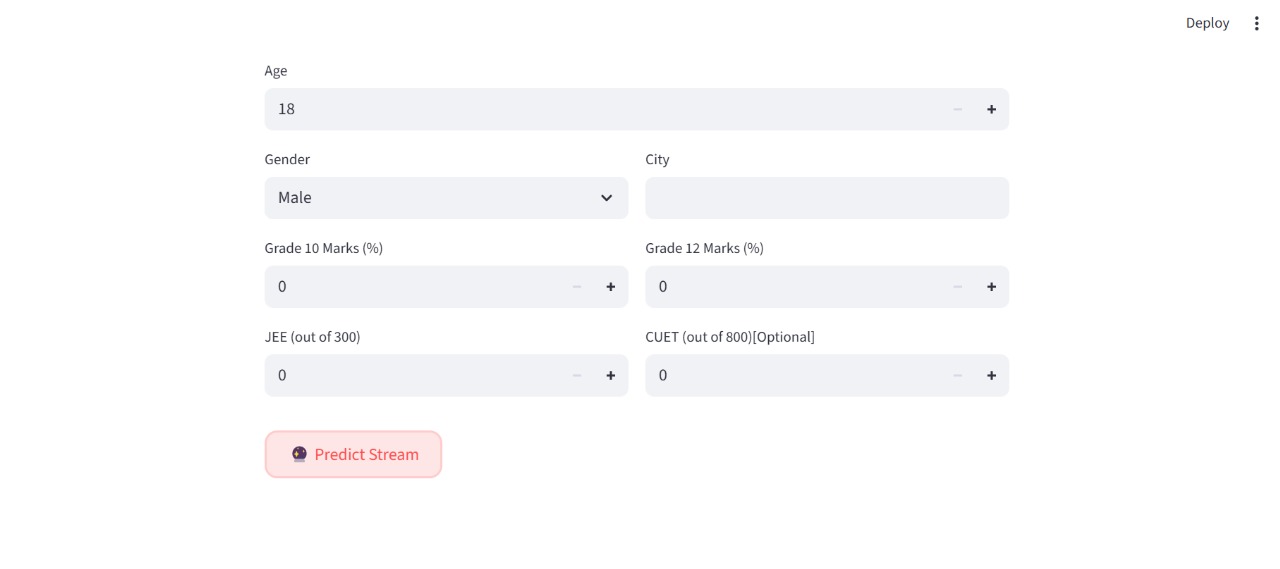


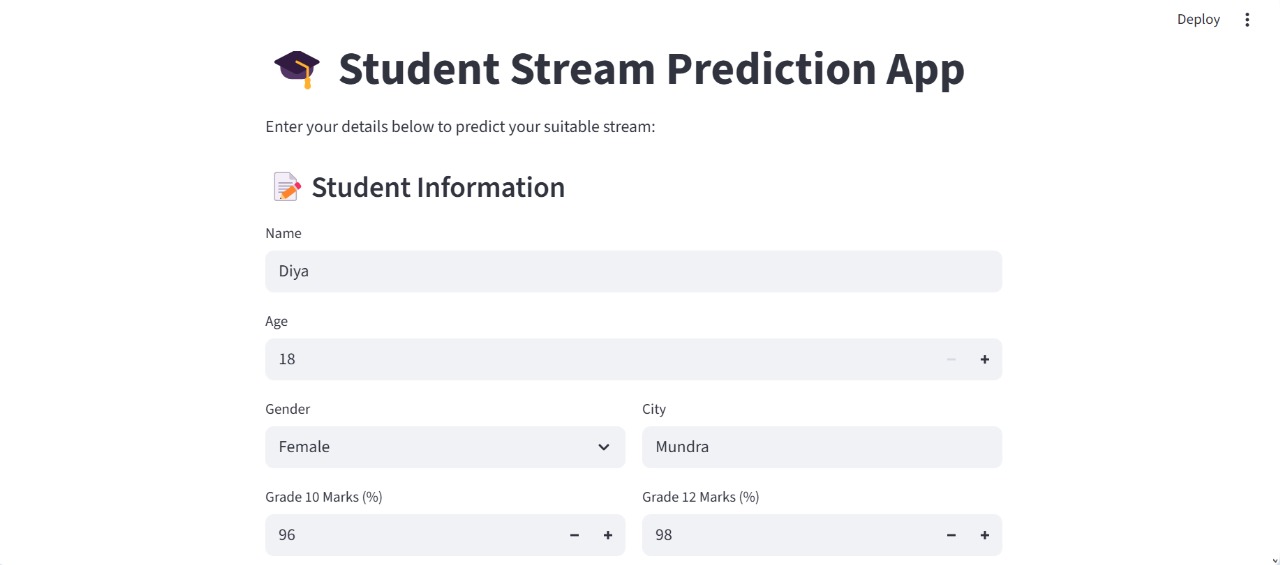
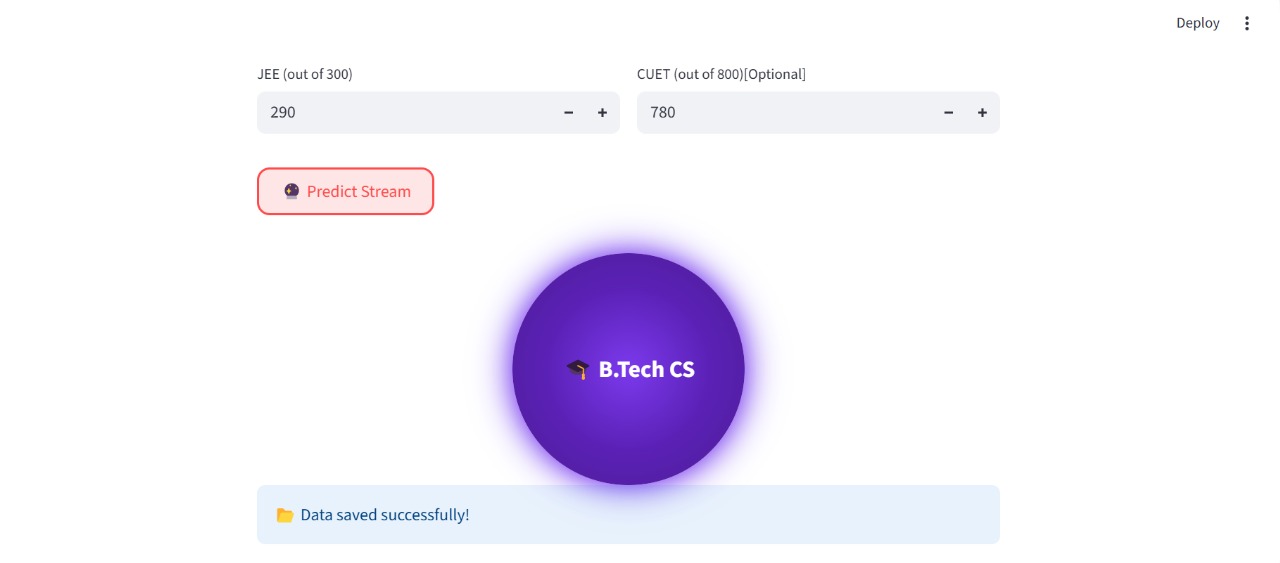
*Explaination:*

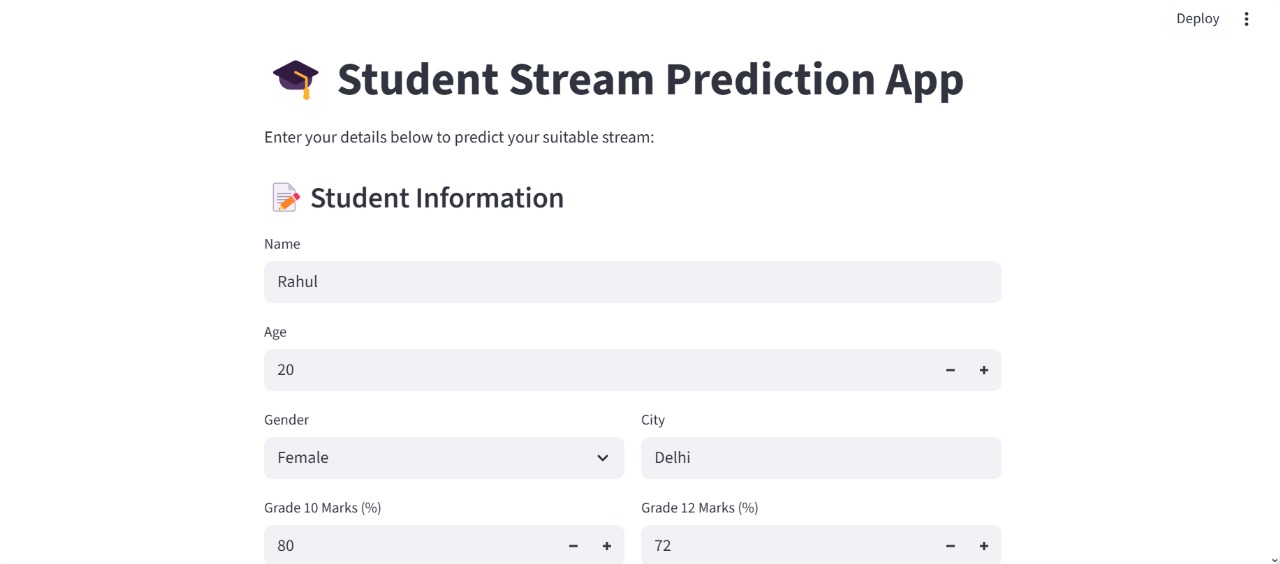
This screenshot shows a **Streamlit app running on localhost (port 8501)**.

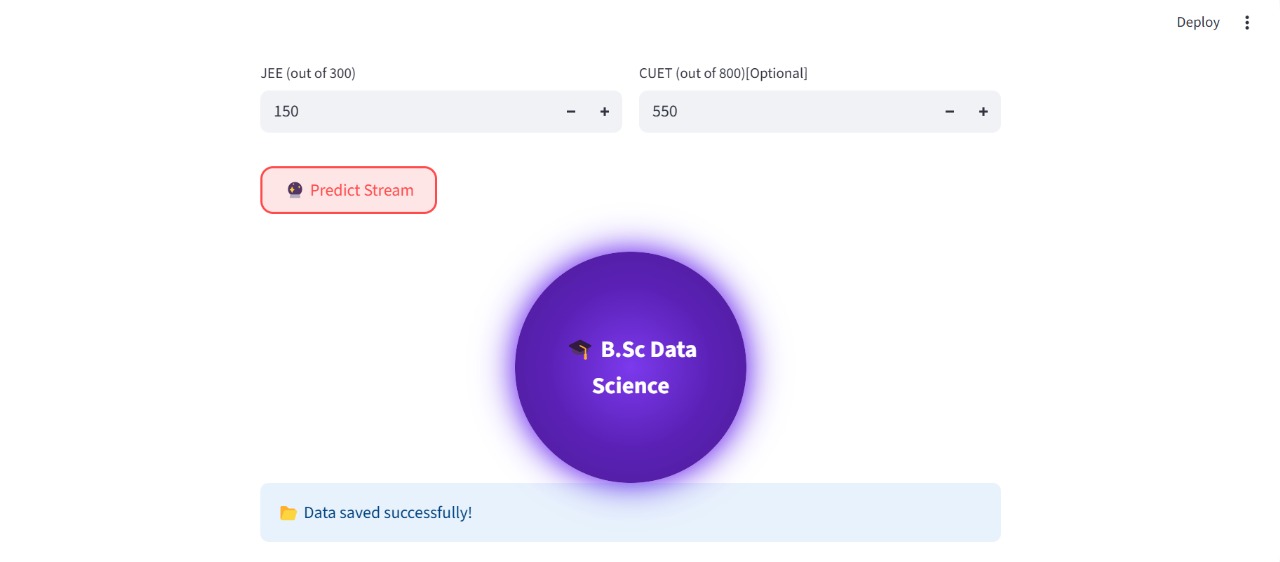
* The dataset is loaded successfully, displaying **student information** (Name, Age, Gender, Phone, Grade\_12, JEE, CUET, Stream, etc.).
* It shows the **first 5 rows** of the dataset.
* Then it prints the **count of missing values** in each column:
  + Columns like Name, Age, Gender, Phone, Stream have **0 missing values**.
  + Columns like Location (133), Grade\_10 (133), Grade\_12 (317), JEE (378), CUET (364) have **missing values**.

**USER INTERFACE OUTPUT:**

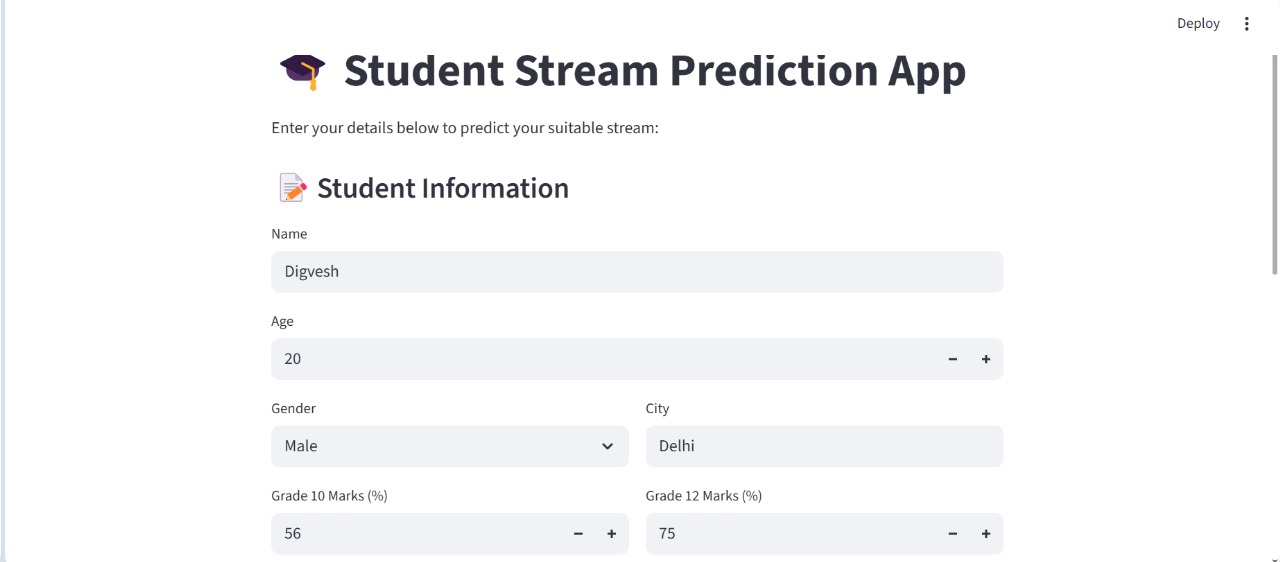


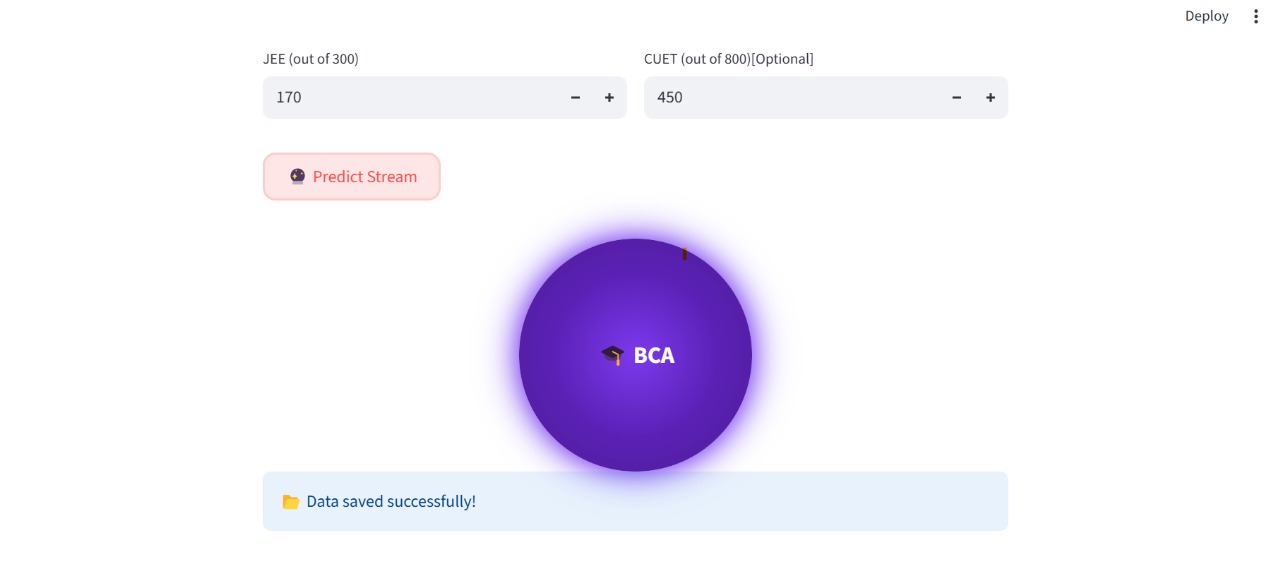
Sample 1:

Sample 2:

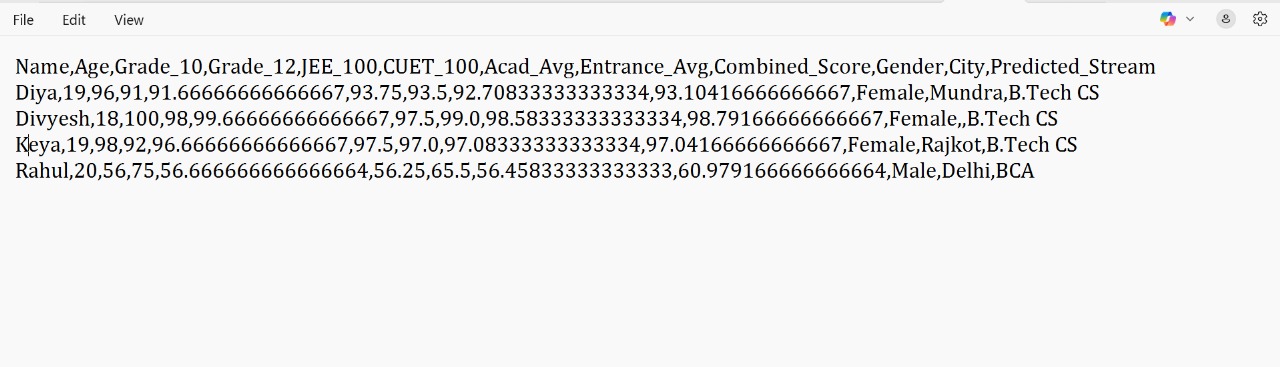


Sample 3:





**Dataset(Predictions Stored here):**



**Chapter 5: Proposed Enhancements**

Proposed Enhancements

Enhancements: larger dataset, deep learning, non-academic parameters, mobile deployment.

It suggests possible improvements for the system or project:

1. **Larger Dataset** – Using more data would improve accuracy, reliability, and generalization of the model.
2. **Deep Learning** – Applying advanced techniques (like neural networks) instead of only basic ML for better predictions.
3. **Non-Academic Parameters** – Including factors beyond academics (e.g., extracurriculars, personality traits, socio-economic background) for more holistic analysis.
4. **Mobile Deployment** – Making the system/app available on mobile platforms for easy accessibility and usability.

**Chapter 6: Conclusion**

Conclusion:

The project successfully demonstrates the use of **Machine Learning (ML) techniques** for predicting academic streams based on student performance data.

Various models were tested, and the **Random Forest classifier** proved to deliver the highest accuracy, highlighting its ability to handle complex datasets and reduce overfitting.

The system is further enhanced with a **Streamlit-based application**, ensuring user-friendliness and accessibility for both students and academic counselors. This interactive interface allows real-time predictions and recommendations, making it practical for educational institutions.

The study shows that integrating ML into academic decision-making can significantly aid students in selecting the most suitable stream, thereby reducing confusion and improving career guidance.

While the current system primarily relies on academic data, future improvements such as incorporating non-academic parameters, larger datasets, and deep learning methods could further enhance prediction accuracy and reliability.

In conclusion, the project not only demonstrates the feasibility of ML in education but also opens pathways for **smart, data-driven career counseling solutions** that are scalable, accessible, and impactful.

**Chapter 7: Bibliography**

Bibliography:

1.Scikit-learn Documentation – <https://scikit-learn.org>

2. Pandas Documentation – <https://pandas.pydata.org>

3. Streamlit Documentation – <https://docs.streamlit.io>

4. Seaborn Documentation – <https://seaborn.pydata.org>

5. Research papers on Student performance prediction.

6. Stack Overflow, GeeksforGeeks, W3Schools for code

references.

**Chapter 8 References**

### **8.1 Offline References**

[1] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2019.  
[2] W. McKinney, Python for Data Analysis: Data Wrangling with Pandas, NumPy, and Jupyter, 2nd ed. Sebastopol, CA, USA: O’Reilly Media, 2017.  
[3] S. Raschka and V. Mirjalili, Python Machine Learning, 3rd ed. Birmingham, UK: Packt Publishing, 2019.

### **8.2 Online References**

[4] Scikit-learn Developers, “Scikit-learn Documentation,” [Online]. Available: [https://scikit-learn.org](https://scikit-learn.org?utm_source=chatgpt.com). [Accessed: July. 29,2025].  
[5] pandas Development Team, “pandas Documentation,” [Online]. Available: [https://pandas.pydata.org](https://pandas.pydata.org?utm_source=chatgpt.com). [Accessed: Aug. 3, 2025].  
[6] Streamlit Inc., “Streamlit Documentation,” [Online]. Available: [https://docs.streamlit.io](https://docs.streamlit.io?utm_source=chatgpt.com). [Accessed: Aug. 4, 2025].  
[7] Seaborn Developers, “Seaborn Documentation,” [Online]. Available: [https://seaborn.pydata.org](https://seaborn.pydata.org?utm_source=chatgpt.com). [Accessed: Aug. 5, 2025].  
[8] Matplotlib Developers, “Matplotlib Documentation,” [Online]. Available: https://matplotlib.org/stable/contents.html. [Accessed: Aug. 6, 2025].

[9] Stack Overflow, “Stack Overflow: Where Developers Learn, Share, & Build Careers,” [Online]. Available: [https://stackoverflow.com](https://stackoverflow.com?utm_source=chatgpt.com). [Accessed: Aug. 7, 2025].  
[10] GeeksforGeeks, “GeeksforGeeks: A Computer Science Portal for Geeks,” [Online]. Available: https://www.geeksforgeeks.org. [Accessed: Aug. 9, 2025].  
[11] W3Schools, “W3Schools Online Web Tutorials,” [Online]. Available: https://www.w3schools.com. [Accessed: Aug. 10, 2025].