ARIGNAR ANNA GOVERNMENT ARTS COLLEGE VILLUPURAM – 605 602.



DEPARTMENT OF COMPUTER SCIENCE

**MACHINE LEARNING WITH PYTHON**

**Project Title:** Intelligent admission : The future of University decision making with machine learning

**Team Id:** NM2023TMID16940

**Team Leader:** PRADEEP C (3BA685731E9438B967C48ACCB11107B0)

**Team member:** PRASANTH K (88015081CD6BE0799514DDDACFBF77D8)

**Team member:** PUSHPARAJ A (DC1E9CF6A561692859C4786CD96BF74B)

**Team member:** RAMAKRISHNAN V (51294FFBE1532E1A6CC4E9852C8B32B3)

Abstract

The intelligent admissions system is a revolutionary approach to the decision-making process for university admissions. This system utilizes machine learning algorithms to analyze various data points such as student demographics, academic achievements, extracurricular activities, and other relevant information. The system then uses this information to predict the probability of a student's success in a particular program or course. The system is designed to provide universities with a more accurate and efficient method of making admissions decisions. Additionally, the system offers students the opportunity to receive personalized recommendations based on their unique profile. This project aims to create an intelligent admissions system that can improve the quality of decision-making and enhance the overall student experience

**Overview:**

# INTRODUCTION

The intelligent admissions system is a project aimed at revolutionizing the university admissions process by utilizing machine learning algorithms to make better and more accurate admissions decisions. The system analyzes various data points such as student demographics, academic achievements, extracurricular activities, and other relevant information to predict the probability of a student's success in a particular program or course.

The intelligent admissions system also offers personalized recommendations to students based on their unique profile, providing them with valuable insights into their strengths and weaknesses and helping them make informed decisions about their academic future.

The ultimate goal of this project is to create a more efficient and effective admissions process that benefits both universities and students. By leveraging the power of machine learning, the intelligent admissions system has the potential to transform the way universities make admissions decisions and provide students with a more personalized and rewarding academic experience.

**Purpose:**

The purpose of the intelligent admissions system is to improve the university admissions process by utilizing machine learning algorithms to make better and more accurate admissions decisions.

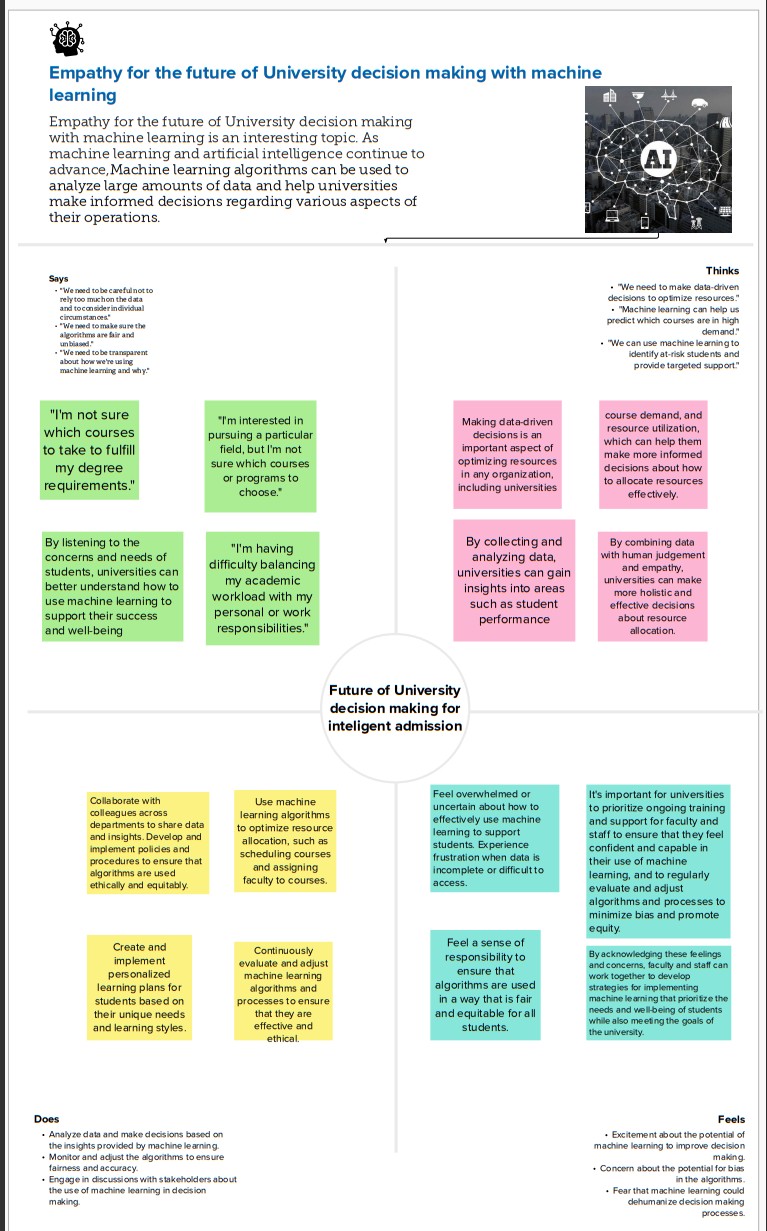
Traditionally, university admissions decisions are based on limited information such as academic transcripts, standardized test scores, and personal essays. However, this information does not always provide a comprehensive picture of a student's potential for success in a particular program or course.

The intelligent admissions system aims to fill this gap by analyzing a wider range of data points and utilizing machine learning algorithms to predict the probability of a student's success. By providing universities with a more accurate and efficient method of making admissions decisions, the system can help universities identify and admit the most qualified and capable students.

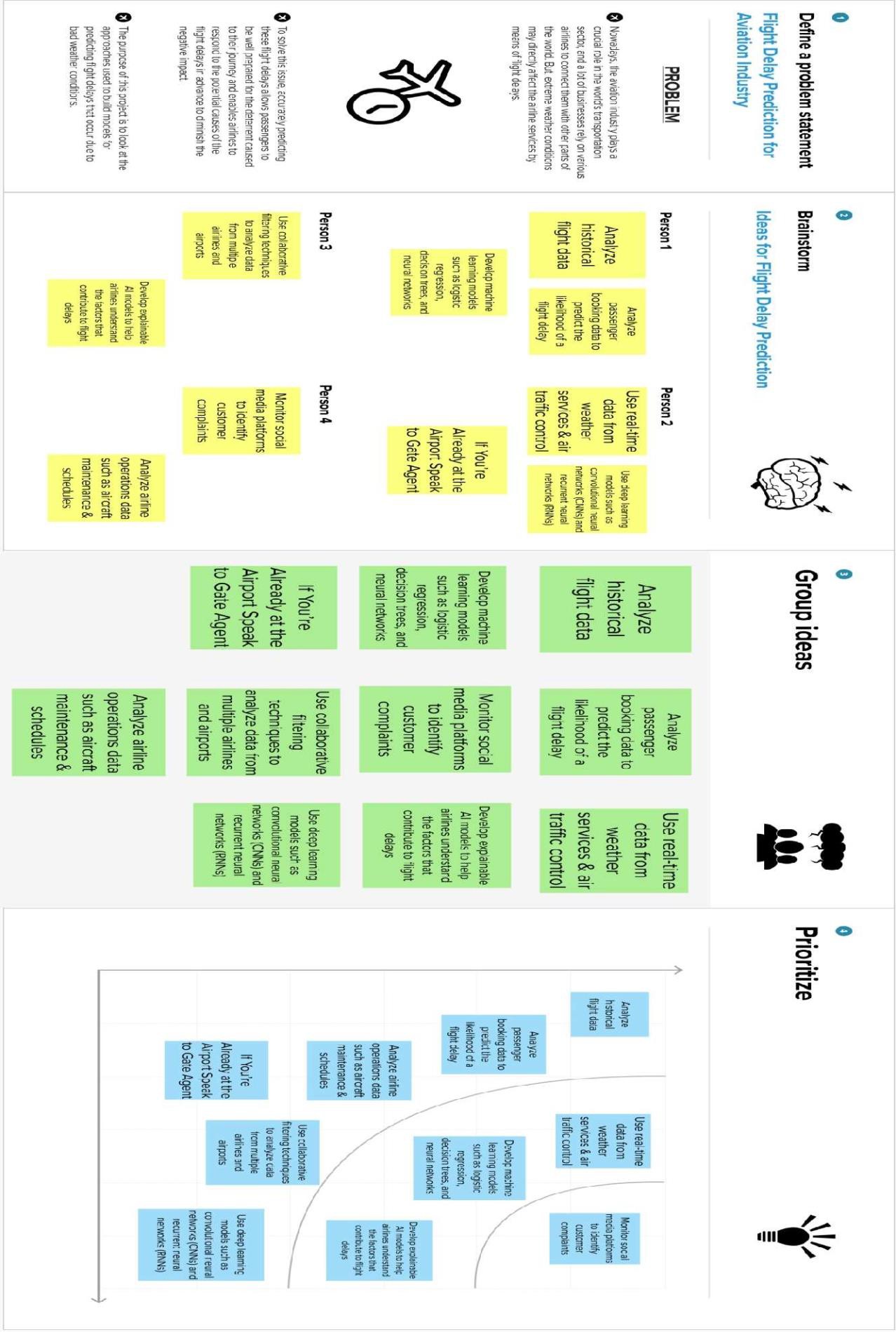
Additionally, the intelligent admissions system offers students personalized recommendations based on their unique profile, which can help them make informed decisions about their academic future. By leveraging the power of machine learning, the intelligent admissions system has the potential to transform the way universities make admissions decisions and provide students with a more personalized and rewarding academic experience.

## PROBLEM DEFINITION AND DESIGN THINKING

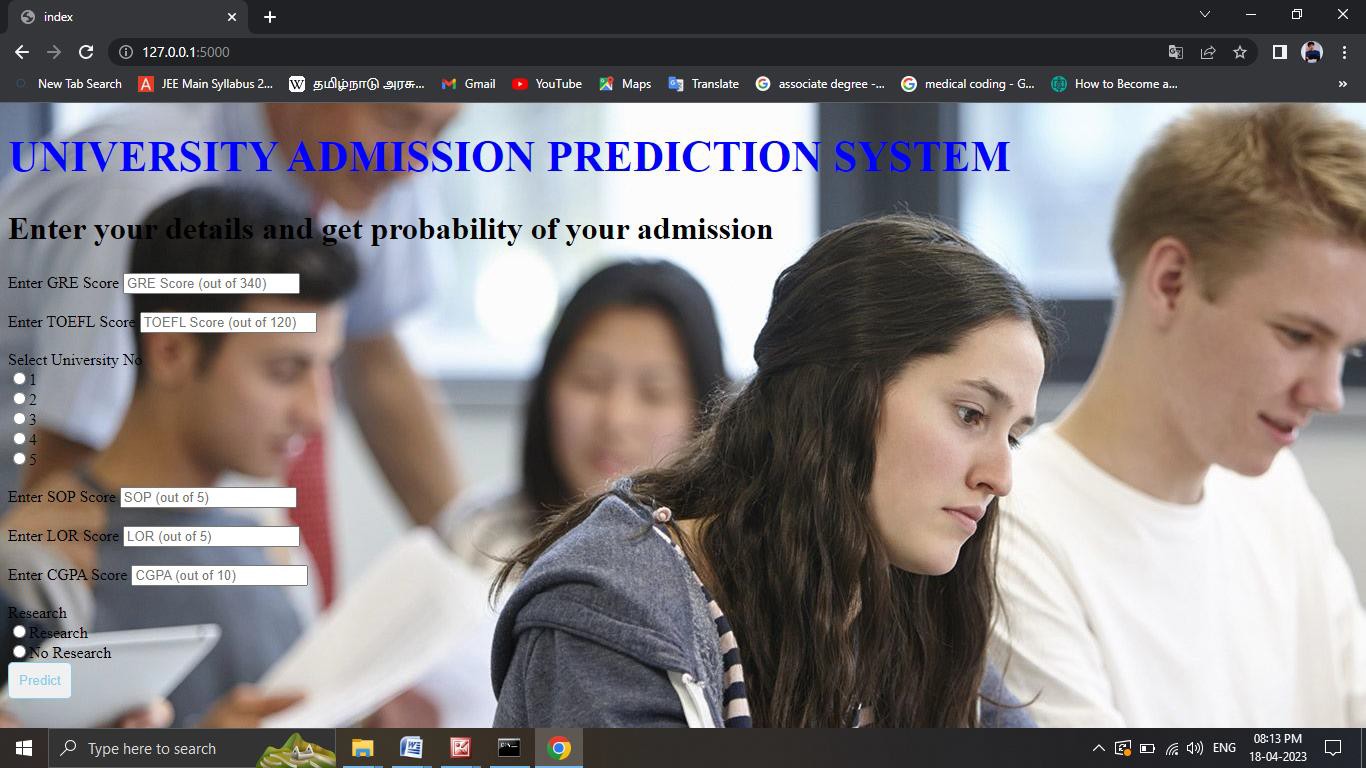
**Empathy map:**

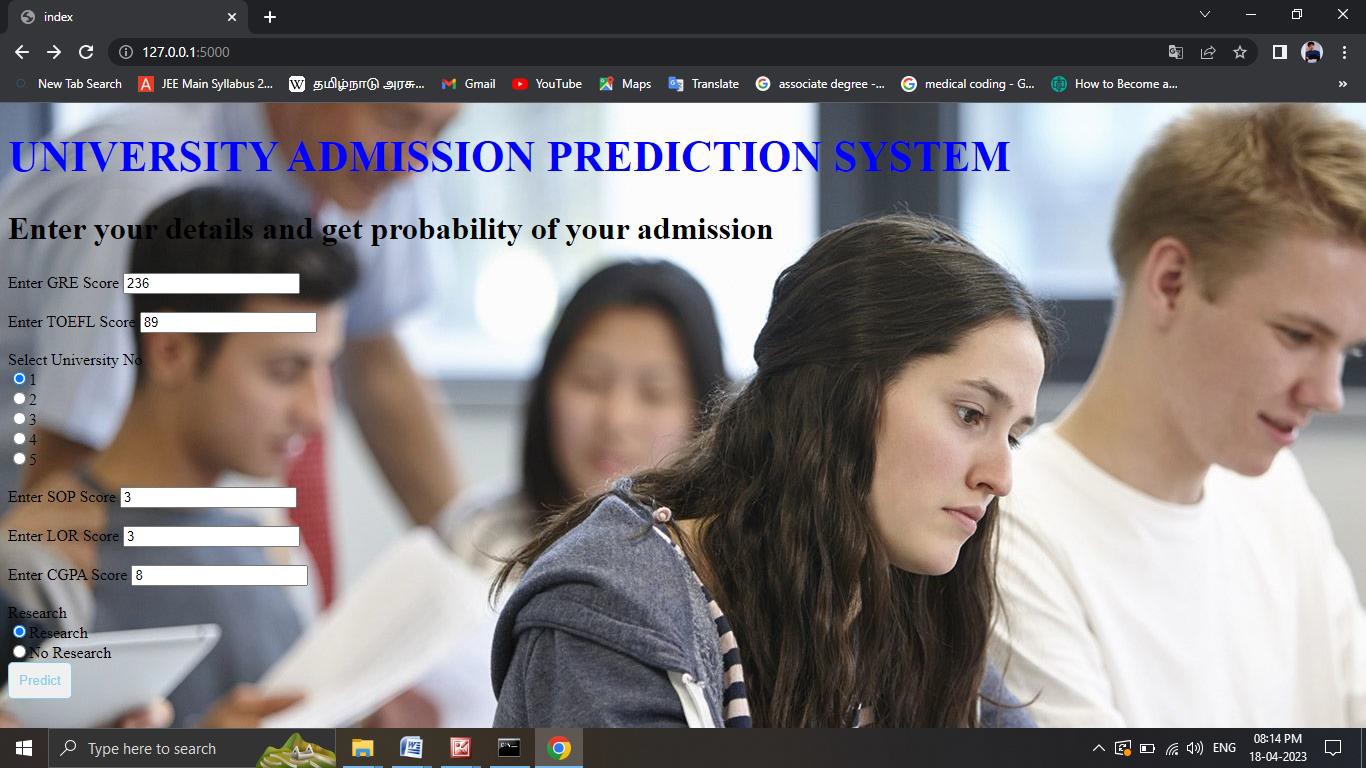


**Ideation & Brainstorming Map:**

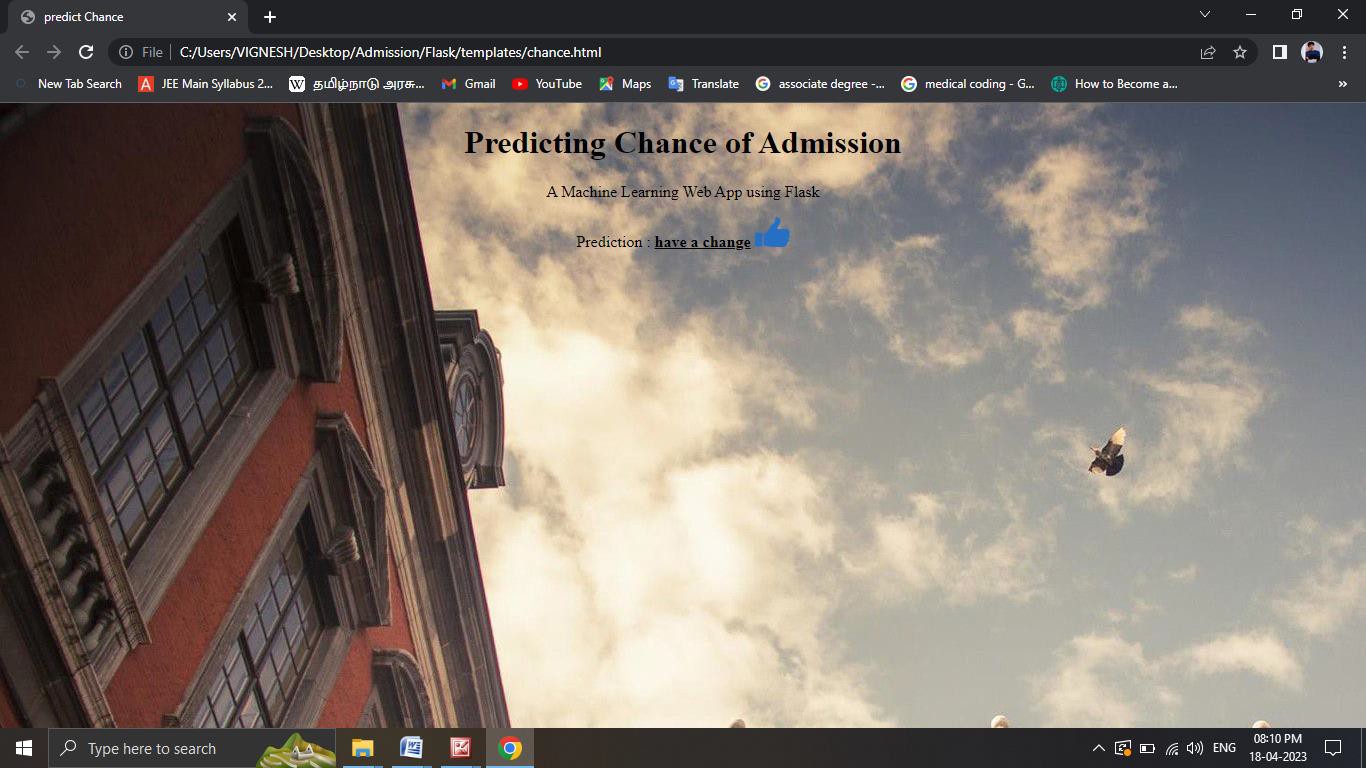


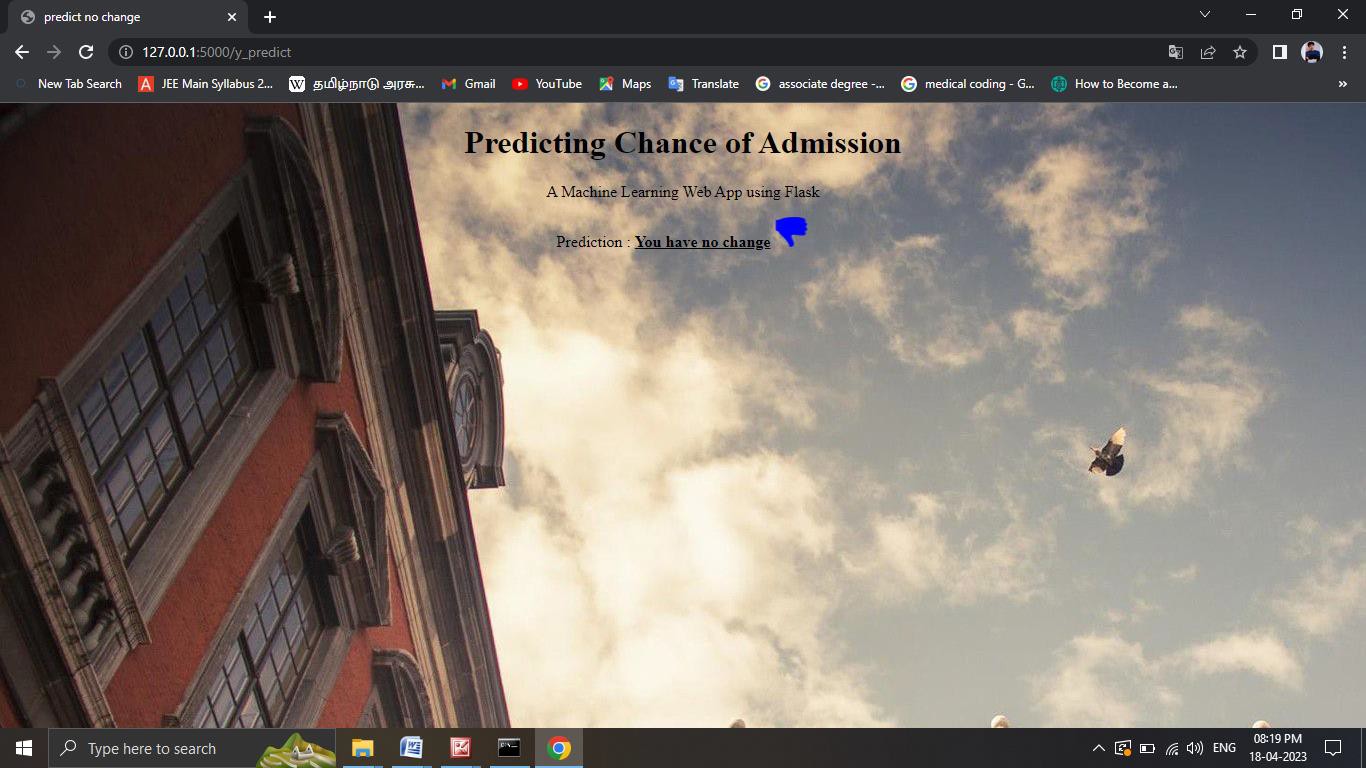
# RESULT





v





# ADVANTAGES AND DISADVANTAGES

**Advantages:**

* **Increased efficiency:** Intelligent admission systems can process large amounts of data quickly and accurately, making the admissions process more efficient. This can save time and reduce the workload for admissions officers, who can focus on other important tasks.
* **Improved accuracy:** Machine learning algorithms can analyze vast amounts of data, including academic records, test scores, extracurricular activities, and other relevant factors, to identify patterns and make more accurate predictions about a student's likelihood of success at a particular university.
* **Fairness and transparency:** Intelligent admission systems can help reduce bias and ensure fairness in the admissions process by relying on objective data and algorithms rather than subjective judgments. This can help level the playing field for students from diverse backgrounds and reduce the impact of implicit bias on admissions decisions.
* **Better student matching:** By using machine learning algorithms to analyze data on student performance and preferences, universities can better match students with programs and courses that suit their strengths and interests. This can lead to higher retention rates and better outcomes for students.
* **Cost savings:** Intelligent admission systems can reduce the need for expensive and time-consuming manual processes, such as reviewing thousands of applications by hand, and can help universities save money on staffing and other resources.

**Disadvantages:**

* **Bias in the data:** Intelligent admission systems rely on historical data to make predictions about future student success, but this data may be biased due to factors such as socioeconomic status, race, gender, or other demographic variables. If the data used to train the machine learning algorithms is biased, this bias may be perpetuated in the admissions decisions, leading to unfair outcomes.
* **Lack of transparency:** Machine learning algorithms can be complex and difficult to understand, which can make it hard for stakeholders to understand how admissions decisions are being made. This lack of transparency can make it difficult for students to challenge decisions or for universities to ensure that the process is fair and equitable.
* **Inability to capture important non-academic factors:** While machine learning algorithms can analyze large amounts of academic data, they may not be able to capture important non-academic factors that are also important for success, such as grit, creativity, or social skills. Focusing too narrowly on academic metrics may miss out on students who have potential but do not fit traditional academic profiles.
* **Dependence on technology:** Intelligent admission systems are dependent on technology, which can be vulnerable to errors or hacking. If the system is not properly designed or maintained, it could lead to errors or biases in admissions decisions.
* **Reduction in human involvement:** While intelligent admission systems can streamline the admissions process, they may also lead to a reduction in human involvement. This could result in a lack of personal interaction and communication with applicants, leading to a less personalized and supportive experience for students.

1. **APPLICATIONS**

* **Predictive analytics:** Intelligent admission systems can use predictive analytics to identify patterns in student data and make predictions about future success. By analyzing academic records, test scores, extracurricular activities, and other relevant factors, these systems can identify students who are likely to succeed in particular programs or courses.
* **Personalized recommendations:** Intelligent admission systems can use machine learning algorithms to analyze student data and make personalized recommendations for courses, programs, and extracurricular activities that are best suited to the student's strengths and interests. This can help ensure that students are matched with programs and courses that are a good fit for their individual needs.
* **Streamlined application process:** Intelligent admission systems can automate many of the manual processes involved in the admissions process, such as data entry and document processing, leading to a more streamlined and efficient process for both students and admissions officers.
* **Improved diversity and inclusion:** Intelligent admission systems can help reduce bias in the admissions process by relying on objective data and algorithms rather than subjective judgments. This can help ensure that students from diverse backgrounds have an equal opportunity to be admitted to university programs.
* **Enhanced student support:** Intelligent admission systems can use data analytics to identify students who may be at risk of dropping out or falling behind and provide personalized support to help them succeed. This can include targeted interventions such as academic coaching, tutoring, or mentorship programs

# CONCLUSION

* In conclusion, intelligent admission has the potential to revolutionize the university decision-making process by leveraging machine learning algorithms and data analytics to improve efficiency, fairness, and accuracy.
* By using predictive analytics to identify patterns in student data and making personalized recommendations for courses, programs, and extracurricular activities
* intelligent admission systems can help ensure that students are matched with programs and courses that are a good fit for their individual needs.
* Additionally, these systems can automate many of the manual processes involved in the admissions process, leading to a more streamlined and efficient process for both students and admissions officers.
* However, it is important to carefully consider both the advantages and disadvantages of intelligent admission, including potential biases in the data, lack of transparency, and dependence on technology.
* Overall, by implementing these systems ethically and with transparency, universities can harness the power of machine learning and data analytics to create a more efficient, equitable, and effective university decision-making process.

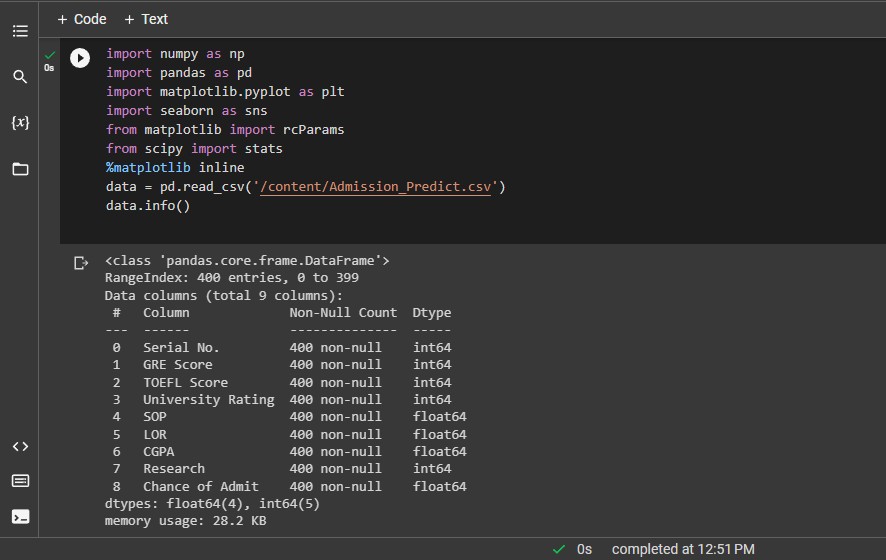
# FUTURE SCOPE

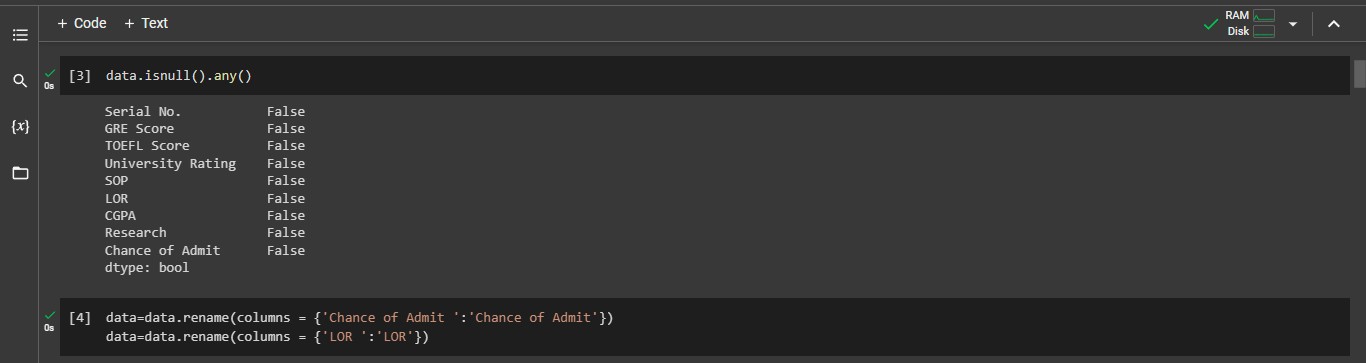
* The future scope of intelligent admission is vast and promising, as machine learning and data analytics continue to advance and become more sophisticated. Here are a few potential areas where intelligent admission could be further developed and expanded in the future
* Multi-dimensional data analysis: While intelligent admission systems currently rely on academic data to make predictions about student success, future systems may incorporate additional data sources such as social media activity, personality assessments, and other non- academic factors. This could provide a more comprehensive understanding of each student's unique strengths, interests, and potential.
* Natural language processing: Natural language processing (NLP) is a subfield of machine learning that focuses on analyzing and understanding human language. In the future, NLP could be used to analyze application essays, personal statements, and other written materials to provide additional insights into each student's potential.
* Improved fairness and equity: While intelligent admission has the potential to reduce bias in the admissions process, future systems could be further developed to ensure fairness and equity. For example, algorithms could be designed to explicitly account for socioeconomic or other demographic factors that may affect a student's academic performance.

# APPENDIX

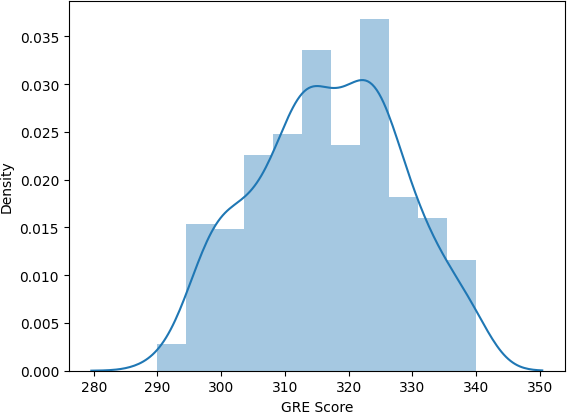
## Source Code:

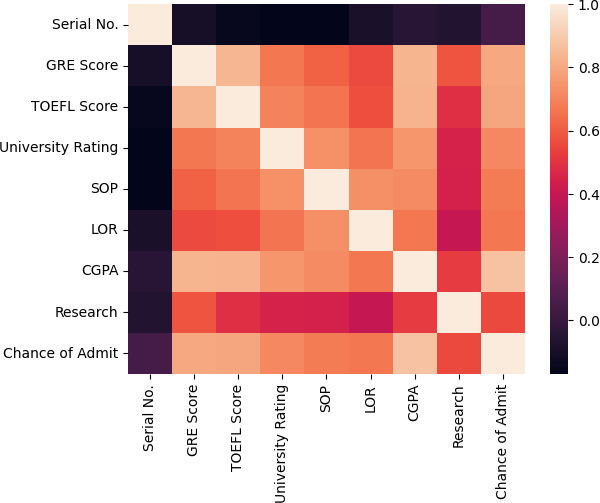
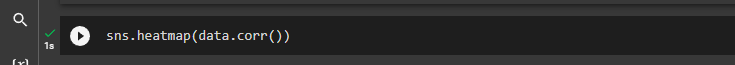
**Milestone 2:**





**Milestone 3:**

C:\Users\VIGNESH\Pictures\Screenshots\Screenshot (68).png

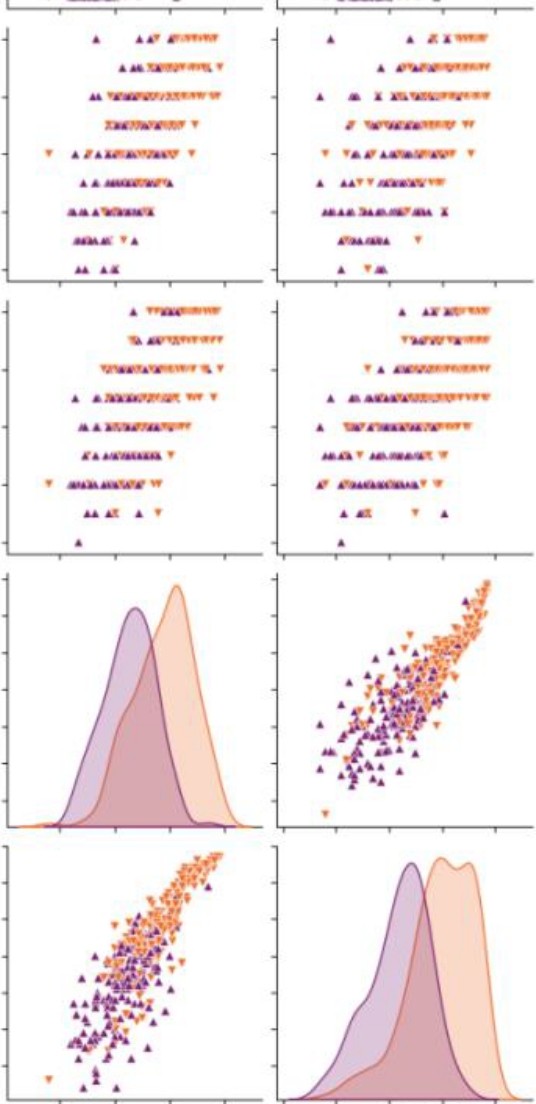
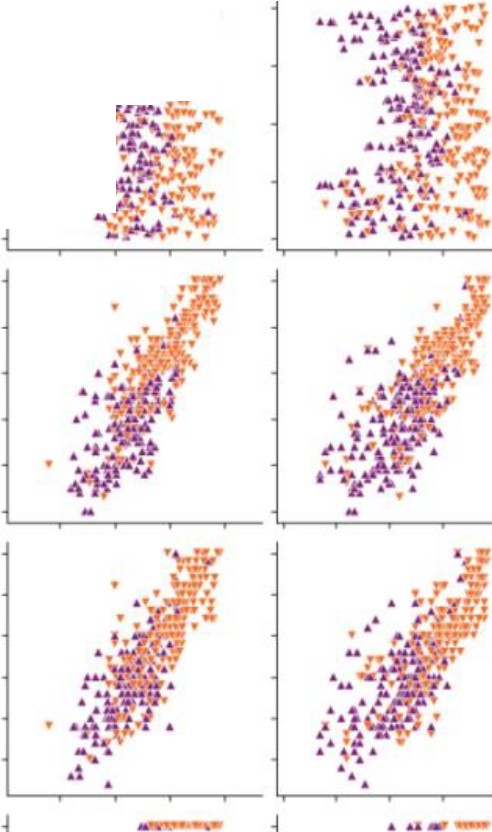
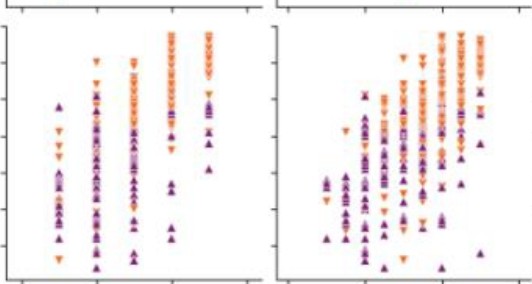
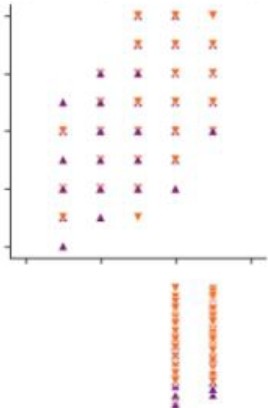
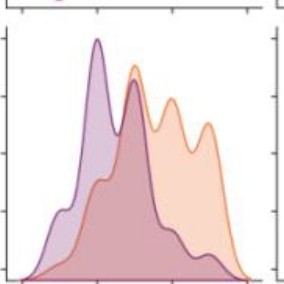
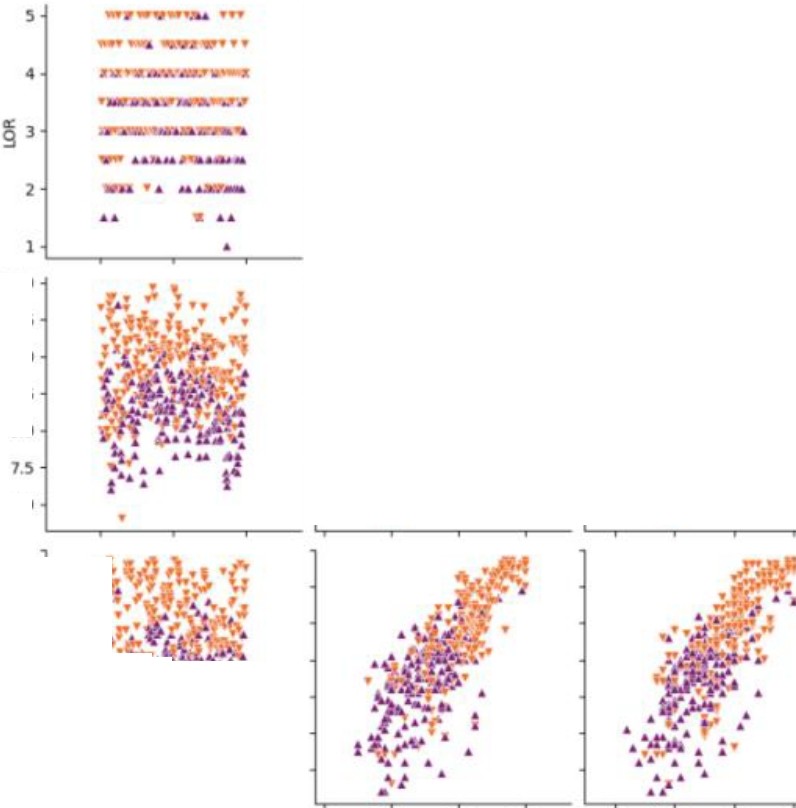
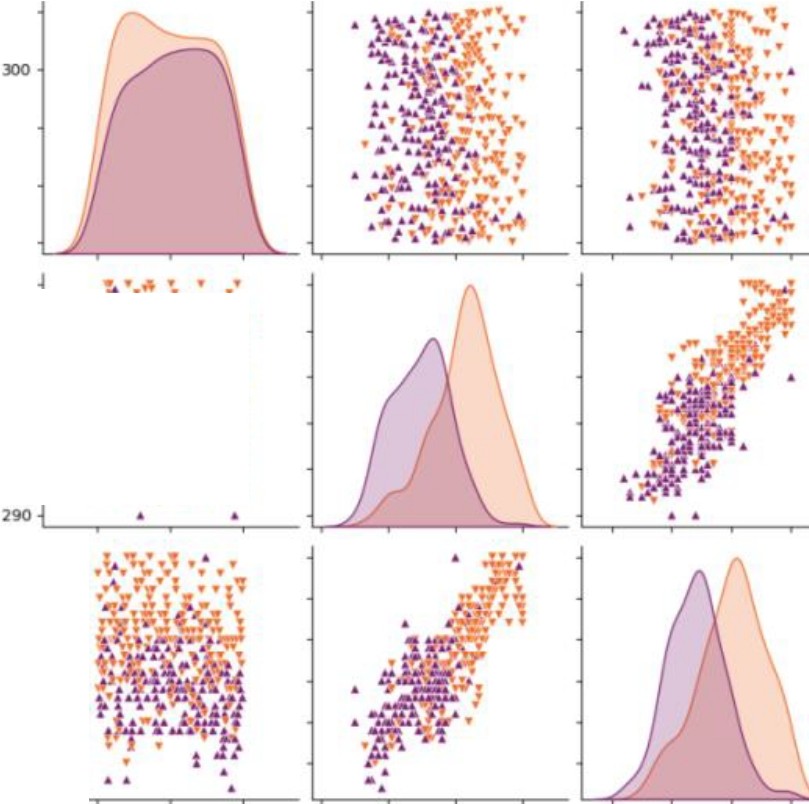


**0** sns.pairplot(data=data,hue='F

405

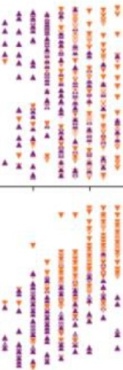
, 1· 11',markers=[' ·, '],palette='i Lr -- >I

**<00**



0

i *·.-:1·;•*

I • --:·n·:,.-- t..•.:.

I

j

**z**

j 200

J

100

340

330

::;: ..'-.'. : ♦: ·

l I

!

-·

·Pl!

* I ;

•• ! **1**I I

; .320

,)!

Jl0

"

.I' I ..

· I I I . :

: I ••

I

**,00**

120

lL5

i 110

* *t*

.'fIi 11.1. I

:11 •

11

..I,;: 1I1P.

I •1

*,c·::!}!,*:''A:.}::*•:*·*:f*r*i.*

. : IIt.I I .I.!• •

,)!

*;t* 105

,- 100

**95**



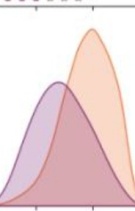
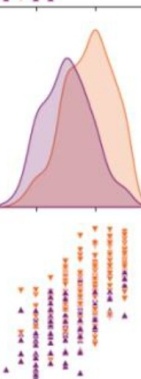


1.0..,0

...

1.5

Aese.,c.h

* 0
* l

I

! I

**1.0**

7.0

i i •

LO

••

0.1

0,7

0.6

5 •0..5.

**\l**

f *ii*

*-,·* ,..,..

..,.

* S• • (

·t '

280 JOO 320 340

GRE score

90 100 110 uo

TOEFLSCore

* + ' 0

University Rating

SOP ' 0

IDR

COPA

10 0.2 0.4 06 0.1 I 0

Cl\ance of Admit

0 sns.scatterplot(x='Un:.\·er·sity Ratirg',y='CGP;:,.',data=data,color="Rec', 5=100)

□

1

&

10.0

9.5

9.0

8.5

8

8.0

7.5

7.0

1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0

•

!

•

•

I

•I

I

•

•

I

I

I

•I

I

I

I

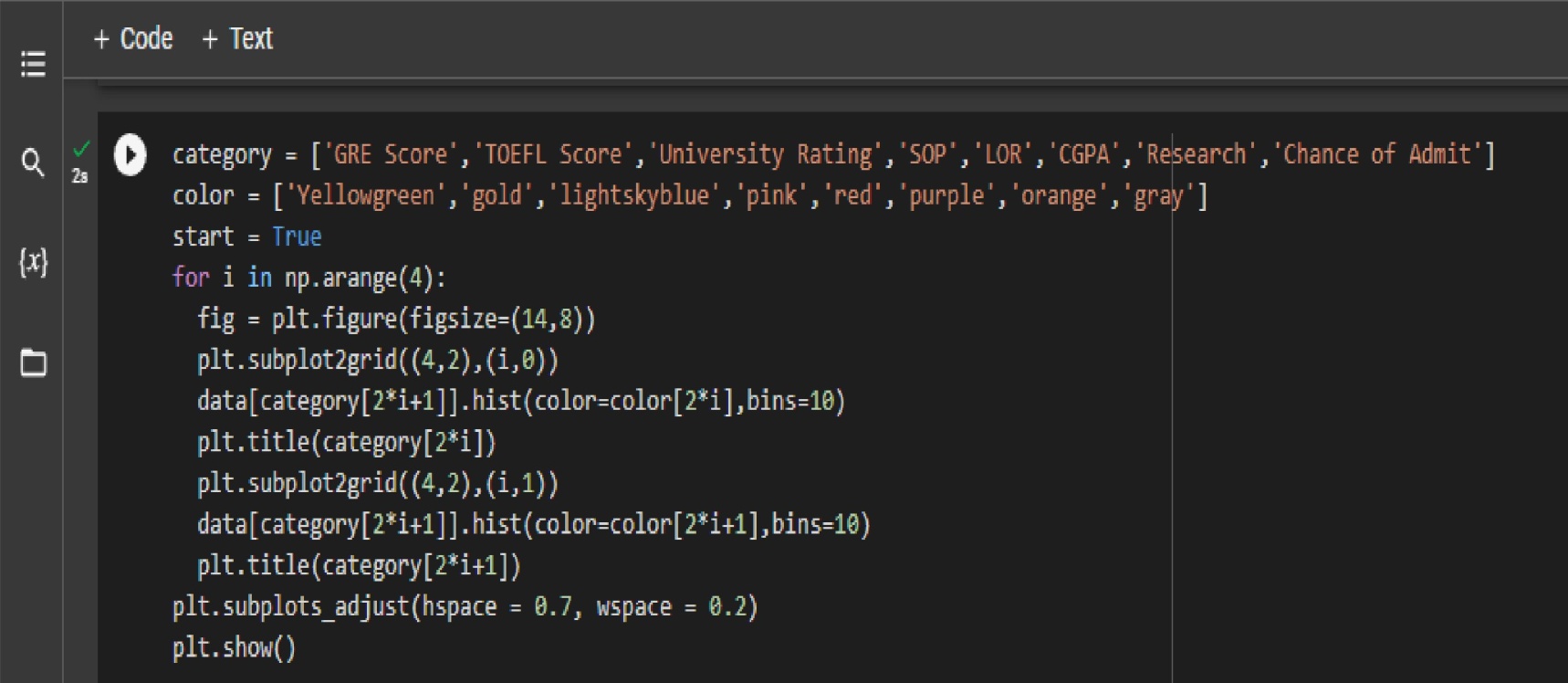
•

i i

•

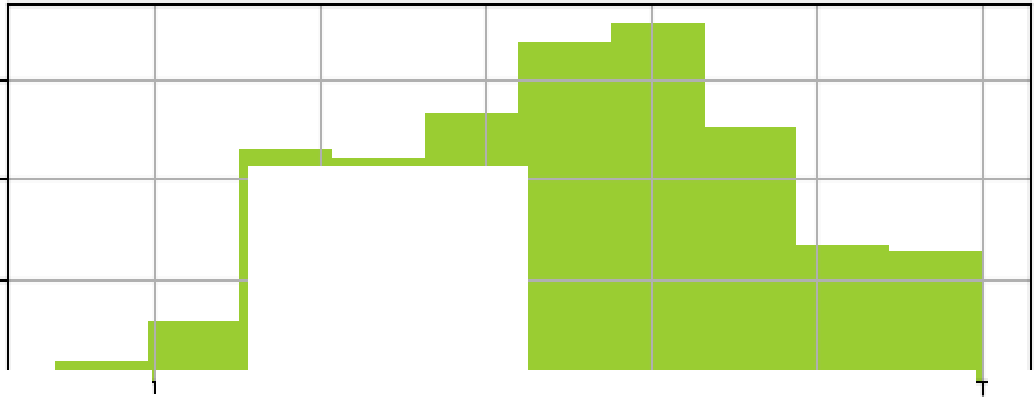
•

University Rating



GRE Score TOEFLScore

60-t--+-----+--------+---I



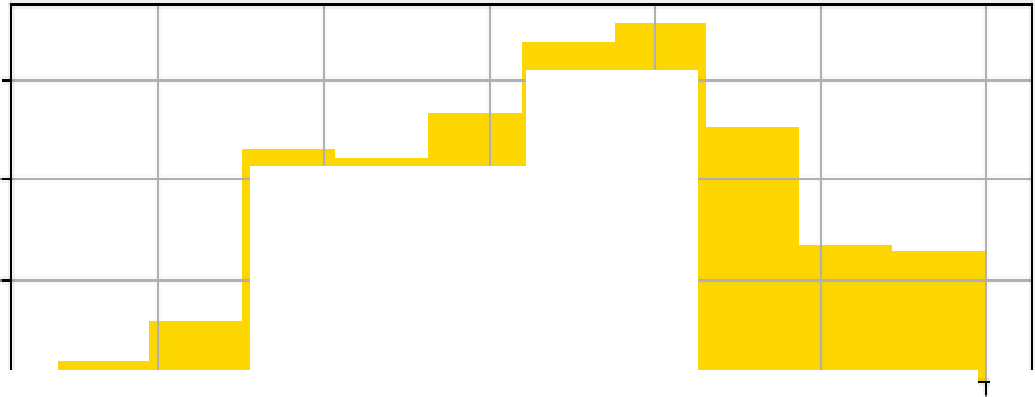
40 +-----+---I

20-1----+ <

o-1

60+---+-----+--------+---I

40+------+ I



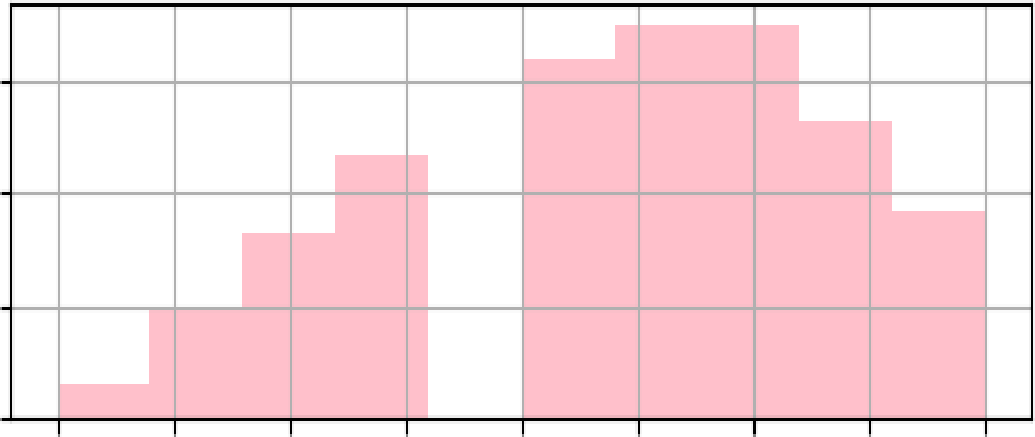
20+-----+ <

+ ' o-1 +

95 100 105 110 115 120 95 100 105 110 115 120

UniversityRating **SOP**

**60 60**



**40 40**

**20 20**

0

1.0 1.5

**2.0**

**2.5 3.0 3.5 4.0**

4.5 **5.0**

0

1.0 1.5

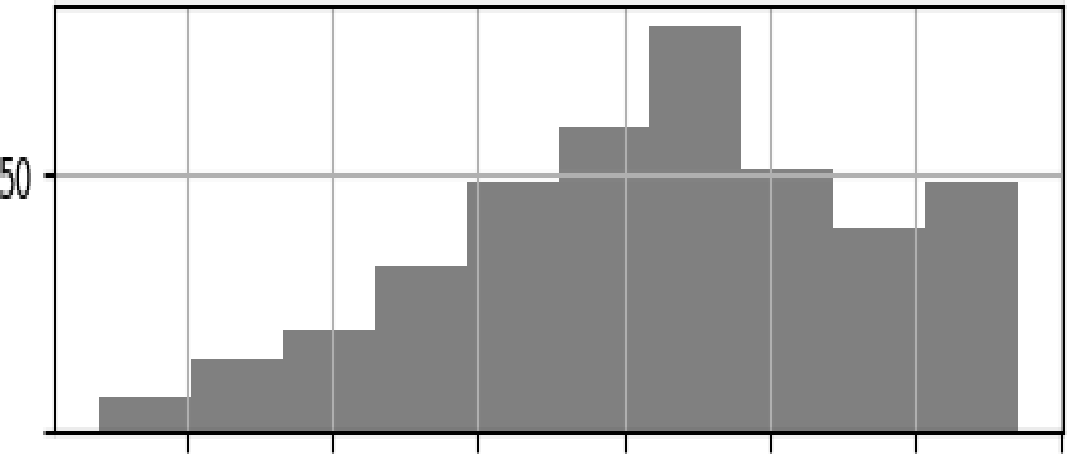
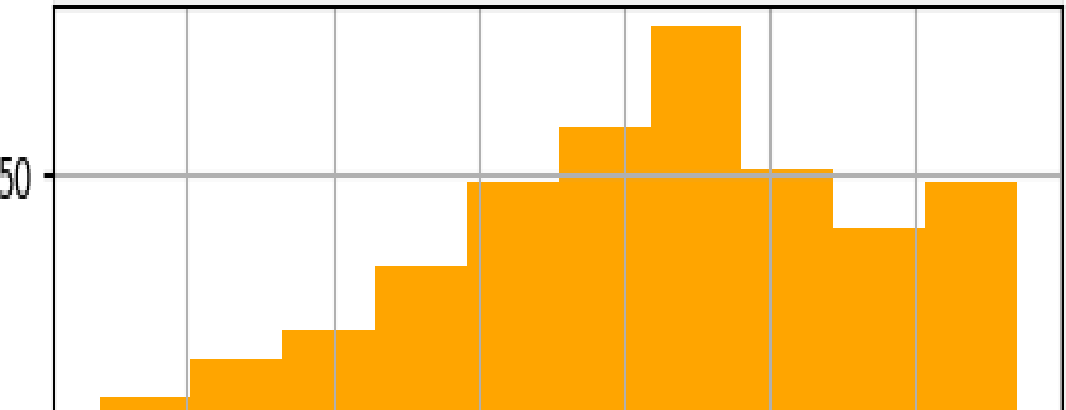
**2.0**

**2.5 3.0 3.5 4.0**

**4.5 5.0**

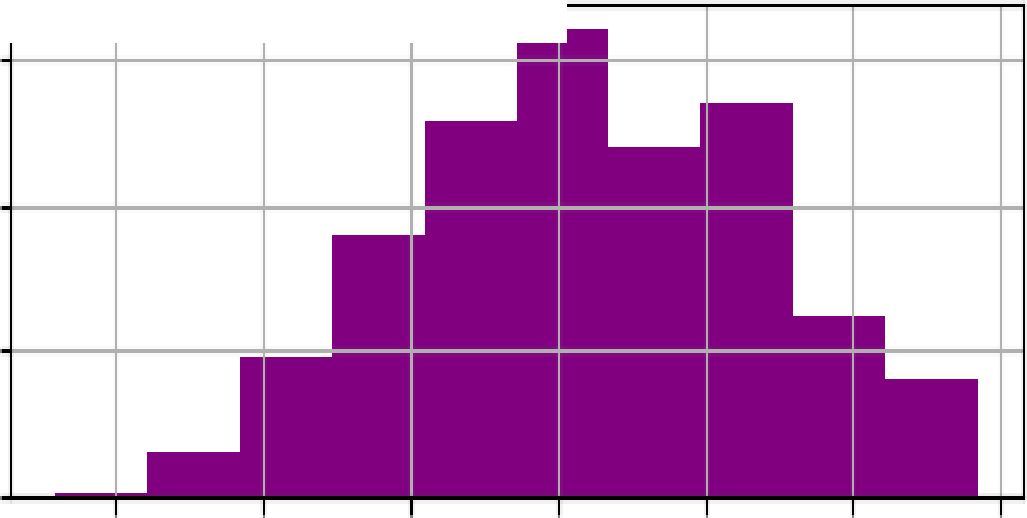
Research Chance of Admit

0 0



0.4 0.5 0.6 0.7 0.8 0.9 1.0 0.4 0.5 0.6 0.7 0.8 0.9 1.0

LOR CGPA



75 75

50 50

25 25

0

7.0 7.5 8.0 8.5 9.0 9.5 10.0

0

7.0 7.5 8.0 8.5 9.0 9.5 10.0

+ Code + Text

MinMaxScaler

0.. ;:: [11]

from sklearn. preprocessirlg sc = MinMaxScaler-() x=data.iloc[:,0:7].values

import

{x}

D

0

[:,

y=data.iloc[:,7:].values print{y)

<>

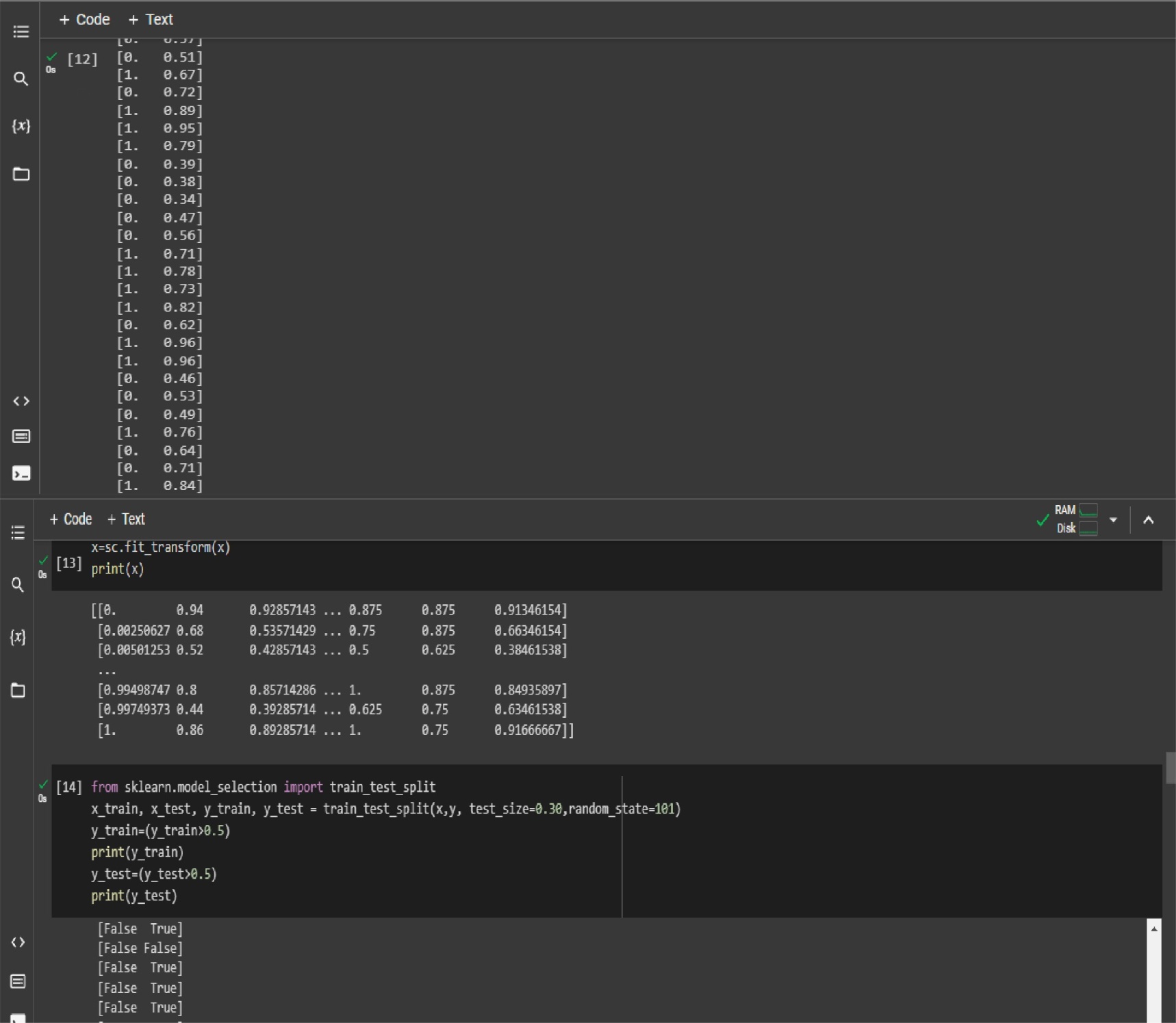
|  |  |
| --- | --- |
| [0. | 0.58] |
| [0. | 0.59] |
| [0. | 0.47] |
| [0. | 0.49] |
| [0. | 0.47] |
| [0. | 0.42] |
| [0. | 0.57] |
| [0. | 0.62] |
| [1. | 0.74] |
| [1. | 0.73] |
| [1. | 0.64] |
| [0. | 0.63] |
| [0. | 0.59] |
| [0. | 0.73] |
| [1. | 0.79] |
| [1. | 0.68] |

13

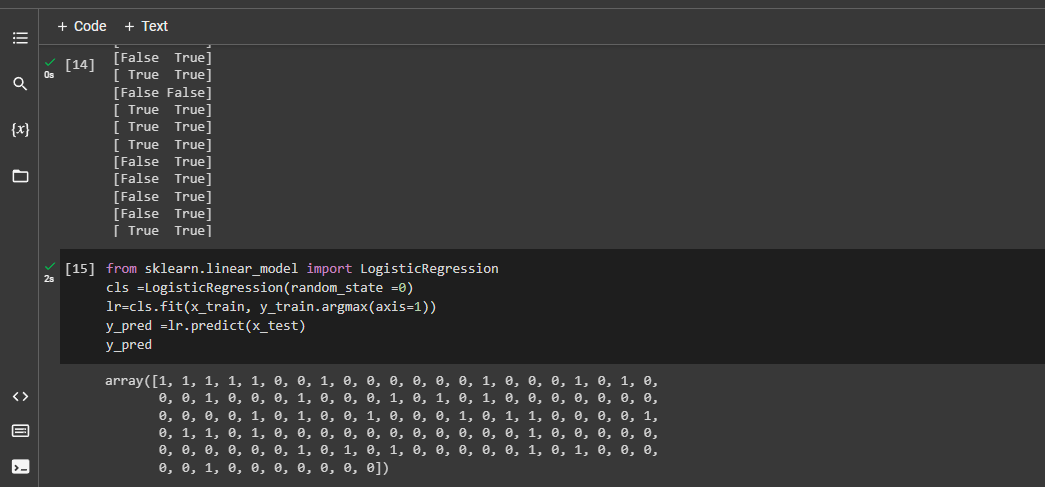
l!I

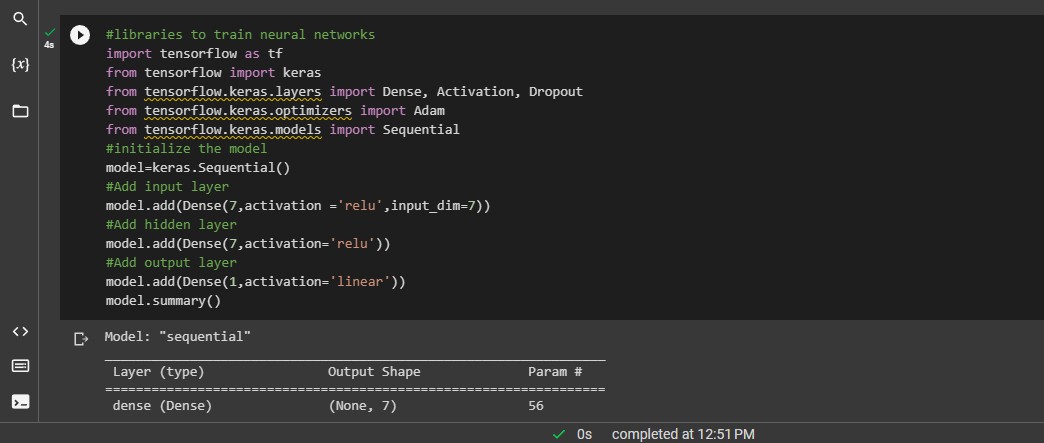
- - -

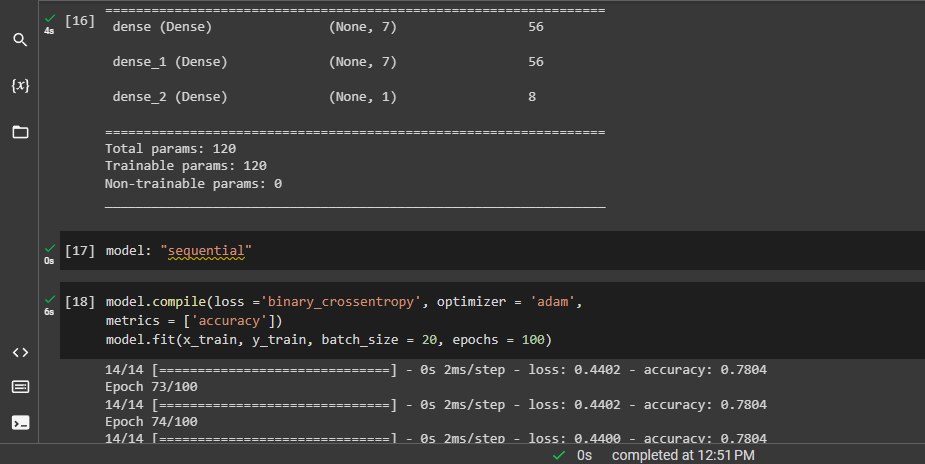
Os completed at 12:51 PM

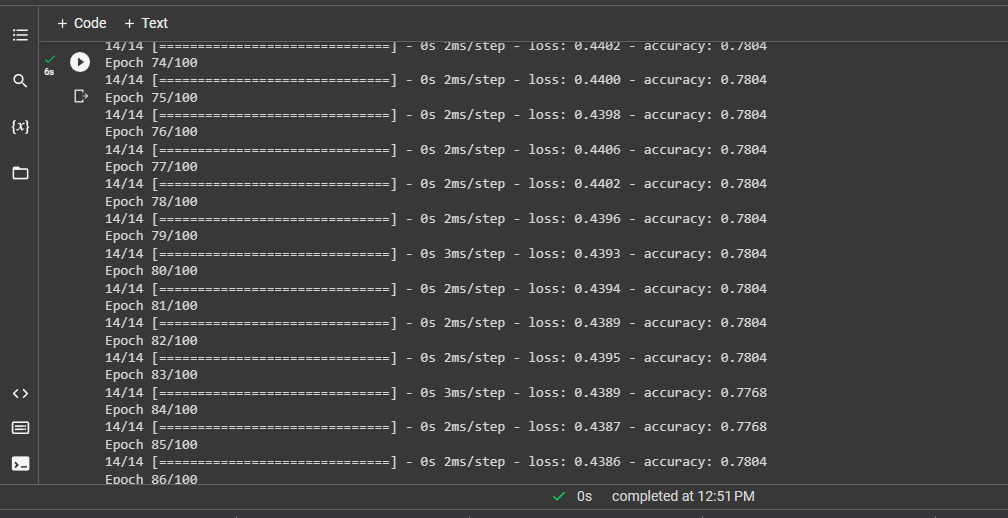


**Milestone 4:**









a,\_

**{x}**

D

<>

El

+ Code +

fi• **0**

Epoch

**14/14**

Epoch

**14/14**

Epoch

**14/14**

Epoch

**14/14**

Epoch

**14/14**

Epoch 14/14

Epoch

**14/14**

Epoch

**14/14**

Epoch

**14/14**

Epoch

**14/14**

Epoch

**14/14**

Epoch

**14/14**

Epoch

1 **,1 /1 ,1**

[•

Text

88/100 [==============================] - 89/100 [==============================] - 90/100 [==============================] - 91/100 [==============================] -

|  |  |
| --- | --- |
| 0s | 2ms/step |
| 05 | 2ms/step |
| 05 | 2ms/step |
| 0s | 2ms/step |
| 0s | 2ms/step |
| 0s | 2ms/step |
| 0s | 2ms/step |
| 0s | 3ms/step |
| 05 | 2ms/step |
| 0s | 2ms/step |
| 0s | 2ms/step |
|  | 2ms/step |

92/100 [==============================] -

93/100 [==============================] -

94/100 [==============================] - 95/100

[==============================] - 96/100 [==============================] - 97/100 [==============================] - 98/100 [==============================] - 99/100

[==============================] - 5

0

**')me/ct-an**

Os completed at 12:51 PM

,1:n,; - **'Jrr11,.-.:J1ru•** "' "771,1<

**rice•**

- loss: 0.4382 -

- loss: 0.4381 -

|  |  |
| --- | --- |
| accuracy: | 0.7804 |
| accuracy: | 0.7804 |
| accuracy: | 0.7804 |
| accuracy: | 0.7804 |
| accuracy: | 0.7804 |
| accuracy: | 0.7804 |
| accuracy: | 0.7768 |
| accuracy: | 0.7804 |
| accuracy: | 0.7804 |
| accuracy: | 0.7768 |
| accuracy: | 0.7804 |
| accuracy: | 0.7804 |

- loss: 0.4379 -

- loss: 0.4380 -

- loss: 0.4380 -

- loss: 0.4378 -

- loss: 0.4378 -

- loss: 0.4400 -

- loss: 0.4391 -

- loss: 0.4396 -

- loss: 0.4378 -

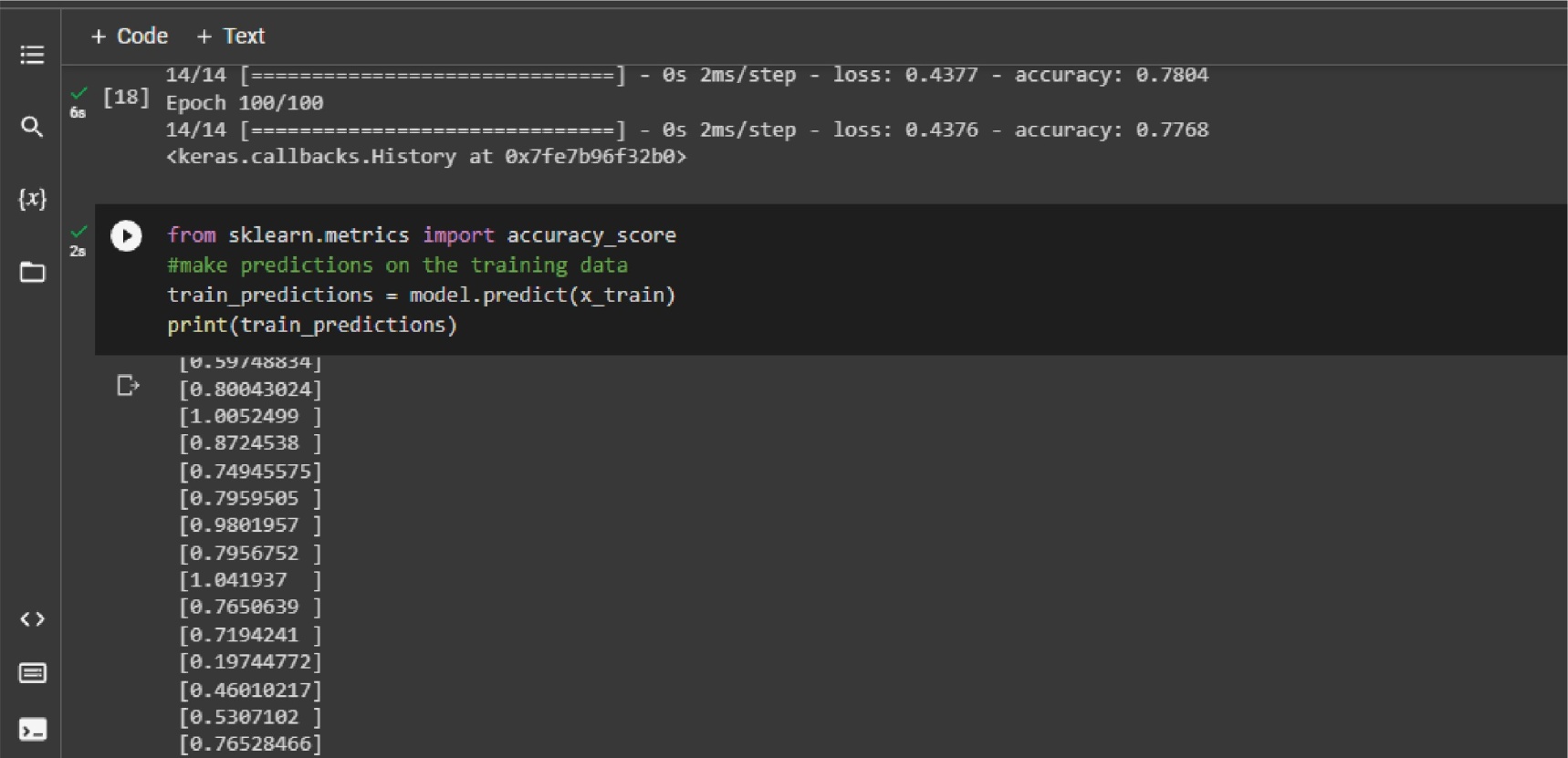
- loss: 0.4377 -

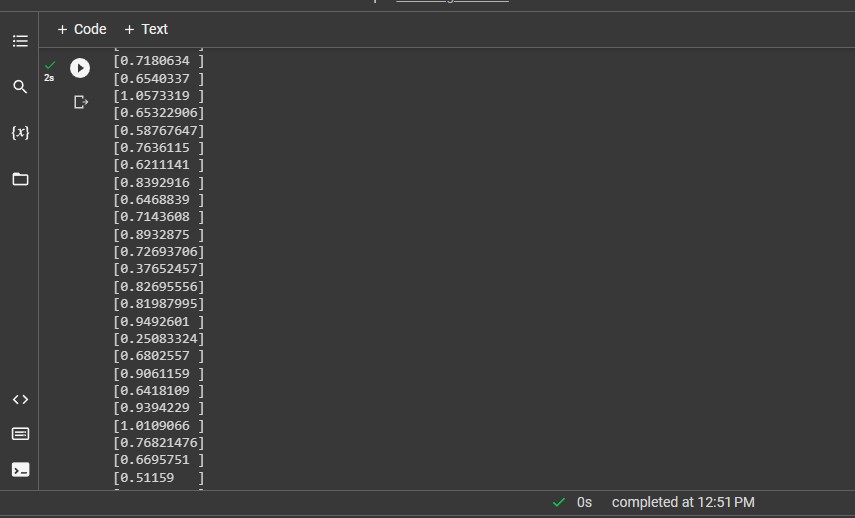
I!!

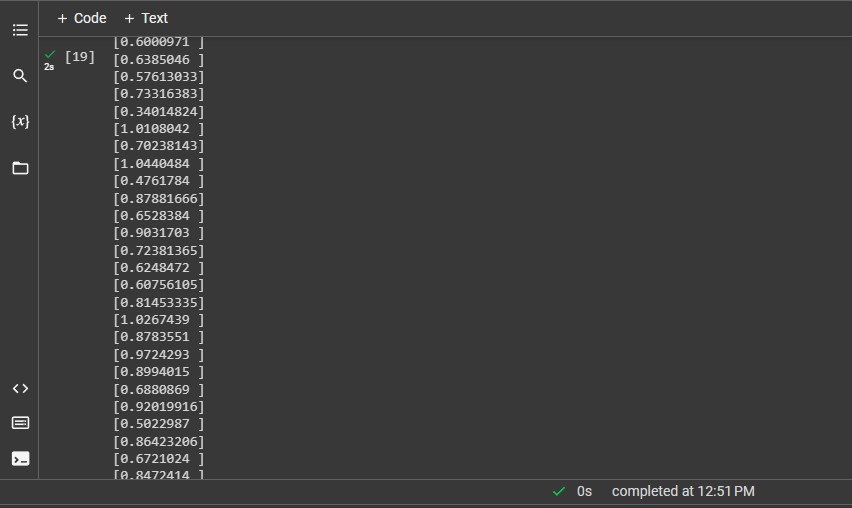
100/100

r------------------------------7 -

- - **l**







+ Code + Text

1. #get the training accuracy

Q.

1&

{x)

□

train ace = rnodel.evaluate(x\_train, y\_train, verbose=0)[1] train ace

,0.7767857313156128

1. #get the test accuracy

Os

test\_acc = rnodel.evaluate(x\_test, y\_test, verbose=0)1[1] print(test\_acc)

,0.6958333253860474

0& 0 pred=rnodel ..

predict(x\_test)

pred = (pred:>0 ..5)

pred

<> [:" [ True],

[ True],

True],

True],

True],

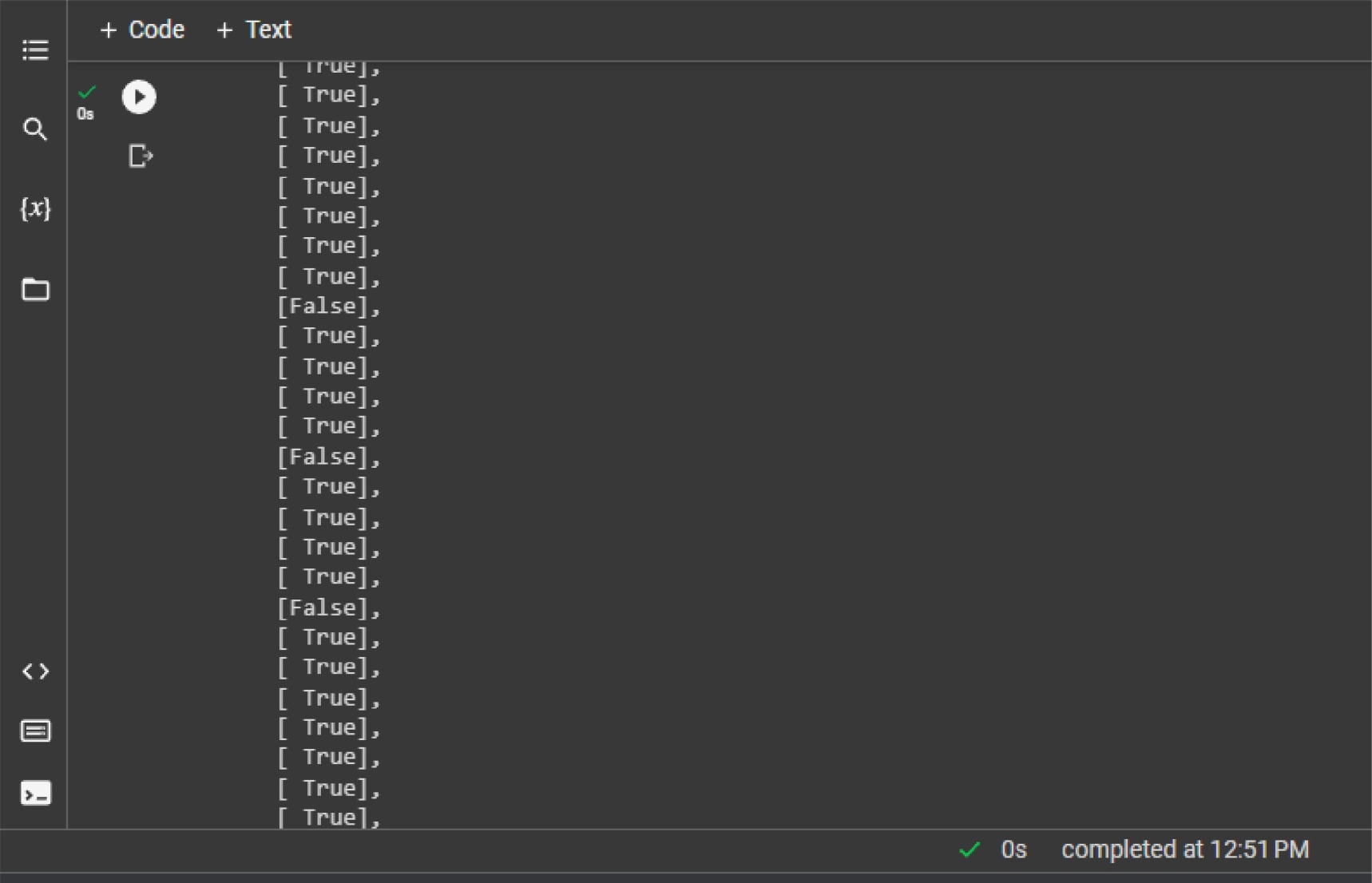
§] [

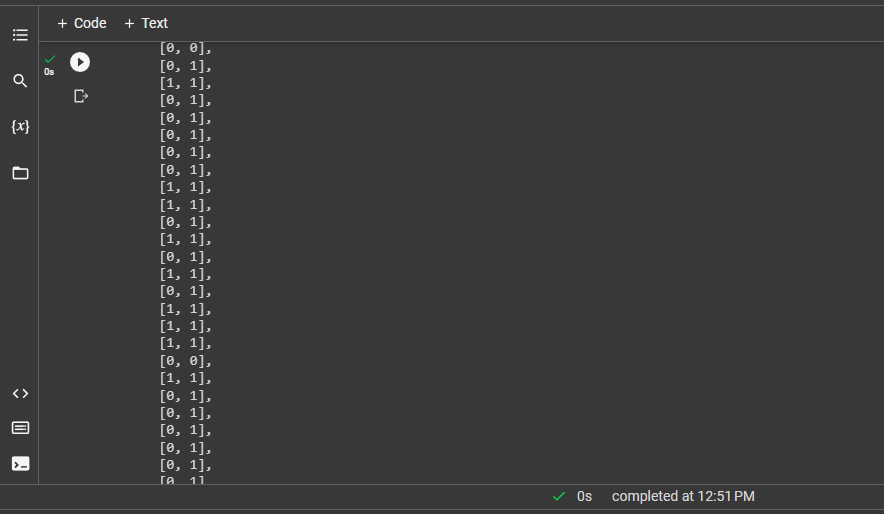
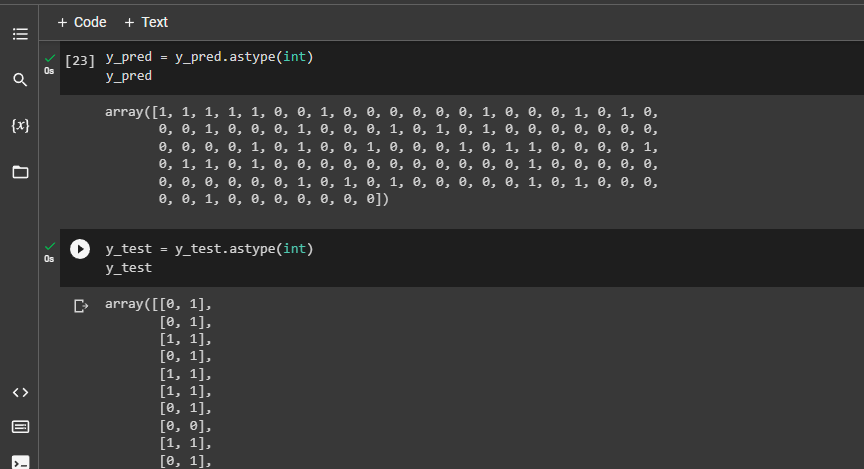
[

l!I [

LTrue\_l.

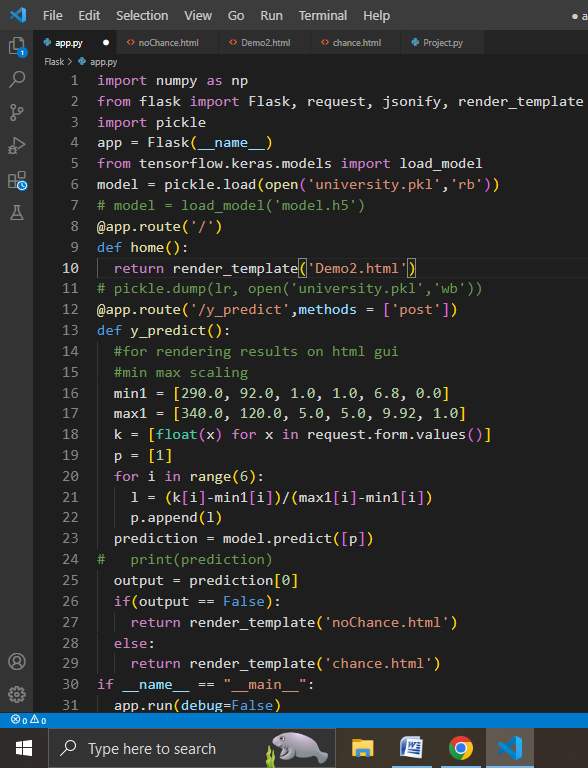
Os completed at 12:51 PM



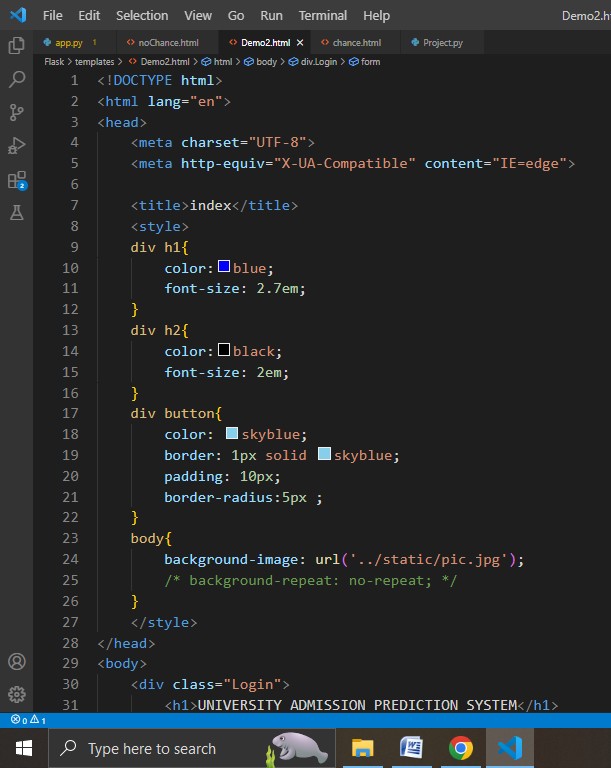
**Milestone 5:**

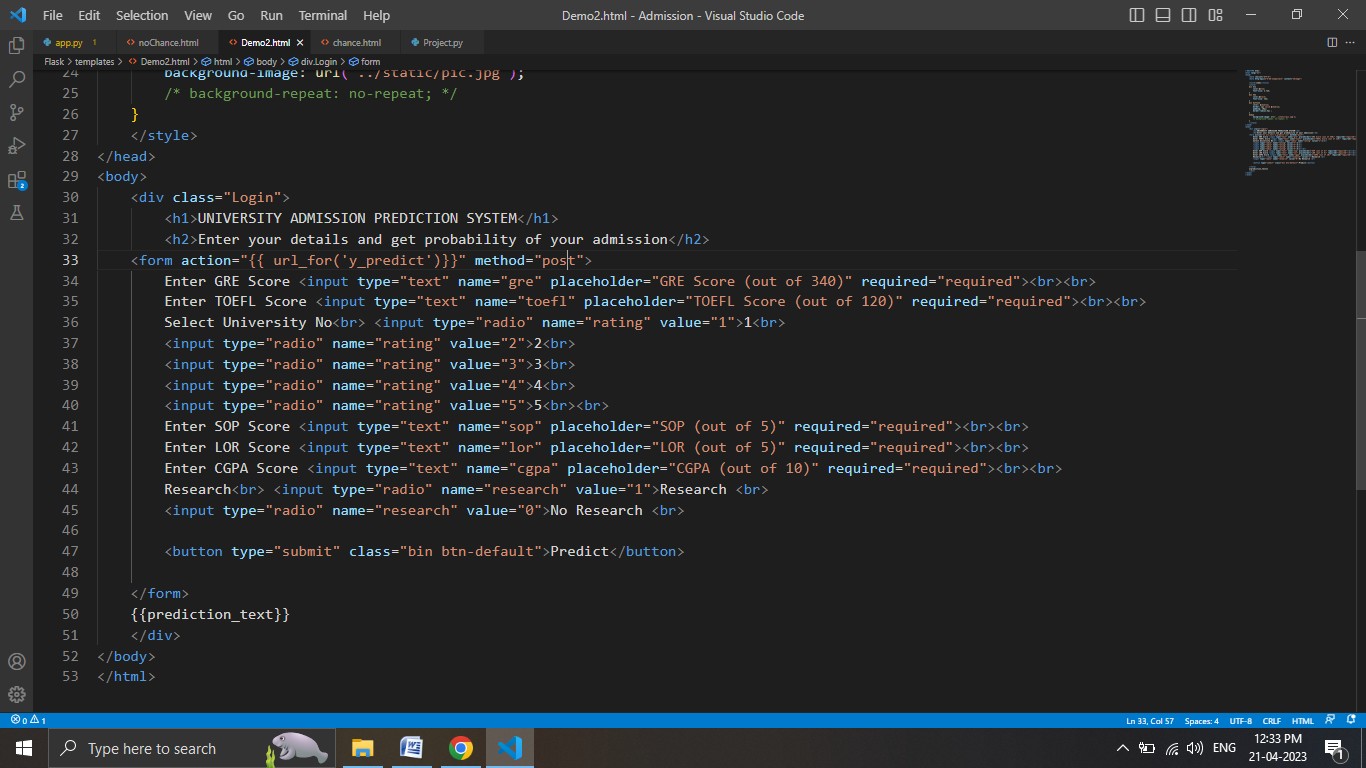
**Milestone 6:**

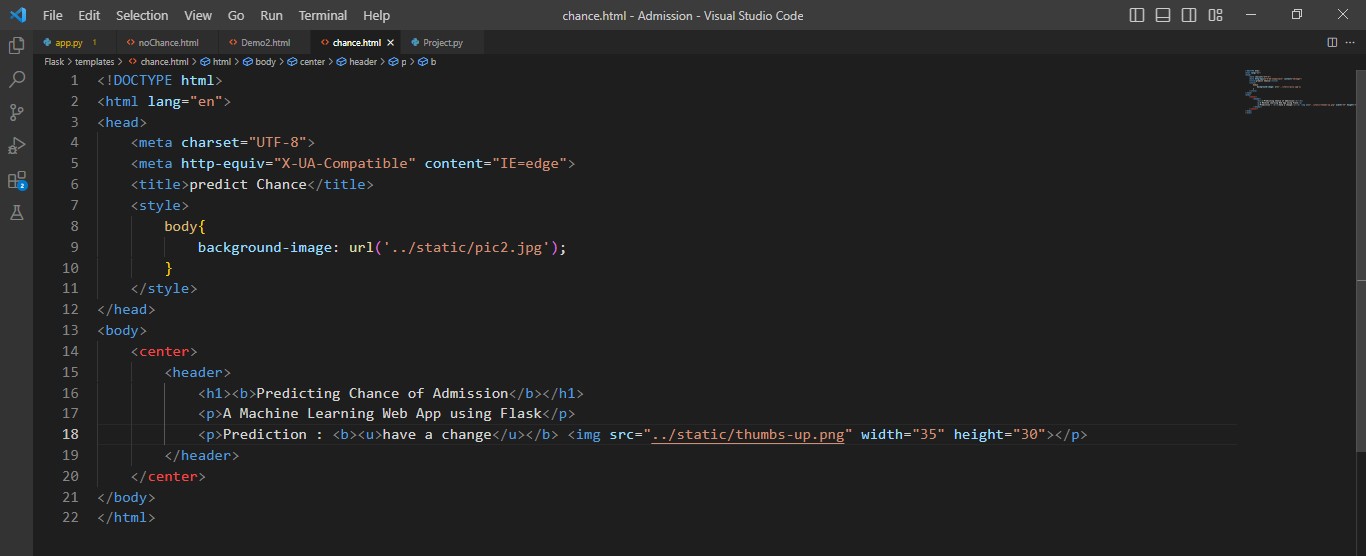
**Flask file ( app.py ):**

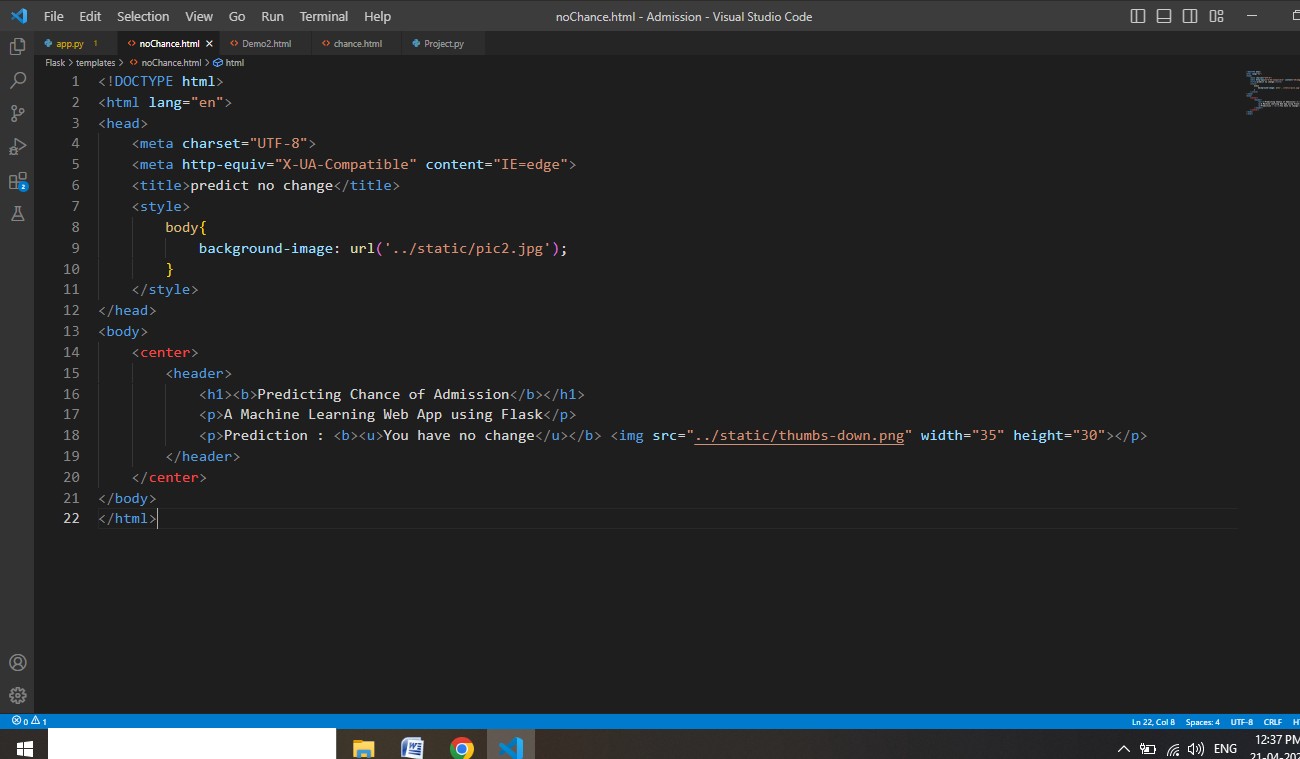


**User Interface ( index.html ):**









*P* Type here to ,earch