INTELLIGENT ADMISSION :
THE FUTURE OF UNIVERSITY DECISION MAKING WITH MACHINE LEARNING.

TEAM MEMBERS NAME : R.ASATH
S.JANARTHANAM
A.JAYAKANTHAN
V.YOGANATHAN

CLASS : III-BSC COMPUTER SCIENCE

TABLE OF INDEX

1. INTRODUCTION

1.1 Overview

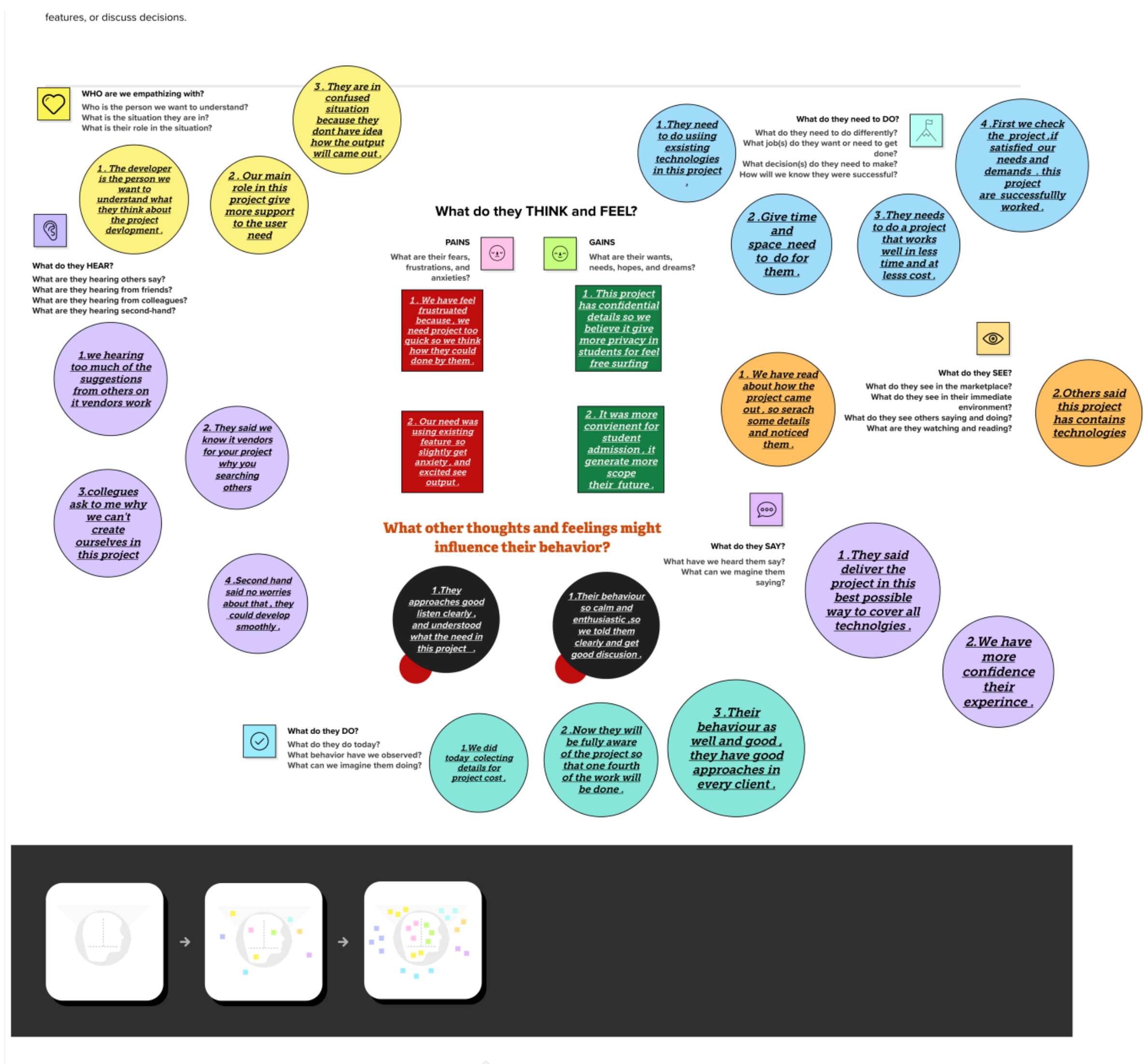
As there is advancement in Machine Learning, it is easy to develop an application accurately. There are various types of applications available for knowing whether we get seat in USA or not but mostly these are not reliable and not much effective and also building such applications is difficult but, Machine Learning provides us with several algorithms that are helpful to build a representation easily. The purpose of this work is to compare different Machine Learning algorithms and find out which algorithm is giving an accurate result for the available dataset. The algorithms we are going to use are Multi Linear Regression, Polynomial Regression and Random Forest. The input for these algorithms is GRE score, TOFEL score and CGPA of candidate. By using dataset we are going to train the representation and finally the output we are obtaining is the percentage of chance to get seat in reputed university.

1.2 Purpose

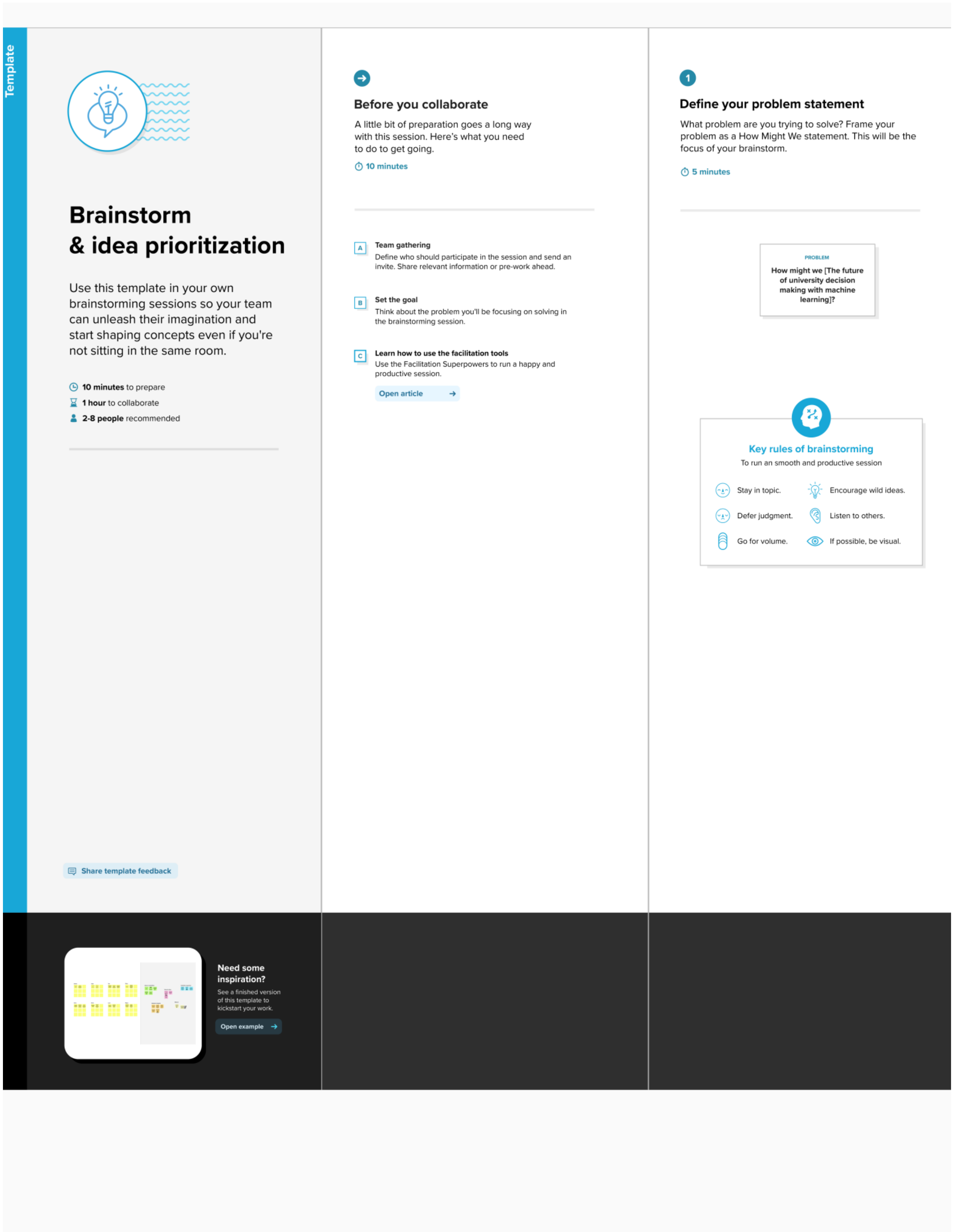
Graduate admission prediction is a useful tool for students who are interested in pursuing higher education at the graduate level. It can help them to determine their likelihood of being accepted into a particular graduate program based on their academic background and other relevant factors. This information can be valuable for students who are considering applying to multiple programs and want to prioritize their efforts. Additionally, graduate admission prediction can provide students with a better understanding of the admissions process and the factors that are considered by admissions committees, which can help them to prepare more effectively for the application process. Overall, graduate admission prediction can be a helpful resource for students who are planning to pursue graduate education.

2. PROBLEM DEFINITION & DESIGN THINKING

2.1 Empathy Map:



2.2 Ideation & Brainstorming Map :



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

JANARTHANAM S

- We want assure data safety .
- Know the back end process how it works in this application .
- We want this software reliable
- Create data set for making prototype modelling .

ASATH R

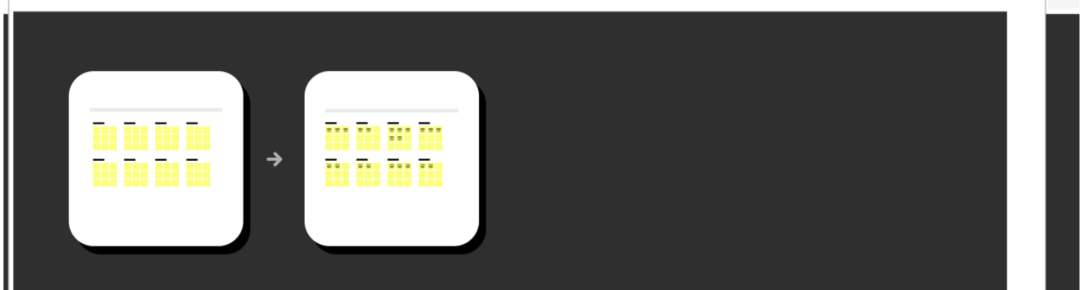
- Surfing online fors get ideas how to design user friendly interface
- We provide smooth interface .
- Creating database for permanent access .
- We should make a blue print .

YOGANATHAN V

- Research about similar project for gaining knowledge .
- In this we using the gradient colours .
- We want to allows site informations anywhere .
- We need to create project more simplified and accessibe .
- Sanctioned amount is too less so that we could not insert the new features .

JAYAKANTHAN A

- We make graph presentation to understand easily .
- We have concentrate more on students cookies details for servic
- The is project only contains genuine details .
- Provide highest performance

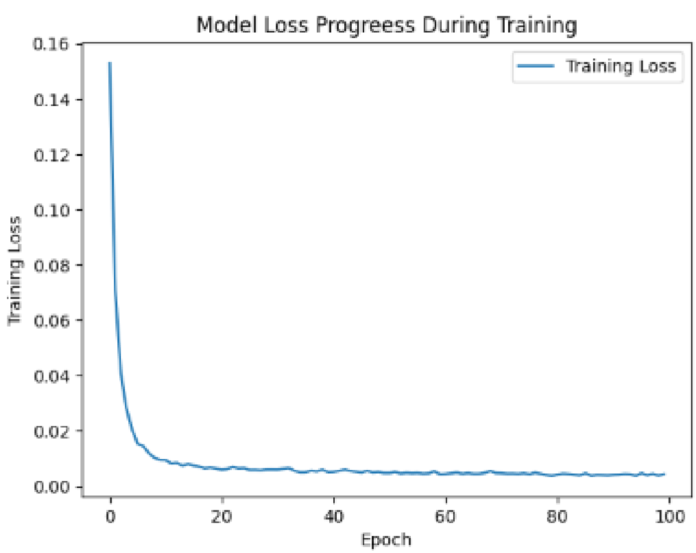


ll be on the same page about what's important
ace your ideas on this grid to determine which
t and which are feasible.
TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.
first we should create blueprint and step by step procedure
Get knowledge and suggestions from senior attuoritizes.
Get knowledge about the projects and collect resources.
Is this project has more confidential student detail so we have put on more security system.
We have organized team meeting in weekend
We does not have any neccecseity to include need more optins on ths site.
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
After you collaborate
You can export the mural as an image or pdf to share with members of your company who might find it helpful.
Quick add-ons
Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.
Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.
Keep moving forward
Strategy blueprint
Define the components of a new idea or strategy.
Open the template →
Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →
Share template feedback

3. RESULT

Statistical Test According to the normality test, the dependent variable is not normally distributed. Therefore, nonparametric test will be performed using PHStat. The test is one-way ANOVA which is performed to determine whether three samples or more have any statistically significant differences between their means or not [23]. The test shows that p-value equals 0.97, which is greater than 0.05, thus, the null hypothesis cannot be rejected, and the tests are not statistically different. B. Mean absolute error The different regression models are performed on Admission dataset through Weka in order to decide which model performs the best based on mean absolute error (MAE) value. The result are shown in table:

MODEL NAME	**ACCURACY RESULT**
Linear Regressor	0.89
Decision Tree	0.55
Random Forest	0.85
ANN Model	0.99



Checking the Score of Regressors

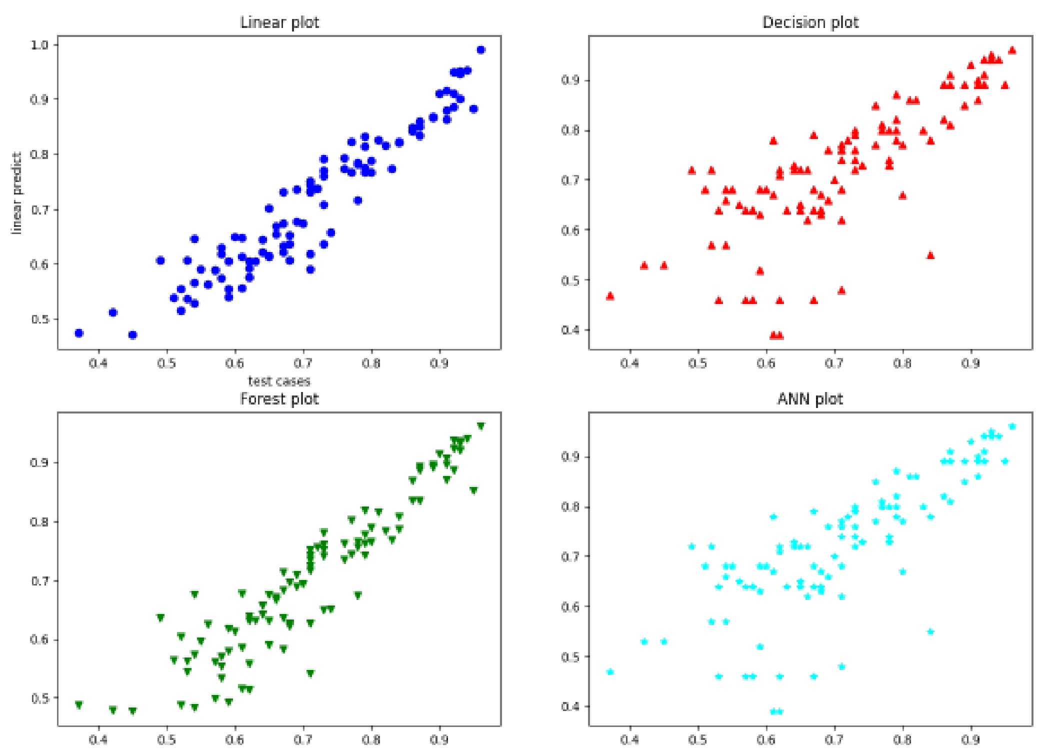
Liner Accuracy : 0.8982869098533859

Decision Accuracy : 0.5595241182894614

Forest Accuracy : 0.8597504374683729

4/4 [==============================] - 0s 4ms/step - loss: 0.0043
ANN Accuracy : 0.9957494358532131

Plotting the plots



4. ADVANTAGES & DISADVANTAGES

Advantages:

- Predictive analytics can provide a more accurate assessment of a student's likelihood of being accepted into a graduate program, based on a range of relevant factors such as academic performance, work experience, and test scores.
- It can help students to identify the programs that are most likely to accept them, which can save time and effort in the application process.
- Predictive analytics can provide insights into the admissions process, including the factors that are most important to admissions committees, which can help students to prepare more effectively for the application process.
- It can help universities to identify potential candidates who may have been overlooked in the traditional application process, which can lead to a more diverse and talented student body.

Disadvantages:

- Predictive analytics may not take into account the unique aspects of each individual applicant, such as their personal statement, letters of recommendation, or other factors that could impact their likelihood of being accepted.
- The use of predictive analytics in admissions could raise concerns about fairness and bias, particularly if the algorithms used are not transparent or if they are based on factors that could disproportionately impact certain groups of students.
- Students may become overly reliant on predictive analytics and miss out on opportunities to apply to programs that may be a good fit for them, but not necessarily predicted by the algorithm.
- Predictive analytics can be expensive to implement and maintain, which could limit access to this technology for smaller universities or institutions with limited resources.

5. APPLICATION

- Universities and Colleges: Universities and colleges can use predictive analytics to identify prospective students who are a good fit for their graduate programs. This can help to streamline the admissions process and ensure that the most qualified applicants are accepted into the program.

- Educational Consultancies: Educational consultancies can use graduate admission prediction methods to help students identify the graduate programs that are most likely to accept them based on their academic background and other relevant factors. This can be a valuable service for students who are considering multiple programs and want to prioritize their efforts.

- Government Agencies: Government agencies can use predictive analytics to identify potential candidates for government-funded graduate programs or scholarships. This can help to ensure that resources are allocated to the most qualified and deserving candidates.

- Non-Profit Organizations: Non-profit organizations can use graduate admission prediction methods to identify candidates for fellowship or grant programs. This can help to ensure that resources are allocated to the most deserving candidates, and that the programs have a positive impact on the communities they serve.

- Overall, graduate admission prediction methods can be used in any setting where there is a need to identify the most qualified candidates for graduate programs or other educational opportunities.

6. CONCLUSION

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. We have developed the machine learning project using python programming language and the reports are shown the above.

7. FUTURE SCOPE

The Future and Scope of graduate admission prediction.

- Graduate admission prediction is a field that has gained a lot of attention in recent years due to the increasing demand for higher education and the growing number of applicants. The use of machine learning algorithms and predictive models has enabled universities and colleges to predict the likelihood of an applicant being accepted into their graduate programs.
- The future of graduate admission prediction looks promising, as more and more institutions are adopting this technology to streamline their admission processes and make data-driven decisions. This will not only save time and resources but also improve the accuracy and fairness of the admission process.
- Additionally, graduate admission prediction has the potential to expand beyond the traditional academic realm. It could be used by employers to predict the success of job candidates based on their educational background, or by financial institutions to assess the creditworthiness of loan applicants based on their educational qualifications.
- In conclusion, graduate admission prediction is a rapidly growing field with a lot of potential for future applications. As more institutions adopt this technology, we can expect to see more accurate and efficient admission processes, as well as new applications in other industries.

8. APPENDIX

Importing Libraries and the Dataset

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import seaborn as sns # data visualization library

import matplotlib.pyplot as plt # mathematical plotting library


#read the dataset

"""import os

for dirname, _, filenames in os.walk('/kaggle/input'):

    for filename in filenames:

        print(os.path.join(dirname, filename));"""


admission_df = pd.read_csv('/content/Admission_Predict_Ver1.1 (1).csv',index_col="Serial No.")

admission_df
```

Perform Exploratory data analysis

```
admission_df.info()

admission_df.nunique()

admission_df.columns

admission_df.describe()

df_univ = admission_df.groupby(by = 'University Rating').mean()

df_univ
```

Perform data visualization

```
admission_df.hist(bins=10, figsize=(20,15))

plt.show()

fig, ax = plt.subplots(figsize=(10,10))

sns.heatmap(admission_df.corr(), annot=True, cmap='Blues')

sns.pairplot(data=admission_df,hue='Research',markers=["^","v"],palette='inferno')

#correlatoin pair plots

sns.pairplot(admission_df.drop(columns=["LOR ","SOP","Research"]));

sns.scatterplot(data=admission_df, x="University Rating",y="CGPA" ,color='Red',s=100)

#sns.jointplot(x="CGPA",y="Chance of Admit ",data=admission_df);
```

High CGPA increases the chance of admission

```
num = [304.91176471, 309.13492063, 315.0308642 , 323.3047619 ,
       327.89041096]

gure(fiplt.figsize=(10,8))

sns.barplot(y="GRE Score",x="University Rating",data=admission_df)

plt.ylim([300,330])

li = 0.1

for i in range(5):

    plt.text(li , num[i]+0.5, np.round(num[i],2) )

    li+=1

plt.title("Expected GRE score vs University Rating");


#sns.boxplot(x="LOR ",y="Chance of Admit ",data=admission_df)

#plt.title("Chance of admission depending on Letter of Recommendation");
```

LOR at 1.5 and 4.5 had an outlier, i.e. the chance of admission of a candidate is higher with low LOR, similarily at 4.5 the chance of admission become less.

```
sns.boxplot(x="SOP",y="Chance of Admit ",data=admission_df)

plt.title("Chance of admission depending on Letter of Recommendation");

plt.figure(figsize=(12,8))

sns.lineplot(x="SOP",y="Chance of Admit ",data=admission_df, label="SOP")

sns.lineplot(x="LOR ",y="Chance of Admit ",data=admission_df, label="LOR")

sns.lineplot(x="University Rating",y="Chance of Admit ",data=admission_df, label="Research")

plt.legend()

plt.title("features affecting admission on Scale of 0-5")

plt.xlabel("Features Scale")

plt.show()

category=admission_df.columns

color=['Red','Pink','Orange','Yellow','Purple','Green','Blue','Brown']

start=True

for i in np.arange(4):

  fig=plt.figure(figsize=(14,8))

  plt.subplot2grid((4,2),(i,0))

  admission_df[category[2*i]].hist(color=color[2*i],bins=10)

  plt.title(category[2*i])

  plt.subplot2grid((4,2),(i,1))

  admission_df[category[2*i+1]].hist(color=color[2*i+1],bins=10)

  plt.title(category[2*i+1])

plt.subplots_adjust(hspace=0.7,wspace=0.2)
```

```
plt.show()
```

Create training and testing dataset

```
#create the dependent and independent dataset

#splitting training and test set

#test set is last 100 observations


X_train=admission_df.iloc[0:400,:-1].values

y_train= admission_df.iloc[0:400,-1].values

X_test=admission_df.iloc[400:500,:-1].values

y_test= admission_df.iloc[400:500,-1].values


#feature scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_train = sc.fit_transform(X_train)

X_test = sc.transform(X_test)


GRE=[]

TOEFL=[]

for i in range(X_train.shape[0]):

    GRE.append(X_train[i][1])

    TOEFL.append(X_train[i][0])

sns.kdeplot(GRE, shade=True, label="GRE")

sns.kdeplot(TOEFL, shade=True, label="TOEFL")

plt.title("Density chart of GRE vs TOEFL"
```

Train and evaluate model

Linear regression

```
from sklearn.linear_model import LinearRegression

linear_reg = LinearRegression()

linear_reg.fit(X_train, y_train)

y_lin_pred = linear_reg.predict(X_test)
```

Decision Tress and Random Forest Models

```
from sklearn.tree import DecisionTreeRegressor

Decision_regressor = DecisionTreeRegressor(random_state = 0)

Decision_regressor.fit(X_train, y_train)

y_decision_pred = Decision_regressor.predict(X_test)

from sklearn.ensemble import RandomForestRegressor

Forest_regressor = RandomForestRegressor(n_estimators = 100, random_state = 0)

Forest_regressor.fit(X_train, y_train)

y_forest_pred = Forest_regressor.predict(X_test)

regr = LinearRegression()

#x_train,x_test,y_train,y_test

rg=regr.fit(X_train,y_train)

y_pred =rg.predict(X_test)

y_pred
```

Train and evaluate an Artificial Neural Network (ANN)

```
#Libraries to train Neural network

import tensorflow as tf
```

```
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam

ANN_model = keras.Sequential()
ANN_model.add(Dense(50, input_dim=7))
ANN_model.add(Activation('relu'))

ANN_model.add(Dense(150))
ANN_model.add(Activation('relu'))
ANN_model.add(Dropout(0.5))

ANN_model.add(Dense(150))
ANN_model.add(Activation('relu'))
ANN_model.add(Dropout(0.5))

ANN_model.add(Dense(50))
ANN_model.add(Activation('linear'))
ANN_model.add(Dense(1))

ANN_model.compile(loss = 'mse', optimizer = 'adam')
ANN_model.summary()

#Using Adam optimizer
ANN_model.compile(optimizer = 'Adam', loss = 'mean_squared_error'
```

```
epochs_hist = ANN_model.fit(X_train, y_train, epochs = 100, batch_size = 20);
y_ann_pred = ANN_model.predict(X_test)
result = ANN_model.evaluate(X_test, y_test)
epochs_hist.history.keys()
plt.plot(epochs_hist.history['loss'])
plt.title('Model Loss Progreess During Training')
plt.xlabel('Epoch')
plt.ylabel('Training Loss')
plt.legend(['Training Loss'])
pred =ANN_model.predict(X_test)
pred = (pred>0.5)
pred
```

Checking the Score of Regressors

```
from sklearn.metrics import accuracy_score
acc_lin = linear_reg.score(X_test, y_test)
print("Liner Accuracy : {}".format(acc_lin))
acc_decision = Decision_regressor.score(X_test, y_test)
print("Decision Accuracy : {}".format(acc_decision))
acc_forest = Forest_regressor.score(X_test, y_test)
print("Forest Accuracy : {}".format(acc_forest))
acc_ANN = 1 - ANN_model.evaluate(X_test, y_test)
print("ANN Accuracy : {}".format(acc_ANN))
```

Plotting the plots

```
plt.figure(figsize= (14,10))

#y_test on x axis

#y_pred on y axis

plt.subplot(221)

plt.plot(y_test, y_lin_pred,'o', color = 'b')

plt.title('Linear plot')

plt.ylabel("linear predict")

plt.xlabel("test cases")


plt.subplot(222)

plt.plot(y_test, y_decision_pred, '^', color = 'r')

plt.title('Decision plot')


plt.subplot(223)

plt.plot(y_test, y_forest_pred, 'v', color = 'g')

plt.title('Forest plot')


plt.subplot(224)

plt.plot(y_test, y_decision_pred, '*', color = 'aqua')

plt.title('ANN plot')
```

Calculate Regression Model KPIs

```
k = X_test.shape[1]

n= len(X_test)

k,n

from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

from math import sqrt

r2 = r2_score(y_test, y_lin_pred)

adj_r2 = 1- (1-r2)*(n-1)/(n-k-1)

MAE = mean_absolute_error(y_test, y_lin_pred)

MSE = mean_squared_error(y_test, y_lin_pred)

RMSE = float(format(np.sqrt(mean_squared_error(y_test, y_lin_pred)),'.3f'))

print('R2 - ', r2, '\nAdjusted R2 - ', adj_r2, '\nMAE - ', MAE, '\nMSE - ', MSE, '\nRMSE - ',
RMSE)

from statsmodels.api import OLS

summ=OLS(y_train,X_train).fit()

summ.summary()
```

#Save the model

```
import pickle

pickle.dump(acc_ANN,open('ANN_model.pkl','wb'))
```