SRM Institute of Science and Technology,

Ramapuram Campus, Chennai-89

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Tamil Nadu Engineering Admissions Predictor

Batch No: 3

DETAILS OF THE PROJECT MEMBERS:

1)A P AISHWARYA LAKSHMI (RA2011026020066)
2) S INDHUMATHI (RA2011026020088)
3) P RIDHANYA (RA2011026020104)

SUPERVISOR DETAILS:

Mr. T. RAJESH

Dr. T.P. LATCHOUMI

OBJECTIVE

Objective :

The aim of the project is to predict the college that a student will be getting into with their PCM marks and caste.

Project Domain:

The domain of the project is Machine Learning. We have used Supervised Learning techniques to obtain the desired results.

ABSTRACT

Increasing student admission and enrollment, especially in engineering and computing programs, is a desirable goal for many universities. At the same time, this goal can be difficult to achieve. The aim of this project is to develop a predictive model that can be used by universities and colleges to improve student admission and enrollment process.

Predictive analytics is the technique of using historical data to create, test, and validate a model to best describe and predict the probability of an outcome.

And we have various institutes training children just to get them into top colleges like IITs, and MITs.There has also been an increasing demand for courses like Computer Science, especially Big Data, IoT, Artificial Intelligence, Cloud computing, and such. To make any process easier, visualization and proper understanding of data is needed, or the ability to predict

INTRODUCTION

The purpose of this project is to predict college admissions in Tamil Nadu. Applying to college can feel uncertain and even mysterious. Students and supporters are inundated with information through marketing and media, and it quickly becomes an overwhelming experience.

A college admissions predictor is a tool that uses predictive analysis to predict a student's chances of being admitted to a particular college or university. It takes into account various factors that are typically considered by admissions officers, such as a student's academic performance, test scores, extracurricular activities,etc.

Using a college admissions predictor can help students and their families make more informed decisions about which colleges to apply to, and how to allocate their time and resources in the application process.

INTRODUCTION

By inputting their credentials and background information, students can estimate their chance of being admitted to a particular college, and can use this information to prioritize their applications and set realistic expectations.

It is important to note, however, that college admissions predictors are not perfect, and should be used as just one tool in the college search and application process. Admissions decisions are often influenced by subjective factors that cannot be easily quantified, and each college has its own unique admissions process and criteria.

Nevertheless, a college admissions predictor can be a useful starting point for students as they navigate the complex and competitive world of college admissions.

SCOPE

The scope of a college admissions predictor, at a basic level, a predictor can provide a simple estimate of a student's chances of being admitted to a particular college or university based on their academic credentials and other factors.

Another aspect of the scope of a college admissions predictor is its level of transparency and accessibility. A good predictor should be transparent about the data and models used to make predictions, and should provide clear explanations of how the predictions are generated.

The predictor should also be accessible to all students, regardless of their background or resources, and should be designed in a way that is easy to use and understand.

SYSTEM REQUIREMENTS

Hardware Requirements:

1. CPU: A modern multicore processor, such as an Intel Core i5 or i7, is recommended for training and deploying machine learning models.
2. RAM: A minimum of 8GB of RAM is recommended, although more may be required for larger datasets.
3. Storage: A large amount of storage is required to store and process large datasets. Solid-state drives (SSDs) are recommended for fast access to data.

SYSTEM REQUIREMENTS

Software Requirements:

1. Operating System: Most machine learning frameworks are compatible with popular operating systems such as Windows, macOS, and Linux.
2. Machine Learning Frameworks: There are many machine learning frameworks available, such as TensorFlow, Keras, and scikit-learn..
3. Libraries: Libraries such as NumPy, Pandas, and Matplotlib are commonly used for data manipulation, analysis, and visualization.

RAMAPURAM SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)

RELATED WORKS

S.NO	TITLE	AUTHOR	METHODOLOGY	TECHNICAL GAP
1.	Using Technology in Undergraduate Admission (2008)	Robin Lindbeck and Brian Fodrey	To identify the current practices and future plans for using technology in admission practices at four year colleges and universities.	The small collection of institutions used in this study makes the results interesting, but far from generalizable. Gathering data from a larger number of institutions and from a wider geography will enhance

2	Gen Z Students' Experiences with College Choice (2014)	Heather Levesque-(East Tennessee State University)	Data mining	The investigation is narrowed down and has not been done for a large scale population.
3	A graphical modeling of student admissions and faculty recruitment problems (2015)	Baiou and Balinsk	A study and analysis of admission data and student educational outcomes is presented in Heinesen	However, the study found no clear evidence that being admitted to one of the higher degree programs listed on the application has an impact on years of education

4.	From early aspirations to actual attainment: “The effects of economic status and educational expectations on university pursuit." (2016)	Ching-Ling Wu and Haiyan Bai	This study investigated the effects of economic status and the educational expectations of significant others on early university aspirations and actual university attainment. .	The criterion which they used to judge is based on one’s economic status.

5.	Research on time series data prediction based on clustering algorithm (2014)	Yaebau	Clustering methodology	Based on this situation, this paper analyzed the data of Yuebao, and according to the user's attributes and the operating characteristics, this paper classified 567 users of Yuebao, and made further predicted the data of Yuebao for every class of users, the results showed that the forecasting model in this paper can meet the demand of forecasting.

6.	Mining Time Series Data with AprioriTid Algorithm (2016)	Hiran Kumar Deva Sharma, Swapnil Mishra	Both the association rule mining algorithms Apriori Tid and modified Apriori Tid are implemented over time series data.	The performances of both the algorithms in terms of computation time requirements for generating frequent itemsets are analyzed and correspondin g corrections are to be made.

7.	ARIMA Time Series Application to Employment Forecasting (2008)	Xiaoguo Wang, and Yuejing Liu	The paper establishes an ARIMA model on the employment information of the computer industry from 2002 to 2007 in China, and using the model, gives a prediction of the situation in 2008.	The paper establishes an ARIMA model on the employment information of the computer industry from 2002 to 2007 in China, and using the model, gives a prediction of the situation in 2008.

8.	Performance comparison and future estimation of time series data using predictive data mining techniques (2018)	Harshita Tanwar; Misha Kakkar	Two models namely linear regression model and ARIMA model are used for analyzing the future prediction	These two models are analyzed with respect to standardized error generated by them, fitted values, residuals, standardized error, square root standardized error are used for forecasting future expenditure prediction. Result shows that both models accurately and approximately predict the same future Expenditure measure.

9.	Decision Tree Classification and Forecasting of Pricing Time Series Data (2016)	Emil Lundkvist	To make forecasts of the product prices within each class, methods of time series analysis were applied and VAR structures were chosen as a model for the data	Although the tool described in this thesis is already distributed in the company and works as it was intended, there are of course many improvement s that can be made.

10.	Using Technology in Undergraduate Admission A Student Perspective (2017)	Robin Lindbeck and Brian Fodrey	Supervised learning	This inquiry offers several opportunities for additional research. First, a small geographically homogenous convenience sample was used in this inquiry. So, although the results raise some interesting ideas, they are not representative of the admission experience for all students.

Existing System

Traditionally, institutions have advertised themselves by posting information on their websites and using multimedia. However, these traditional methods are increasingly becoming insufficient on their own .

Therefore, they should be supported by a predictive analytics approach that utilizes personal attributes to appeal to the interests of prospective students; and predict the probability that students will accept an offer and enroll into a course.

More so, predictive analytics can provide accurate information and knowledge about future admission trends, and thus support planning, resource allocation, and decision making regarding the growth of an institution. If the school has an anticipation for a growth in student enrollment then they can plan accordingly to provide adequate resources required to educate students.

Proposed System

The idea is to build a College admissions predictor.

● A college admissions predictor would be a valuable tool for high school students, who are often unsure of their chances of admission to different universities. Such a predictor could consider factors such as PCM, caste, and so on and use machine learning algorithms to make predictions based on historical data.

● A college admissions predictor can be created using machine learning algorithms that analyze a variety of data points. Here's a list of data points that are included in the predictor:

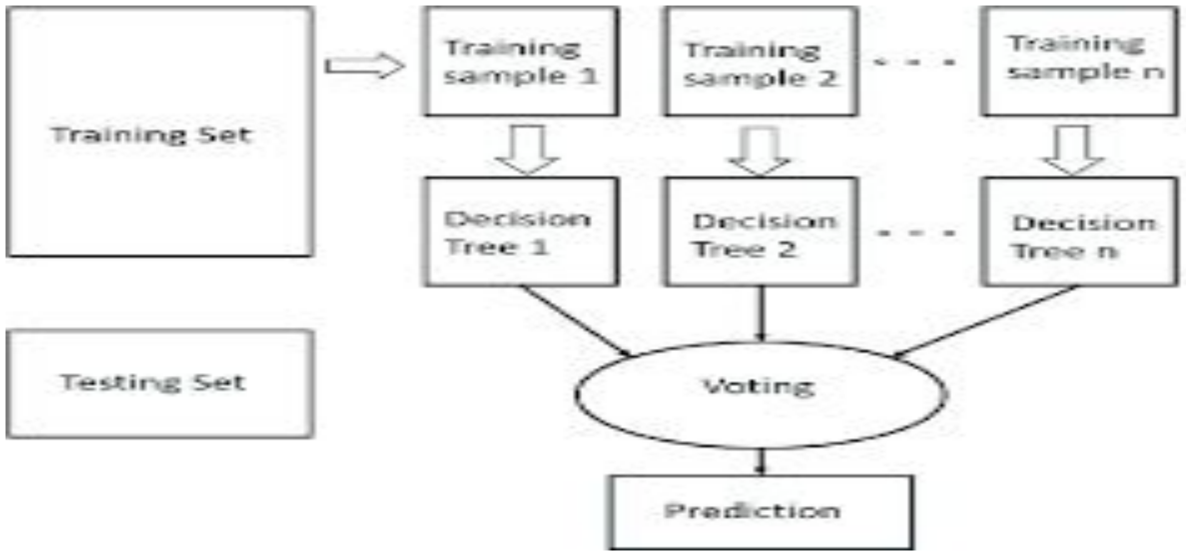
1. Physics, Chemistry and Mathematics (PCM)
2. College name
3. Branch name
4. Demographic information (Caste)

● Once these data points are collected and organized, model is trained using machine learning algorithms to predict a student's likelihood of being admitted to certain colleges or universities.

● The predictor can be created by training a model on historical admissions data, where the model learns from the patterns and trends in past applicants that were accepted or rejected from different colleges. This model can then be used to analyze new applicants and determine their chances of admission.

● This predictor can be made available as a free online tool for high school students to use.However, it's important to note that college admissions are a complex process, and other factors may also influence admissions decisions.

Architecture Diagram



MODULE DESCRIPTION

1. **Data pre-processing**:

   This module would prepare the data for analysis, including cleaning the data, imputing missing values, and scaling or transforming the data as necessary.

2. **Creating/ Building the Model :**

- **Model training**: This module would train the classification model on the input and output variables using a chosen algorithm. The algorithm could be linear regression, multiple linear regression, Random forest Classifier multiouput classification or another type of regression/ classification algorithm.

- **Prediction**: This module would use the trained model to predict the values of the output variables for new input data. It could also provide uncertainty estimates or confidence intervals for the predictions.

MODULE DESCRIPTION

3. **Creating the user interface**:

This module would create a user interface using the Streamlit library that allows users to input their data and receive a predicted outcome for their college admissions chances. The user interface can include fields for entering PCM marks and castes.

- **Deploying app: This module would** deploy the app using Streamlit to make it accessible over the internet.

Algorithm Used in each module

1. Data Collection and Data pre-processing: `Nominal encoding(Integer representation), fillna() method`

2. Building and Training the model: `RandomForestClassifier`

3. Making predictions: `MultiOutputClassifier`

Module 1 :

1. **Data collection:**
   TNEA (Tamil Nadu Engineering Admissions) is a website managed by the Directorate of Technical Education, Tamil Nadu, India. The website provides information on the admission process for undergraduate engineering programs in the state of Tamil Nadu.

   Admission data: The website collects data on the admission process, including the cutoff marks for various engineering courses, the counseling schedule, and the seat allotment process.

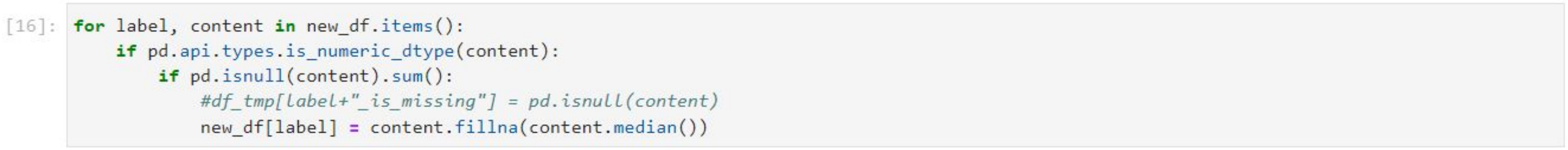
   The TNEA website provides a wealth of information on the admission process for engineering programs in Tamil Nadu, including data on colleges, courses, and the admission process itself.

Module 1 :

This data can be used by students, parents, and education professionals to make informed decisions about college admissions in Tamil Nadu.

https://cutoff.tneaonline.org/

1. **Data pre-processing:**
   Filling missing values with the median of all values.



```
[16]: for label, content in new_df.items():
          if pd.api.types.is_numeric_dtype(content):
              if pd.isnull(content).sum():
                  #df_tmp[label+"_is_missing"] = pd.isnull(content)
                  new_df[label] = content.fillna(content.median())
```

Module 1

Filling missing values with median values is a technique used in data preprocessing to handle missing data in a dataset. When a dataset contains missing values, it can cause problems for machine learning models, which may not be able to handle missing data. Filling in missing values with median values is a common technique used to address this problem. Here's how it works:

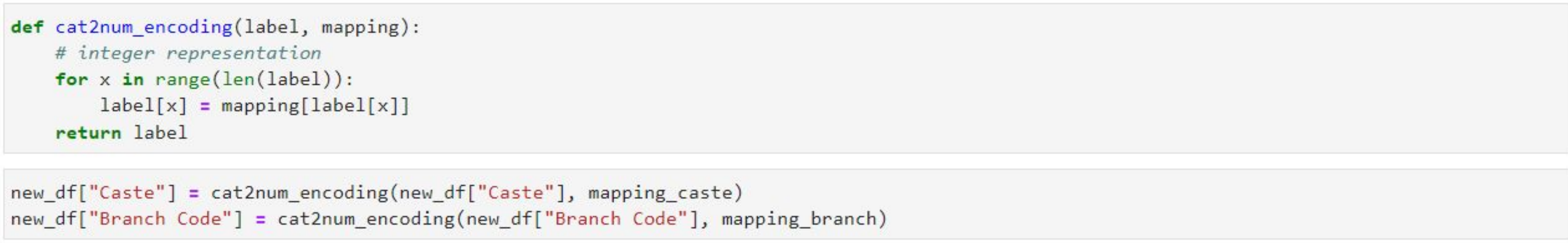
1. Identify missing values: First, identify the missing values in the dataset. This can be done using a variety of techniques, such as using the Pandas library in Python to search for null values.
2. Calculate the median: Next, calculate the median value for each column in the dataset. The median is the middle value in a dataset when the values are sorted in ascending order. If the dataset contains an even number of values, the median is the average of the two middle values.
3. Fill missing values: Finally, replace the missing values in each column with the median value for that column. This is done using the Pandas library in Python, which provides a method called "fillna" that can be used to fill in missing values with a specified value.

Filling missing values with median values is a simple and effective way to handle missing data in a dataset. It can help to improve the accuracy of machine learning models by ensuring that all data is complete and consistent.

Module 1 :

Multi output Classifier Model :

- **Algorithm Used :** RandomForestClassifier



```
def cat2num_encoding(label, mapping):
    # integer representation
    for x in range(len(label)):
        label[x] = mapping[label[x]]
    return label
```

```
new_df["Caste"] = cat2num_encoding(new_df["Caste"], mapping_caste)
new_df["Branch Code"] = cat2num_encoding(new_df["Branch Code"], mapping_branch)
```

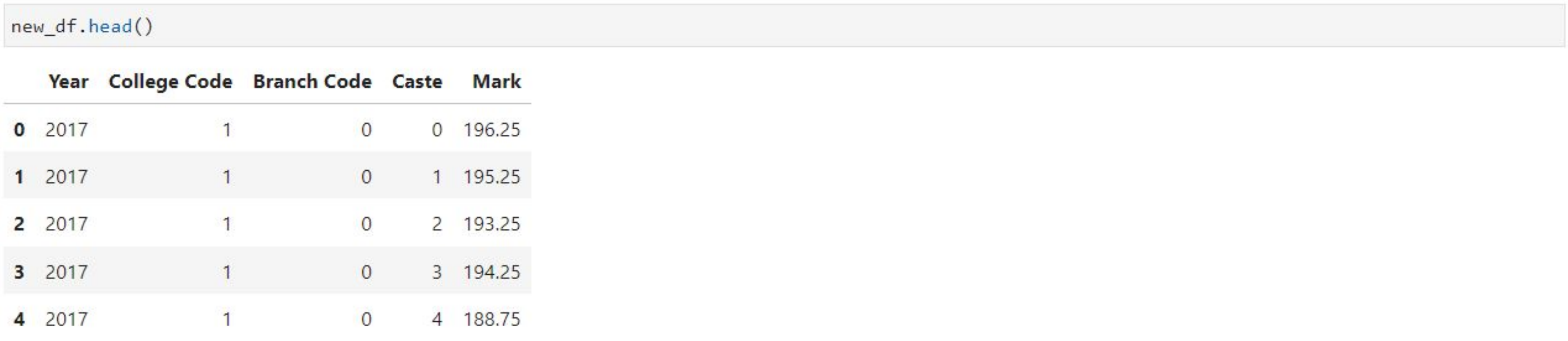
Module 1

Nominal encoding is a technique used in data preprocessing to convert categorical variables into a numerical format that can be used in machine learning algorithms. Categorical variables are variables that represent discrete values or categories, such as colors, names, or labels.

Nominal encoding involves assigning a unique numerical value to each category in the categorical variable. This is typically done using an integer encoding, where each category is assigned a different integer value. For example, if a dataset has a categorical variable "color" with three categories (red, green, and blue), each category might be assigned a unique integer value (red = 1, green = 2, blue = 3).

Module 1

NOMINAL ENCODING:



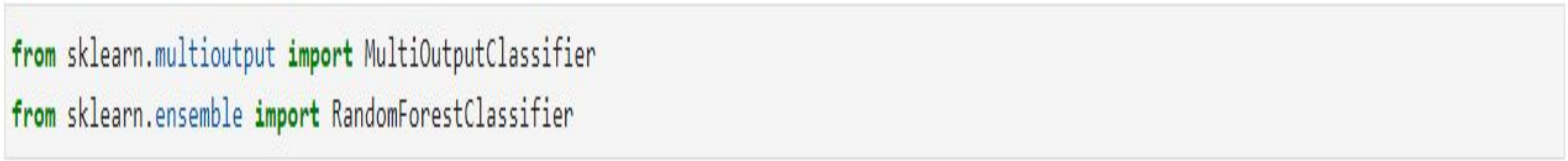
```
new_df.head()
```

	Year	College Code	Branch Code	Caste	Mark
0	2017	1	0	0	196.25
1	2017	1	0	1	195.25
2	2017	1	0	2	193.25
3	2017	1	0	3	194.25
4	2017	1	0	4	188.75

Module 2

BUILDING AND TRAINING THE MODEL

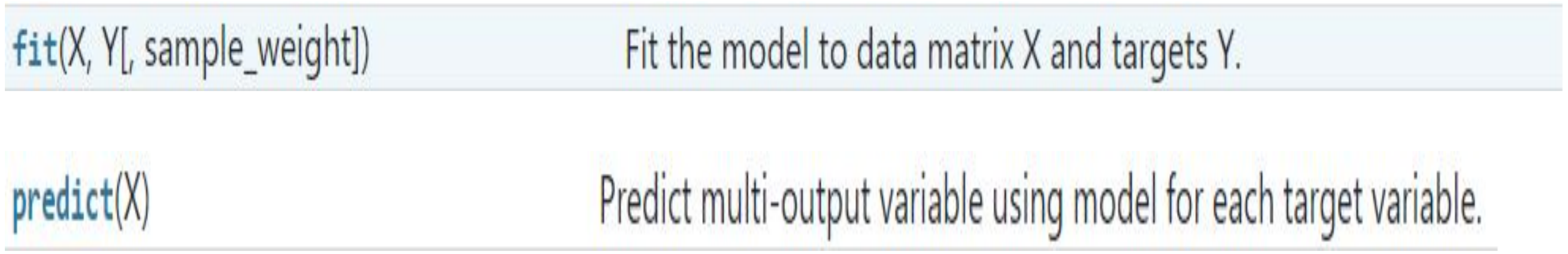
- Multi target classification.
- This strategy consists of fitting one classifier per target. This is a simple strategy for extending classifiers that do not natively support multi-target classification.



```
from sklearn.multioutput import MultiOutputClassifier
from sklearn.ensemble import RandomForestClassifier
```

Module 2

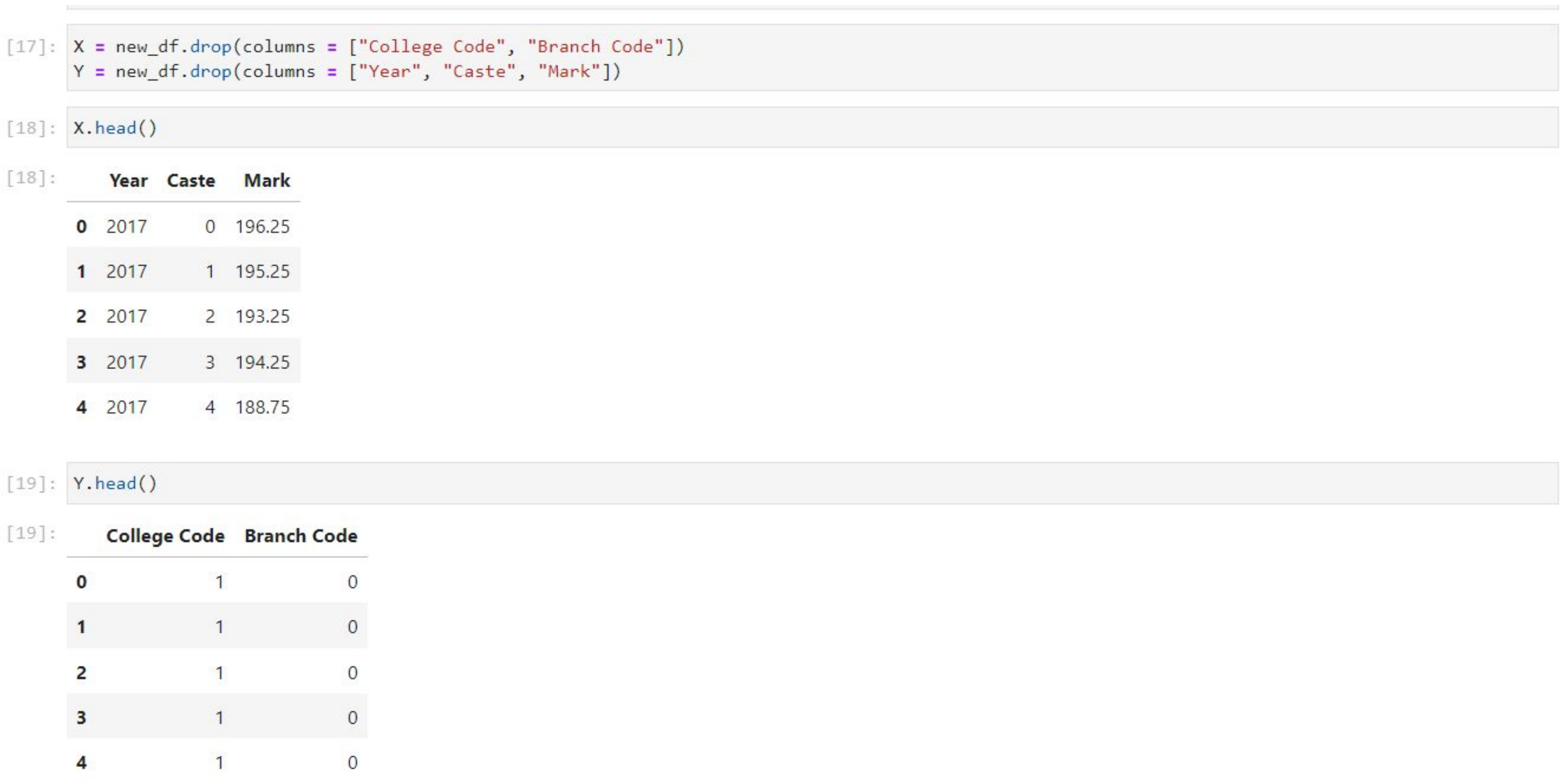
- Multi-output classification is a type of machine learning that predicts multiple outputs simultaneously. In multi-output classification, the model will give two or more outputs after making any prediction. In other types of classifications, the model usually predicts only a single output.
- An example of a multi-output classification model is a model that predicts the type and color of fruit simultaneously. The type of fruit can be, orange, mango and pineapple. The color can be, red, green, yellow, and orange. The multi-output classification solves this problem and gives two prediction results.



fit(X, Y[, sample_weight])	Fit the model to data matrix X and targets Y.
predict(X)	Predict multi-output variable using model for each target variable.

Module 2

TRAINING THE MODEL



```
[17]: X = new_df.drop(columns = ["College Code", "Branch Code"])
      Y = new_df.drop(columns = ["Year", "Caste", "Mark"])
```

```
[18]: X.head()
```

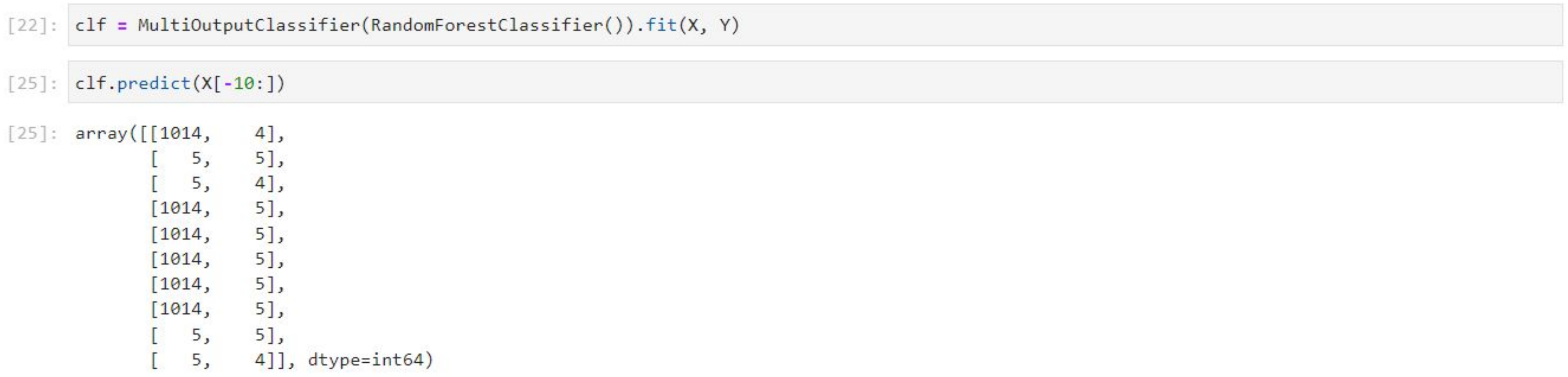
```
[18]:
      Year  Caste    Mark
   0  2017      0  196.25
   1  2017      1  195.25
   2  2017      2  193.25
   3  2017      3  194.25
   4  2017      4  188.75
```

```
[19]: Y.head()
```

```
[19]:
      College Code  Branch Code
   0             1            0
   1             1            0
   2             1            0
   3             1            0
   4             1            0
```

Module 2

MAKING PREDICTIONS: From the output above, the model has produced two prediction outputs. Therefore, we have successfully built our multi-output text classification model.



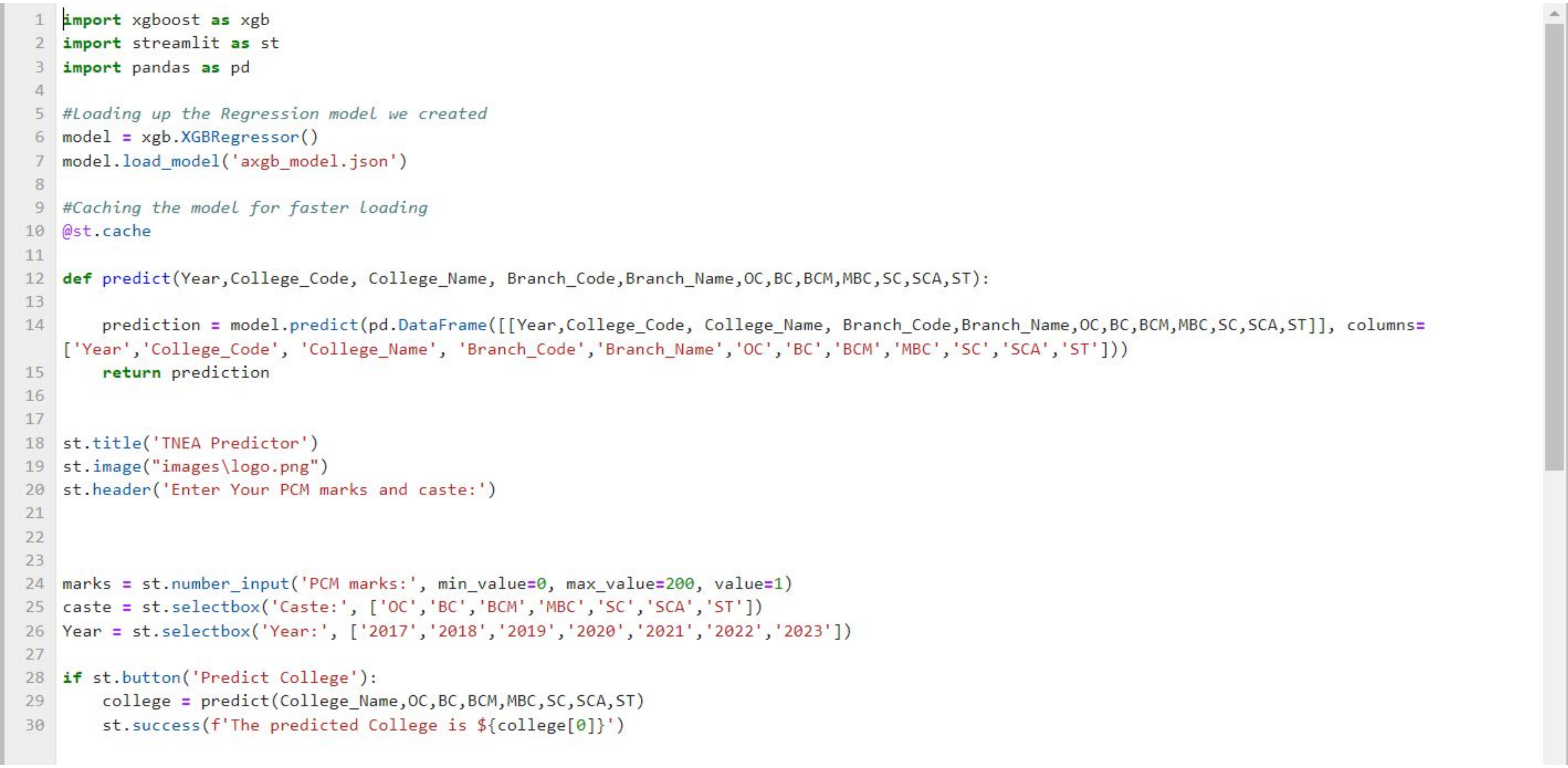
```
[22]: clf = MultiOutputClassifier(RandomForestClassifier()).fit(X, Y)

[25]: clf.predict(X[-10:])

[25]: array([[1014,    4],
             [   5,    5],
             [   5,    4],
             [1014,    5],
             [1014,    5],
             [1014,    5],
             [1014,    5],
             [1014,    5],
             [   5,    5],
             [   5,    4]], dtype=int64)
```

Module 3

Creating the User Interface



```python
import xgboost as xgb
import streamlit as st
import pandas as pd

#Loading up the Regression model we created
model = xgb.XGBRegressor()
model.load_model('axgb_model.json')

#Caching the model for faster loading
@st.cache

def predict(Year,College_Code, College_Name, Branch_Code,Branch_Name,OC,BC,BCM,MBC,SC,SCA,ST):

    prediction = model.predict(pd.DataFrame([[Year,College_Code, College_Name, Branch_Code,Branch_Name,OC,BC,BCM,MBC,SC,SCA,ST]], columns=
['Year','College_Code', 'College_Name', 'Branch_Code','Branch_Name','OC','BC','BCM','MBC','SC','SCA','ST']))
    return prediction


st.title('TNEA Predictor')
st.image("images\logo.png")
st.header('Enter Your PCM marks and caste:')



marks = st.number_input('PCM marks:', min_value=0, max_value=200, value=1)
caste = st.selectbox('Caste:', ['OC','BC','BCM','MBC','SC','SCA','ST'])
Year = st.selectbox('Year:', ['2017','2018','2019','2020','2021','2022','2023'])

if st.button('Predict College'):
    college = predict(College_Name,OC,BC,BCM,MBC,SC,SCA,ST)
    st.success(f'The predicted College is ${college[0]}')
```

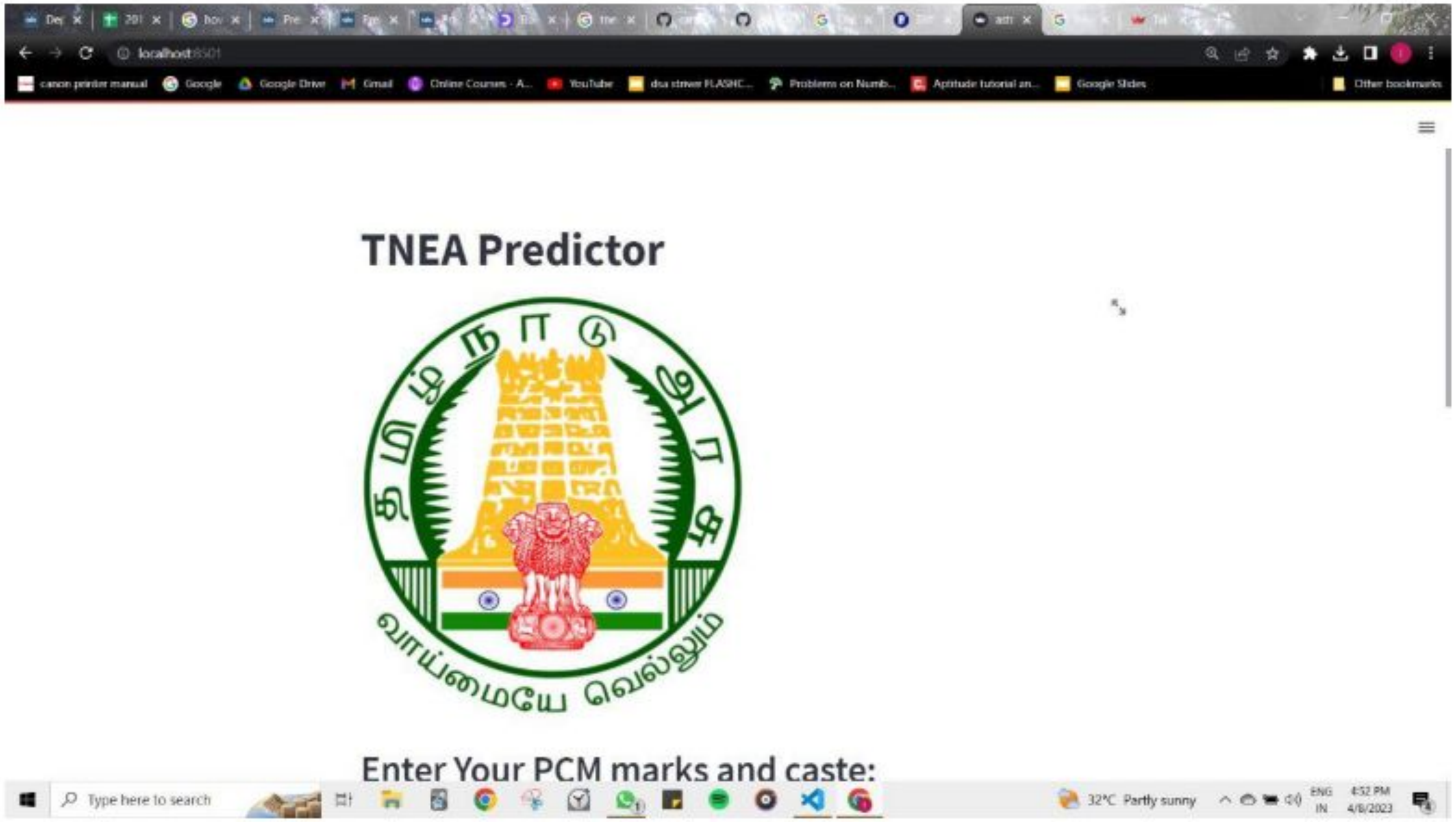
Module 3

Deploying the app:

Deploying a model using Streamlit refers to the process of creating a web application that allows users to interact with a trained machine learning model through a graphical user interface. Streamlit is a Python library that simplifies the process of building web applications and visualizations for machine learning models.

Once the app is deployed, users can access it through a web browser and interact with the machine learning model by entering input data and viewing the model's predictions or other results. The Streamlit app provides a user-friendly and accessible way to share machine learning models with a wider audience and enable them to benefit from the insights provided by the model.

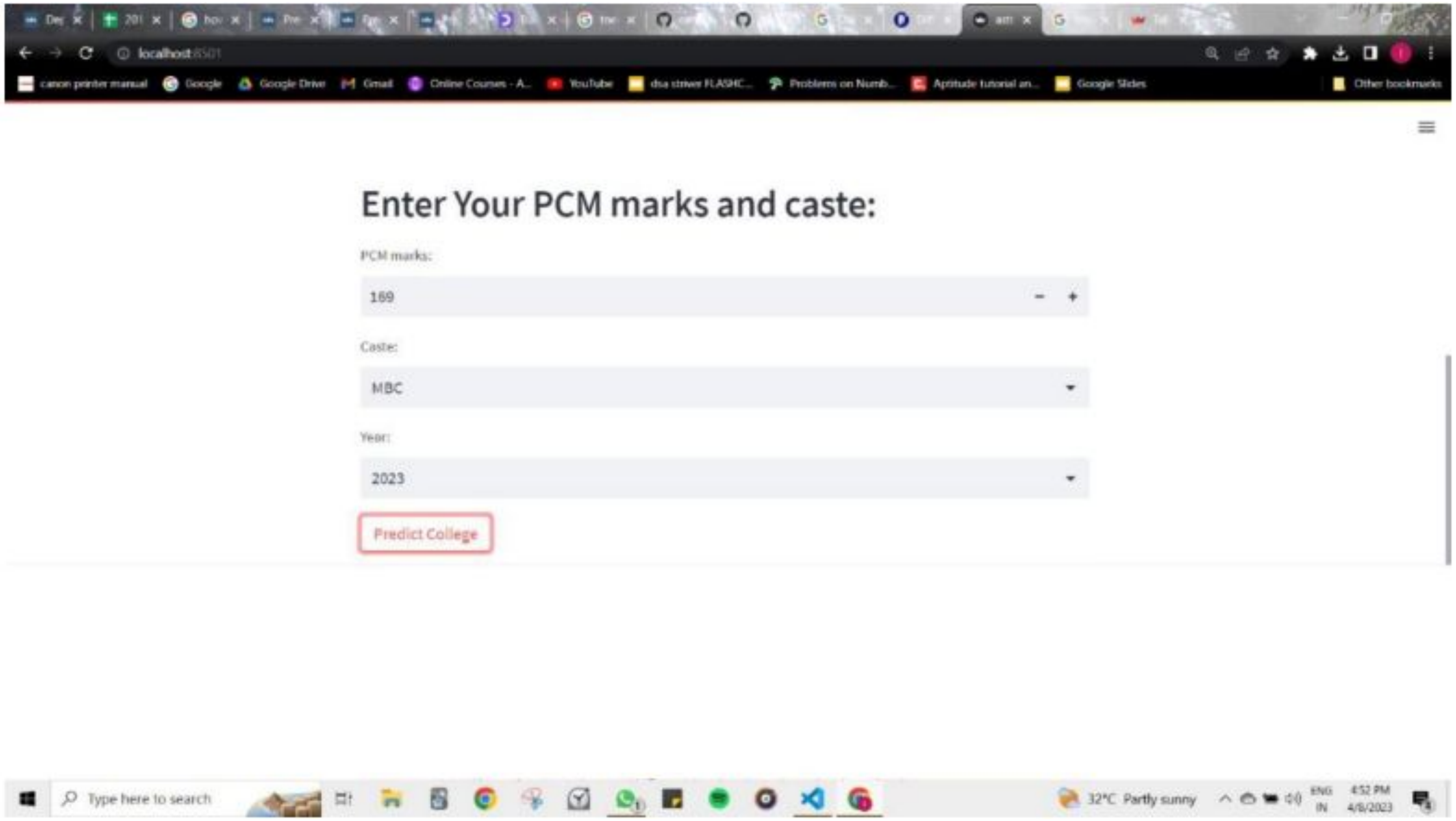
Module 3



TNEA Predictor

Enter Your PCM marks and caste:

Module 3



Enter Your PCM marks and caste:

PCM marks: 169

Caste: MBC

Year: 2023

REFERENCES

1. https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputClassifier.html
2. https://medium.com/nerd-for-tech/nominal-and-ordinal-encoding-in-data-science-c93872601f16#:~:text=types%20of%20encoding%2C-,Nominal%20Encoding,order%20or%20rank%2C%20or%20sequence.