**Intelligent Admissions: The Future Of University Decision Making With Machine Learning**

University admission is the process by which students are selected to attend a college or university. The process typically involves several steps, including submitting an application, taking entrance exams, and participating in interviews or other evaluations.

Students are often worried about their chances of admission in University. the university admission process for students can be demanding, but by being well-informed, prepared, and organized, students can increase their chances of being admitted to the university of their choice.

The aim of this project is to help students in short listing universities with their profiles. Machine learning algorithms are then used to train a model on this data, which can be used to predict the chances of future applicants being admitted. With this project, students can make more informed decisions about which universities to apply to, and universities can make more efficient use of their resources by focusing on the most promising applicants.The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

**Technical Architecture:**



**Project Flow:**

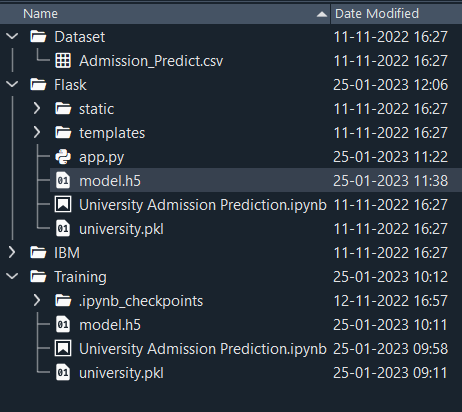
* User interacts with the UI to enter the input.
* Entered input is analysed by the model which is integrated.
* Once model analyses the input the prediction is showcased on the UI

To accomplish this, we have to complete all the activities listed below,

* Define Problem / Problem Understanding
  + Specify the business problem
  + Business requirements
  + Literature Survey
  + Social or Business Impact.
* Data Collection & Preparation
  + Collect the dataset
  + Data Preparation
* Exploratory Data Analysis
  + Descriptive statistical
  + Visual Analysis
* Model Building
  + Training the model in multiple algorithms
  + Testing the model
* Performance Testing & Hyperparameter Tuning
  + Testing model with multiple evaluation metrics
  + Comparing model accuracy before & after applying hyperparameter tuning
* Model Deployment
  + Save the best model
  + Integrate with Web Framework
* Project Demonstration & Documentation
  + Record explanation Video for project end to end solution
  + Project Documentation-Step by step project development procedure

**Project Structure:**

Create the Project folder which contains files as shown below



* We are building a flask application which needs HTML pages stored in the templates folder and a python script app.py for scripting.
* model.h5 is our saved model. Further we will use this model for flask integration.
* Training folder contains a model training file.

### Specify The Business Problem

University admission is the process by which students are selected to attend a college or university. The process typically involves several steps, including submitting an application, taking entrance exams, and participating in interviews or other evaluations.

Students are often worried about their chances of admission in University. the university admission process for students can be demanding, but by being well-informed, prepared, and organized, students can increase their chances of being admitted to the university of their choice.

The aim of this project is to help students in short listing universities with their profiles. Machine learning algorithms are then used to train a model on this data, which can be used to predict the chances of future applicants being admitted. With this project, students can make more informed decisions about which universities to apply to, and universities can make more efficient use of their resources by focusing on the most promising applicants.The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

### Business Requirements

The business requirements for a machine learning model to predict chances of student admission in the university. A project aims to predict the chances of a student getting admitted to a particular university based on certain factors The business value of this project is that it will help students make more informed decisions about which universities to apply to, and help university counselors to better advise students on the universities they are most likely to be admitted to the university.

**Literature Survey**

The University Chances of Admission project is a well-researched topic in the field of education and machine learning. Many studies have been conducted to predict university admission using different machine learning techniques.One study by (Hsu and Chen, 2019) used decision tree, random forest, and logistic regression algorithms to predict the chance of university admission based on students' GPA, test scores, and personal information. The study found that the random forest algorithm performed the best with an accuracy of 85.5%.Another study by (Al-Shammari et al., 2018) used the k-nearest neighbor (KNN) algorithm to predict the chance of university admission based on students' GPA, test scores, and family income. The study found that the KNN algorithm performed well with an accuracy of 81.2%.A study by (Najafabadi et al., 2015) used a neural network to predict the chance of university admission based on students' GPA, test scores, and personal information. The study found that the neural network performed well with an accuracy of 94.3%..Overall, these studies suggest that various machine learning algorithms can be used to predict the chance of university admission with high accuracy.

**Social Or Business Impact**

Social Impact:- The ability to accurately predict the chances of university admission can help students make more informed decisions about which universities to apply to, increasing their chances of being admitted and ultimately gaining access to higher education.

Business Model/Impact:- 1. using machine learning models to predict university admission, the service can help universities more efficiently process and evaluate applications, potentially increasing the number of successful admissions.

2.An increase in the number of successful admissions can lead to an increase in revenue for universities, as well as for the company providing the prediction service.

**Data Collection & Preparation**

ML depends heavily on data. It is the most crucial aspect that makes algorithm training possible. So this section allows you to download the required dataset.

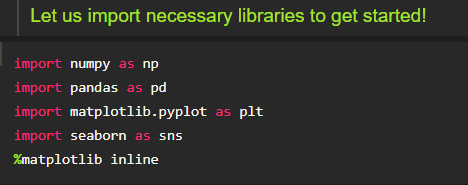
### Collect The Dataset

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc

In this project we have used .csv data. This data is downloaded from kaggle.com. Please refer to the link given below to download the dataset.

**Importing The Libraries**

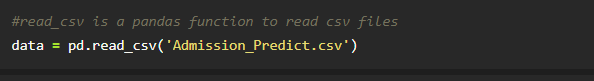
Import the necessary libraries as shown in the image. (optional) Here we have used visualisation style as fivethirtyeight.



**Read The Dataset**

Our dataset format might be in .csv, excel files, .txt, .json, etc. We can read the dataset with the help of pandas.

In pandas we have a function called read\_csv() to read the dataset. As a parameter we have to give the directory of the csv file.



**Data Preparation**

As we have understood how the data is, let's pre-process the collected data.

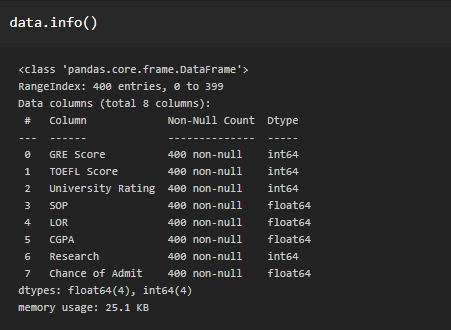
The download data set is not suitable for training the machine learning model as it might have so much randomness so we need to clean the dataset properly in order to fetch good results. This activity includes the following steps.

* Handling missing values
* Handling categorical data
* Handling Imbalance Data

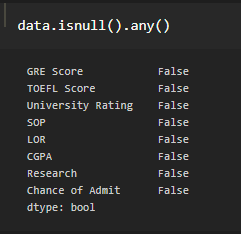
Note: These are the general steps of pre-processing the data before using it for machine learning. Depending on the condition of your dataset, you may or may not have to go through all these steps.

**Handling Missing Values**

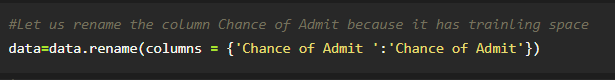
Let’s find the shape of our dataset first. To find the shape of our data, the df.shape() method is used. To find the data type, df.info() function is used.



For checking the null values, df.isnull() function is used. To sum those null values we use .sum() function. From the below image we found that there are no null values present in our dataset.



Let us rename the column, in python have a inbuilt function rename( ). We  can easily rename the column names.



**Exploratory Data Analysis**

In this milestone, we will see exploratory data analysis

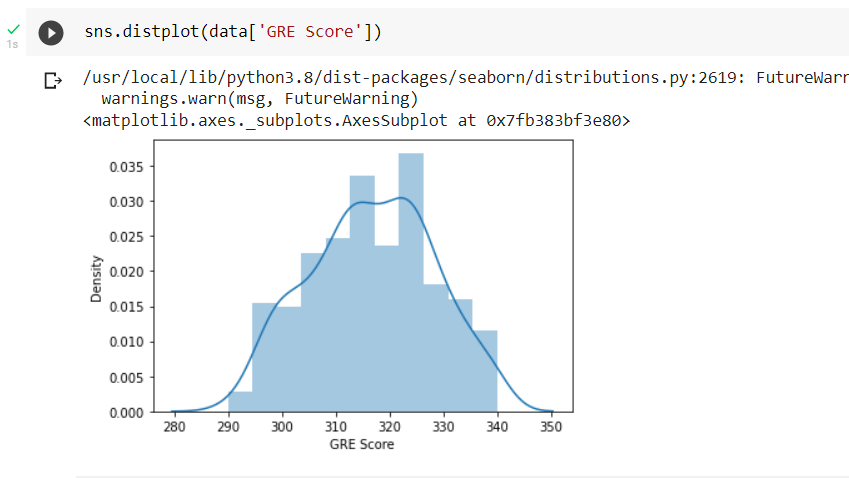
**Visual Analysis**

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

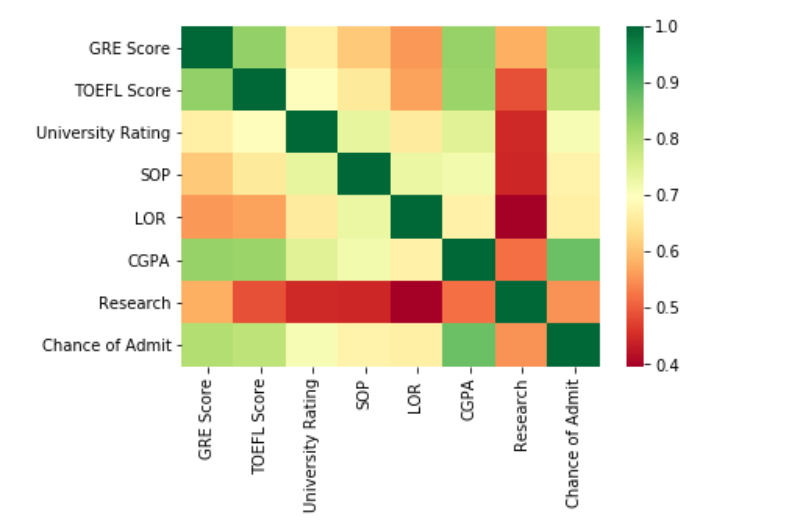
**Univariate Analysis**

In simple words, univariate analysis is understanding the data with a single feature. Here we have displayed two different graphs such as distplot and countplot.

* The Seaborn package provides a wonderful function distplot. With the help of distplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot.



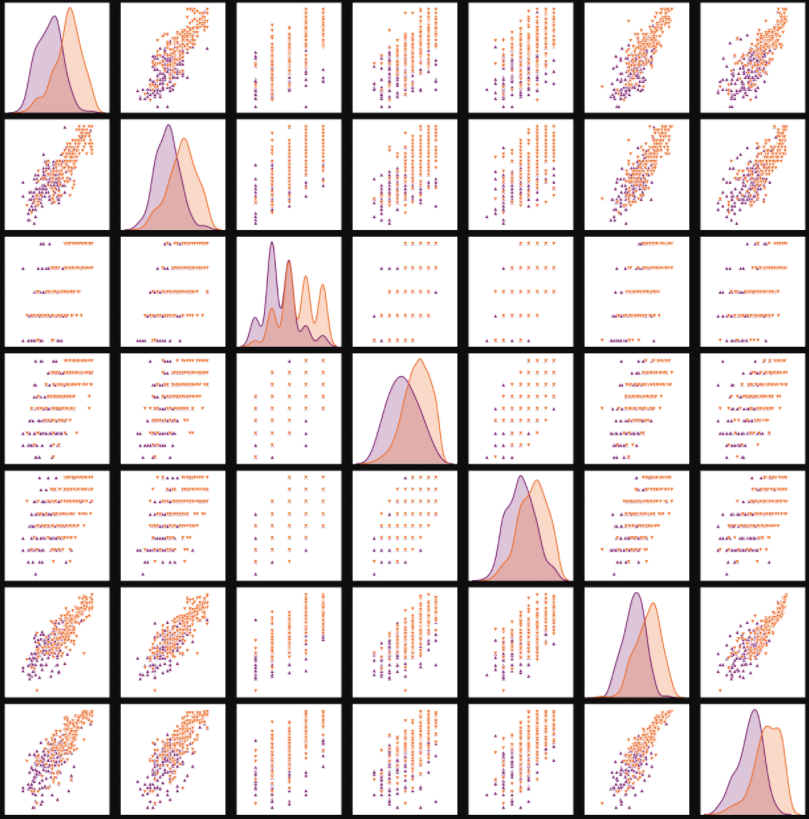
**Bivariate Analysis**



We see that the output variable "Chance of Admit" depends on CGPA, GRE, TOEFEL. The columns SOP, LOR and Research have less impact on university admission.

**Pair Plot**: Plot pairwise relationships in a dataset

https://lh4.googleusercontent.com/Bq8f-YTqlLS1fZl1ERD89gGfbEHbLetJpGTNbc2zPAi0VmkU8kCs4QZeLyIjIrctY_jwBhq2lZlapblxdL_RwPsYRX_PyCjjvc_HIooj-cI5cfcPE-_Z7WKrN8_20dRU3IxEVMWWCNHn6EW99QprxD3ARvxXVmWaejScjtT_nutzB5NVx4K_pNKs04dUvV5aFAgj3TLPmA



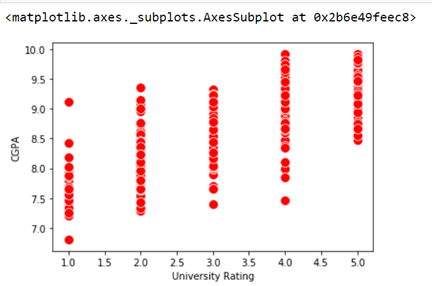
Pair plot usually gives pair wise relationships of the columns in the dataset

1.GRE score TOEFL score and CGPA all are linearly related to each other

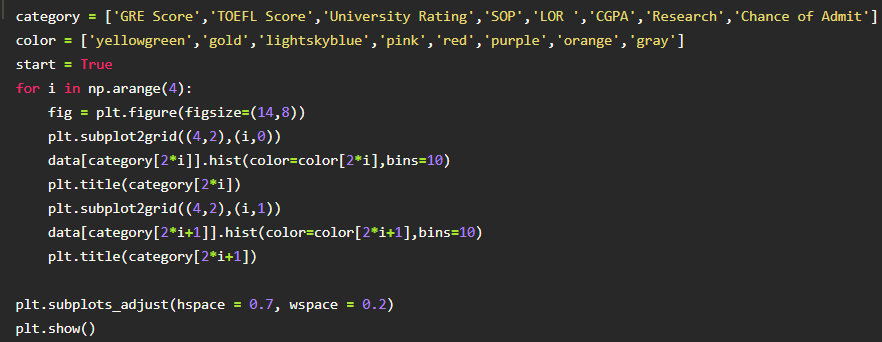
2. Students in research score high in TOEFL and GRE compared to non research candidates

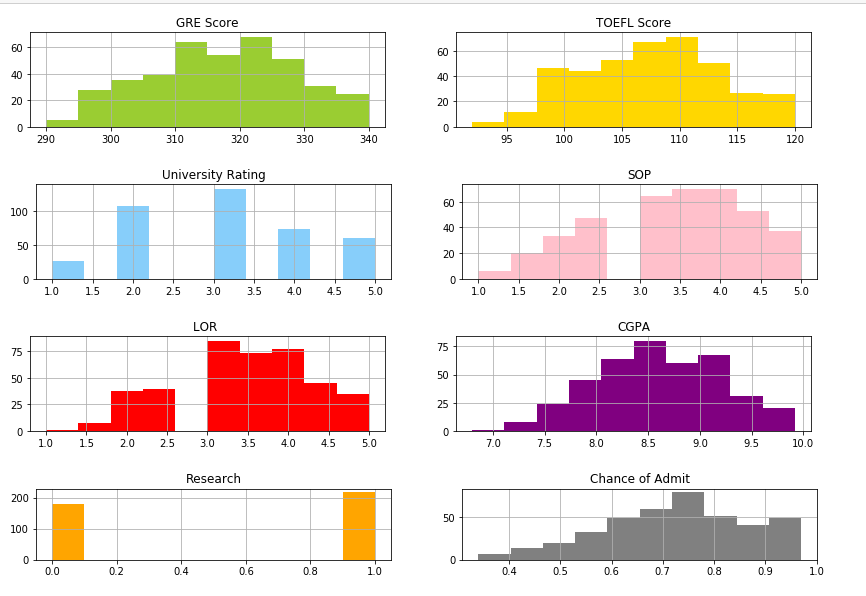
**Scatter Plot**: Matplot has a built-in function to create scatterplots called scatter(). A scatter plot is a type of plot that shows the data as a collection of points

https://lh5.googleusercontent.com/bf_sDnILd3us9zcZtifUnfzsVnCioa9Fi1ynNNj9E5wHKWSJPJJIiOplKuiS1K6FJuwTDaAq-3PVFm1z1TgJvwI2ca3nvNtxkkCC2VVNwaFlpD7rNHhkOirpQh9cHO9CkfdoFUkTMtUISay9W_BJCPlbr5KmiYetMAJ_oi4fdGNIBAOegm28mBopBydpevHenkKDjRodpw



Visualizing the Each column in a dataset using subplot( ).

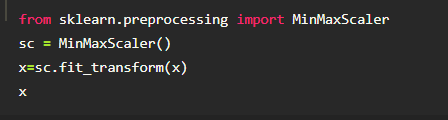




**Scaling the Data**

Scaling is one the important process, we have to perform on the dataset, because of data measures in different ranges can leads to mislead in prediction

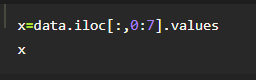
Models such as KNN, Logistic regression need scaled data, as they follow distance based method and Gradient Descent concept.

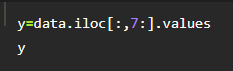


We will perform scaling only on the input values.Once the dataset is scaled, it will be converted into an array and we need to convert it back to a dataframe.

**Splitting data into x and y**

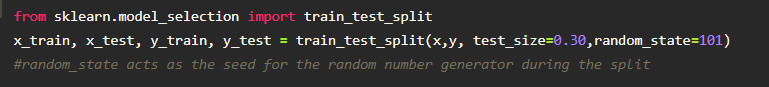
Now let’s split the Dataset into x and y





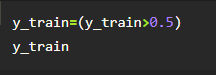
Changes: first split the dataset into x and y and then split the data set

Here x and y variables are created. On x variable, df is passed with dropping the target variable. And on y target variable is passed. For splitting training and testing data we are using the train\_test\_split() function from sklearn. As parameters, we are passing x, y, test\_size, random\_state.



**Let us convert it into classification problem**

chance of admit>0.5 as true chance of admit<0.5 as false



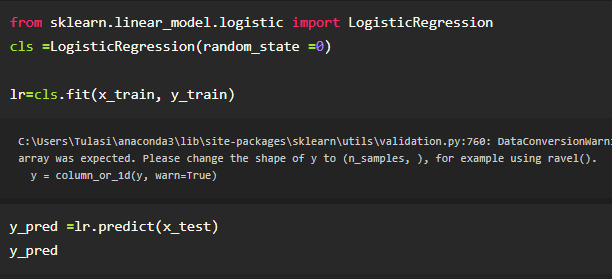
https://lh5.googleusercontent.com/eAnl6HBvgLfdxI8zXDHBz5NWDzUBomq_Px2vbuiivCpoVslxZcmzc7jUJkanU90vQi5ga-S1ZbJ1_coPdlc9kFCnAtaqSHkv4MAwXPnTThpSsCK_NffdmVv6d_iDHPO1xnul6IFRCcKbCxnZQpOM44MOCYGgwBYeyAHBVwwQZF1YpPDZUP5xIbZ9DyAT545N1I15rhQPMA

**Training The Model In Multiple Algorithms**

Now our data is cleaned and it’s time to build the model. We can train our data on different algorithms. For this project we are applying four  classification algorithms. The best model is saved based on its performance.

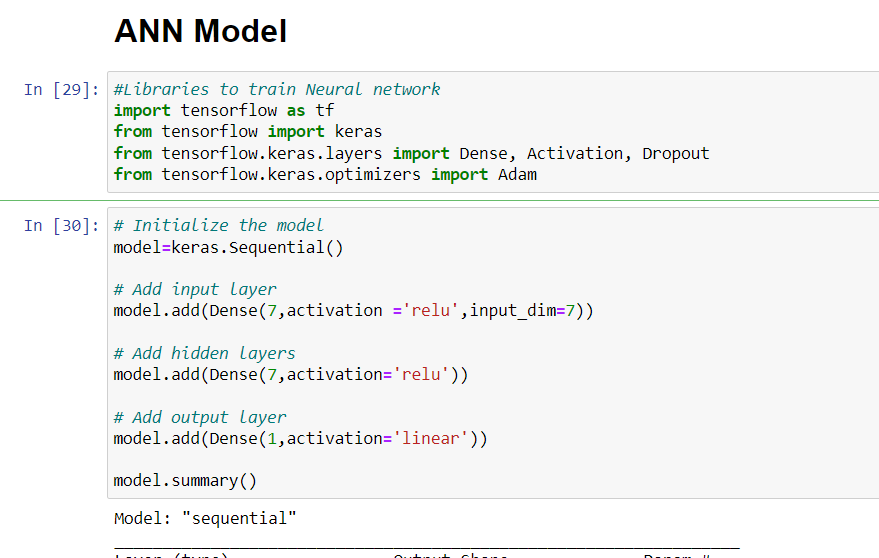
**Logistic Regression Model**

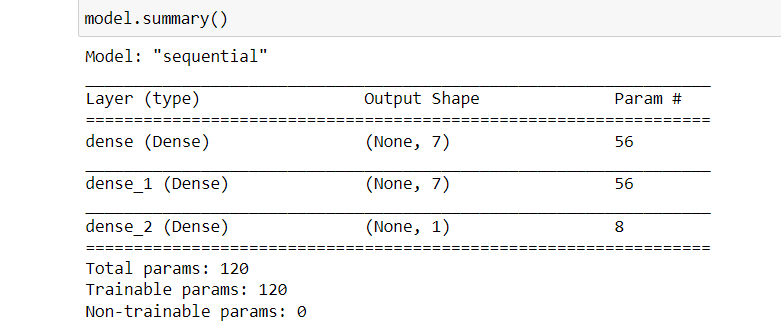
A LogisticRegression algorithm is initialised and training data is passed to the model with the .fit() function. Test data is predicted with .predict() function and saved in a new variable. For evaluating the model, a confusion matrix and classification report is done.

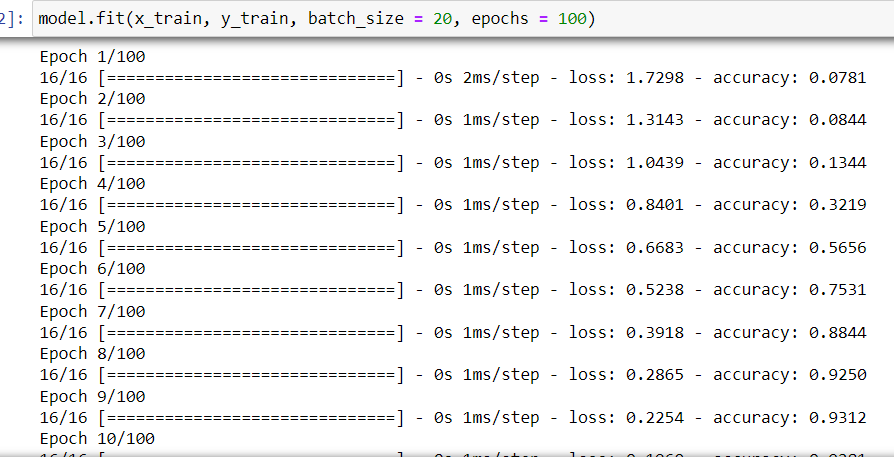


**ANN Model**

Building and training an Artificial Neural Network (ANN) using the Keras library with TensorFlow as the backend. The ANN is initialised as an instance of the Sequential class, which is a linear stack of layers. Then, the input layer and two hidden layers are added to the model using the Dense class, where the number of units and activation function are specified. The output layer is also added using the Dense class with a sigmoid activation function. The model is then compiled with the Adam optimizer, binary cross-entropy loss function, and accuracy metric. Finally, the model is fit to the training data with a batch size of 100, 20% validation split, and 100 epochs.







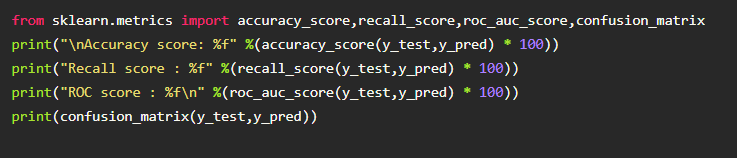
**Testing Model With Multiple Evaluation Metrics**

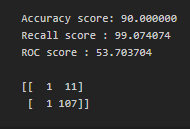
Multiple evaluation metrics means evaluating the model's performance on a test set using different performance measures. This can provide a more comprehensive understanding of the model's strengths and weaknesses. We are using evaluation metrics for classification tasks including accuracy, precision, recall, support and F1-score.

**Compare The Model**

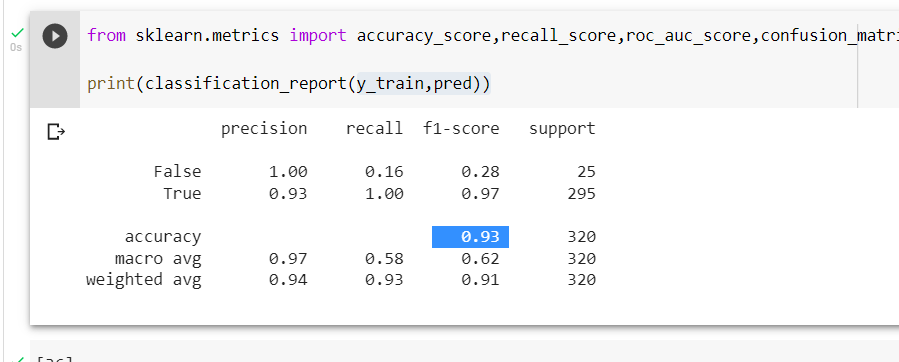
For comparing the above four models, the compareModel function is defined.

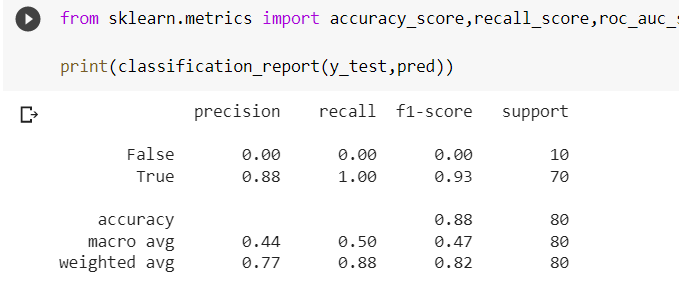
**Logistics Regression model**

****

****

**ANN Model** : Training Accuracy





the results of models are displayed as output. From the both models ANN is performing well. From the below image, We can see the accuracy of the model. ANN is giving the accuracy of 93.% with training data , 88% accuracy for the testing data.

**Model Deployment**

In this Milestone, We will see the model deployment.

**Save The Best Model**

Saving the best model after comparing its performance using different evaluation metrics means selecting the model with the highest performance and saving its weights and configuration. This can be useful in avoiding the need to retrain the model every time it is needed and also to be able to use it in the future.



**Integrate With Web Framework**

In this section, we will be building a web application that is integrated to the model we built. A UI is provided for the uses where he has to enter the values for predictions. The enter values are given to the saved model and prediction is showcased on the UI.

This section has the following tasks

* Building HTML Pages
* Building server side script
* Run the web application

**Building Html Pages**

For this project create two HTML files namely

* home.html
* predict.html

and save them in the templates folder.