Intelligent Admissions: The Future of University Decision Making with Machine Learning

**PROJECT RECORD TEMPLATE**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER** | **TITLE** | | **PAGE.NO.** |
| 1 | INTRODUCTION   * 1. OVERVIEW   2. PURPOSE | |  |
| 2 | PROBLEM DEFINITION & DESIGN THINKING  2.1 EMPATHY MAP  2.2 IDEATION & BRAINSTOMING MAP | |  |
| 3 | RESULT | |  |
| 4 | ADVANTAGES & DISADVANTAGES | |  |
| 5 | APPLICATIONS | |  |
| 6 | CONCLUSION | |  |
| 7 | FUTURE SCOPE |  | |
| 8 | APPENDIX  8.1 SOURCE CODE |  | |

**CHAPTER 1**

**INTRODUCTION**

* 1. OVERVIEW

The project (Intelligent Admission: The future of university decision making with machine learning) to build a machine learning model that can predict In recent years, the use of machine learning in the admissions process for universities has become an increasingly popular topic. With the vast amounts of data that universities collect on applicants, machine learning algorithms have the potential to analyze this data and make more informed decisions about which students to admit. This approach, known as intelligent admissions, has the potential to improve the fairness, efficiency, and effectiveness of the admissions process.

By using machine learning algorithms, universities can identify patterns and trends in data that might not be immediately apparent to human admissions officers. This can help to eliminate biases that might exist in the current admissions process and allow universities to consider a wider range of factors when making decisions about which applicants to admit.

* 1. PURPOSE

The purpose of this project is to develop a machine learning model the purpose of the "Intelligent Admissions: The Future of University Decision Making with Machine Learning" project is to explore the potential benefits of using machine learning algorithms in the admissions process for universities. The project aims to build a machine learning model that can analyze the vast amounts of data collected by universities on applicants and make more informed decisions about which students to admit.

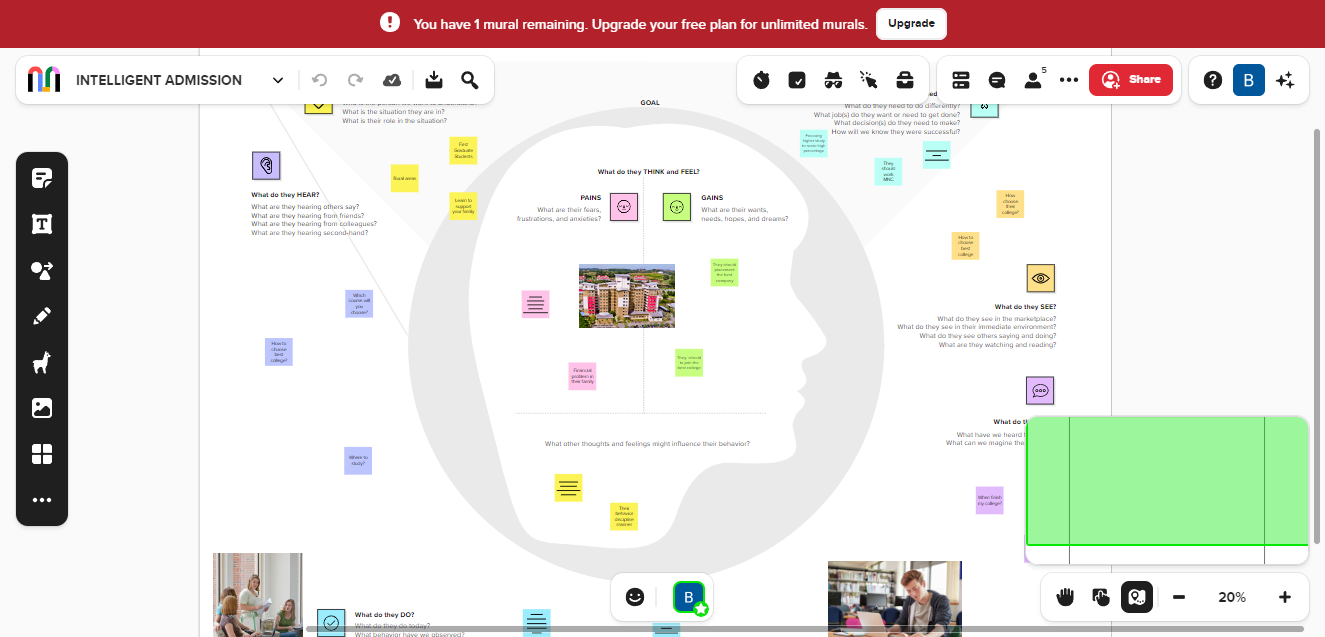
The project seeks to address some of the limitations of the current admissions process by eliminating biases that might exist and allowing universities to consider a wider range of factors when making decisions about admissions. Additionally, the project aims to improve the efficiency and effectiveness of the admissions process by automating certain aspects of the process, reducing the time and resources needed to make admissions decisions.

Overall, the purpose of the project is to explore how machine learning can be used to create a fairer, more efficient, and more effective admissions process for universities.

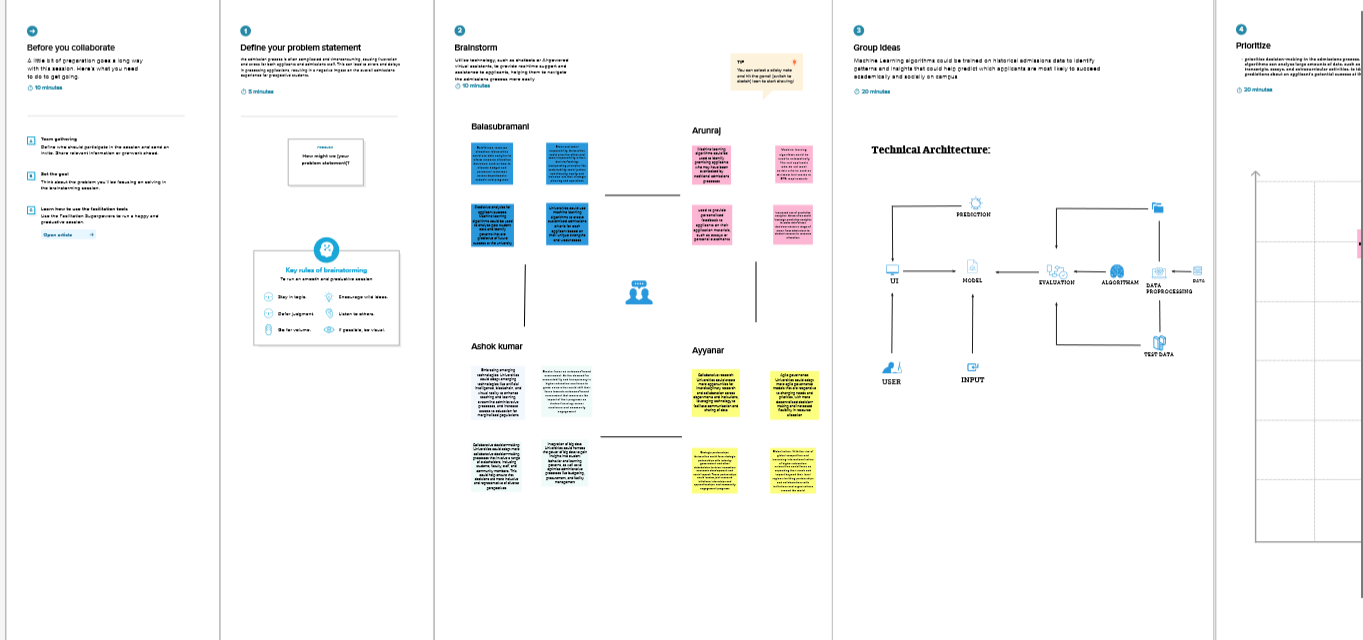
**CHAPTER 2**

**PROBLEM DEFINITION & DESIGN THINKING**

2.1 PROBLEM DEFINITION:



2.2 IDEATION & BRAINSTOMING MAP:



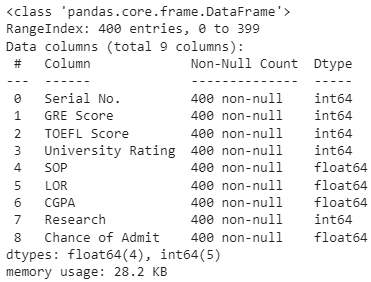
**CHAPTER 3**

**RESULT**

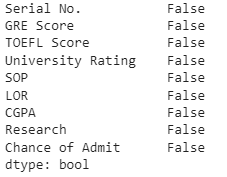
Result 1:

* Import all the tools we need.
* All needed tools import successful.

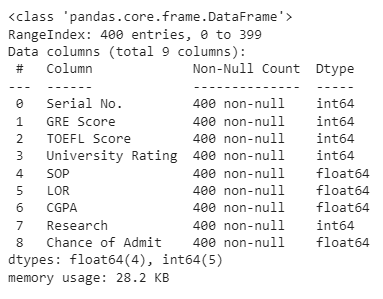
Result 2:



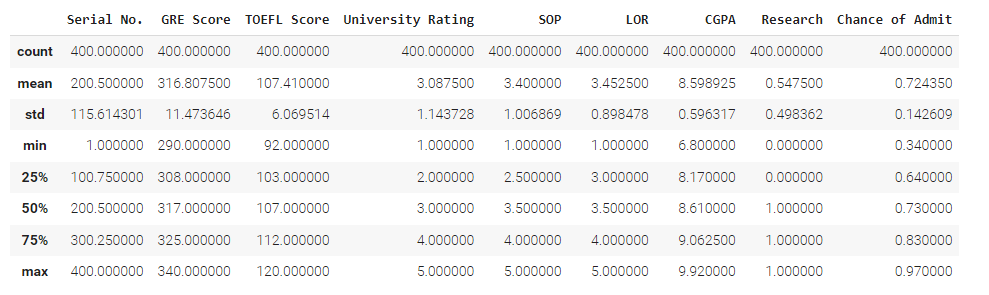
Result 3:



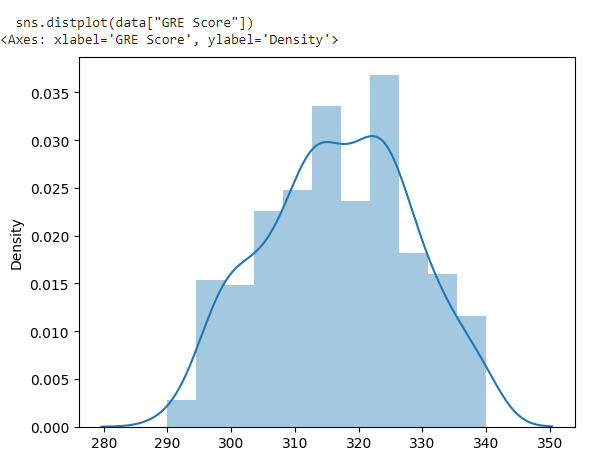
Result 4:



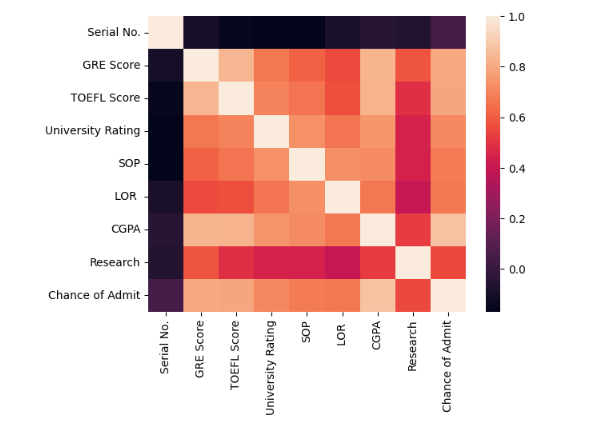
Result 5:



Result 6:



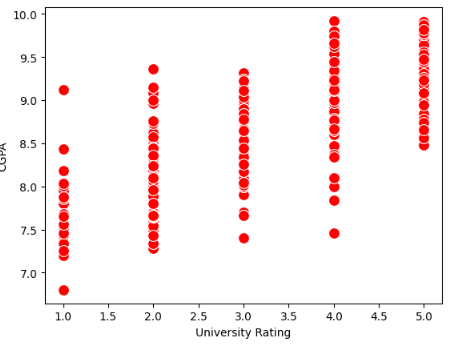
Result 7:



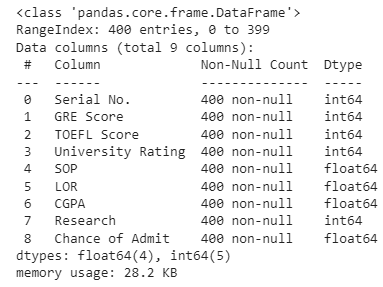
Result 8:



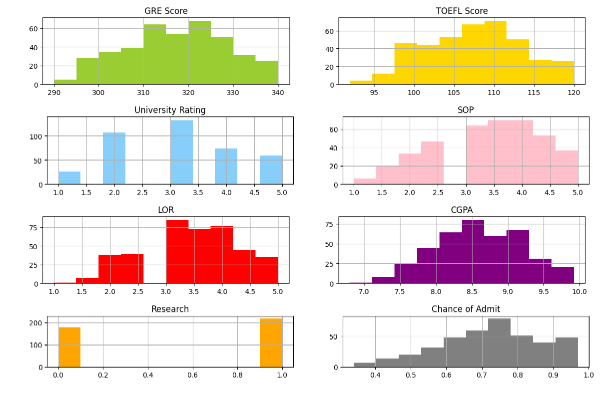
Result 9:



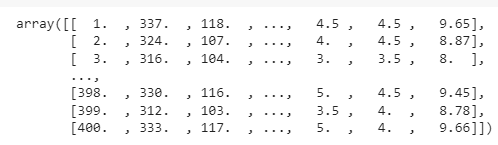
Result 10:



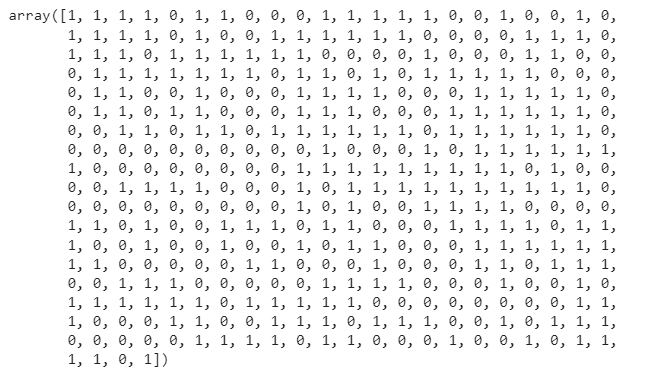
Result 11:



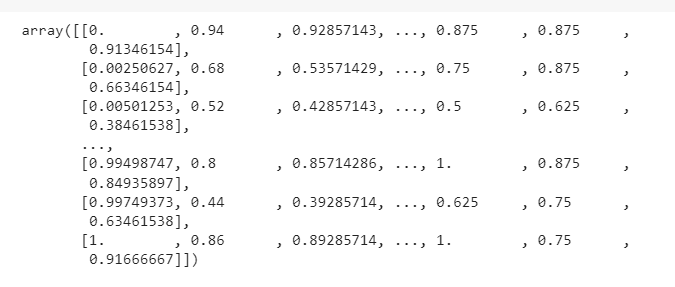
Result 12:



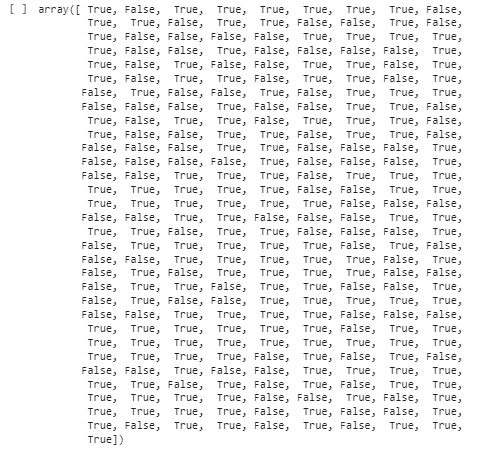
Result 13:



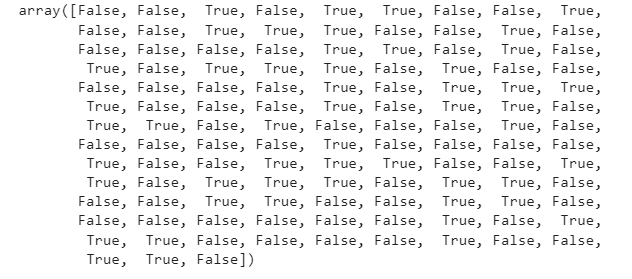
Result 14:



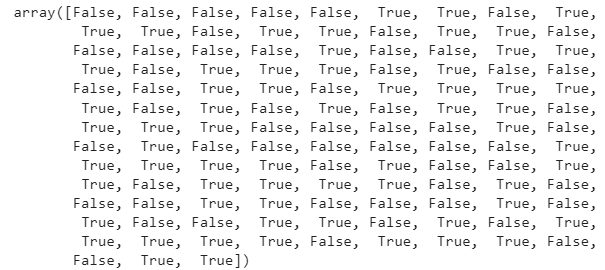
Result 15:



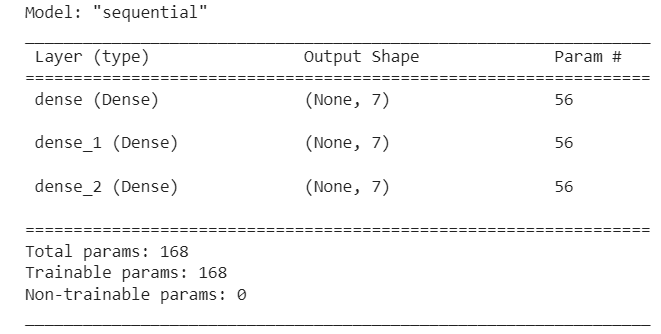
Result 16:



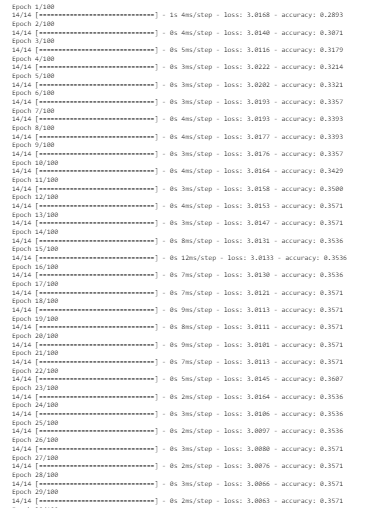
Result 17:



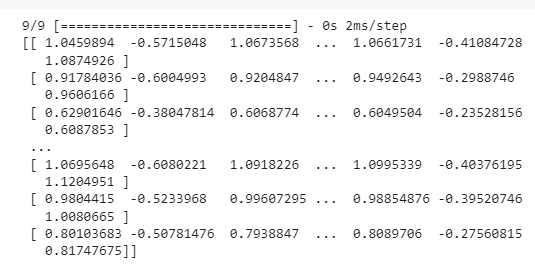
Result 17:



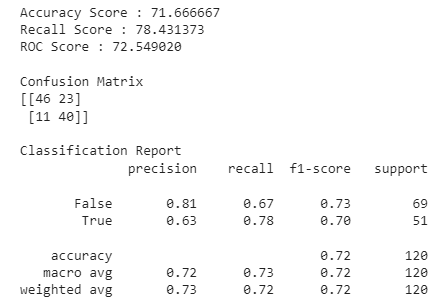
Result 17:



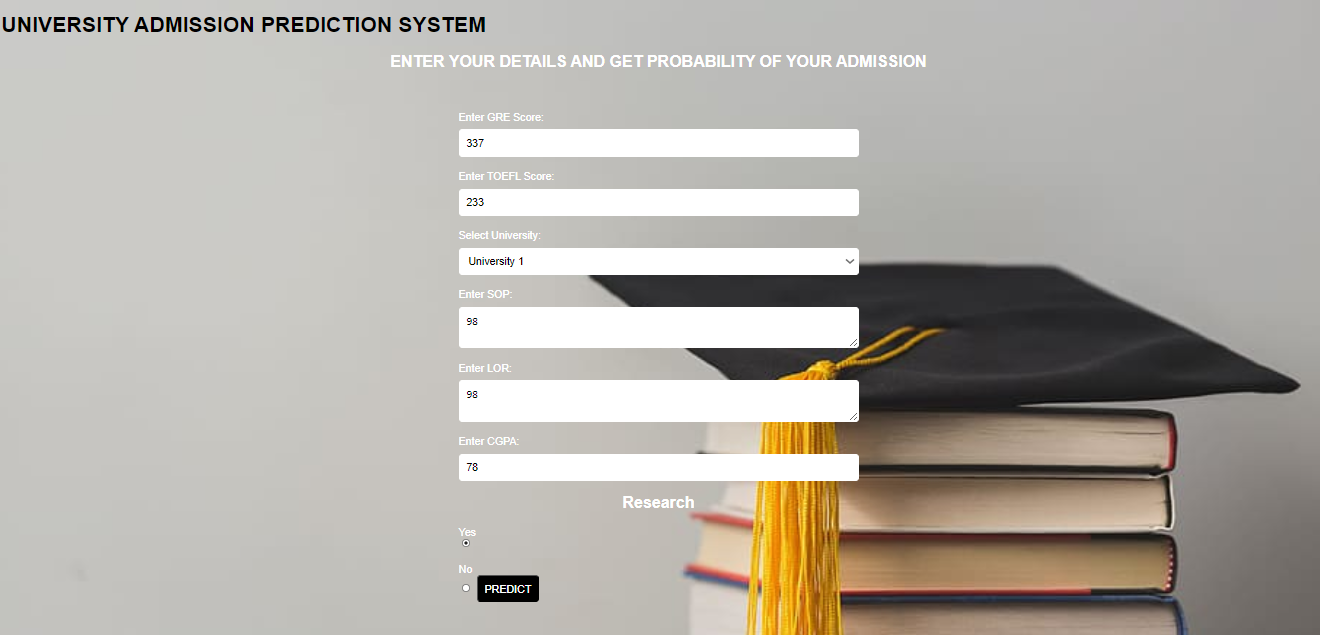
Result 18:



Result 19:



Result 20:



Result 21:



**CHAPTER 4**

**ADVANTAGES & DISADVANTAGES**

ADVANTAGES:

* Efficient and Accurate: Intelligent admission can process large volumes of data quickly and accurately, helping universities to make more informed decisions and reducing of risk of human error.
* Machine learning algorithms can help eliminates biases and discrimination in the admission process, ensuring that candidates are based on their merit rather than their demographic characteristics.
* By admitting the most qualified candidates, University can improve students overcomes, such as graduation employment prospects.

By automating the admission process, University can save time and money on administrative tasks, freeing up resources to invest in other areas

DISADVANTAGES:

* Intelligent admission relies solely on data and algorithms to make admission decision, which can lead to a lack of personal touch and a failure to consider intangible factors that may be important process.
* Machine learning algorithms require large amounts of data to train and make accurate predictions, which may not always be available or accessible for some university.
* Although machine learning algorithms can help eliminate bias and discrimination, they can also perpetuate it if the data used o train them is biased or incomplete.
* Machine learning algorithms can be complex and difficult to understand, which may make it challenging for university to explain admission decisions to students and other stakeholders.

**CHAPTER 5**

**APPLICATIONS**

Machine learning is a subfield of artificial intelligence that involves the use of algorithms and statistical models to analyze data and make predictions. The application of machine learning techniques in university admissions can streamline the process, reduce bias, and improve the accuracy of decision-making.

Here are some potential applications of machine learning in university admissions:

* Predictive modeling: Machine learning algorithms can be used to develop predictive models that analyze various factors such as grades, test scores, extracurricular activities, and personal statements. These models can predict a student's likelihood of success at the university and help admissions officers make informed decisions.
* Bias reduction: Machine learning algorithms can reduce the impact of unconscious bias by removing identifying information such as gender, race, and ethnicity from applications. This can help ensure that admissions decisions are based solely on merit.
* Data analysis: Machine learning algorithms can analyze large volumes of data to identify patterns and trends. This can help admissions officers identify the characteristics of successful students and make informed decisions based on data-driven insights.
* Personalization: Machine learning algorithms can be used to personalize the admissions process by tailoring questions and feedback to individual applicants. This can help applicants feel more engaged with the process and improve their overall experience.
* Fraud detection: Machine learning algorithms can be used to detect fraudulent applications by analyzing patterns and anomalies in the data. This can help prevent admissions fraud and ensure the integrity of the admissions process.

Overall, the application of machine learning in university admissions has the potential to revolutionize the way universities make admissions decisions. By streamlining the process, reducing bias, and improving the accuracy of decision-making, machine learning can help universities select the best candidates and build a more diverse and successful student body

Top of Form

Bottom of Form

**CHAPTER 6**

**CONCLUSION**

Every year millions of students apply to universities to begin their educational life. Most of them don’t have proper resources, prior knowledge and are not cautions, which in turn created a lot problem as applying to the wrong university/college, which further wastes their time, money and energy.

With the help of our project, we have tried to help out such students who are finding difficulty in finding the right university for them. It is very important that a candidate should apply to colleges that he has a good chance of getting into, instead of applying to colleges that they may never get into. This will help in reduction of cost as students will be applying to only those university that they are highly likely to get into this will help in reduction to get into.

Our prepared models work to a satisfactory level of accuracy and may be of great assistance to such people. This is a project with good future, especially of our age group who want to pursue their education in their dream college.

**CHAPTER 7**

**FUTURE SCOPE**

Here are some potential enhancements that can be made in the future for the intelligent admission classification model:

* Feature selection and engineering: While the current model uses all the features available in the dataset, further analysis can be done to identify the most important features for accurate classification. Additionally, new features can be engineered from the existing ones to improve the model's performance.
* Hyper parameter tuning: The current model uses default hyper parameters for the decision tree classifier, but more advanced techniques such as grid search or random search can be used to optimize the hyper parameters for better performance.
* Ensemble learning: Ensemble learning techniques such as random forests or gradient boosting can be used to combine multiple decision tree classifiers to improve the overall accuracy and generalization of the model.
* Improved dataset: The current dataset only contains 200 samples, and collecting more data can help in building a more robust and accurate model.

Overall, there are many potential enhancements that can be made to the classification model, and future research can focus on exploring these areas to improve the accuracy and applicability of the model in settings.

.

**CHAPTER 8**

**APPENDIX**

8.1 SOURCE CODE

**from** **mpl\_toolkits.mplot3d** **import** Axes3D

**from** **sklearn.preprocessing** **import** StandardScaler

**import** **matplotlib.pyplot** **as** **plt** *# plotting*

**import** **numpy** **as** **np** *# linear algebra*

**import** **os** *# accessing directory structure*

**import** **pandas** **as** **pd** *# data processing, CSV file I/O (e.g. pd.read\_csv)*

**import** **seaborn** **as** **sns**

%matplotlib inline

**from** **google.colab** **import** files

**import** **io**

In [3]:

data=files.upload()

data=pd.read\_csv('/content/Admission\_Predict.csv')

In [6]:

data.info()

data.isnull().any()

data=data.rename(columns ={'Chance of Admit ': 'Chance of Admit','LOR ':'LOR'})

In [10]:

data.info()

data.describe()

sns.distplot(data["GRE Score"])

<Axes: xlabel='GRE Score', ylabel='Density'>

ax = sns.heatmap(data.corr(), annot=False)

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

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

data.info()

category=["GRE Score",'TOEFL Score','University Rating','SOP','LOR','CGPA','Research','Chance of Admit']

color=["yellowgreen","gold","lightskyblue",'pink','red','purple','orange','grey']

start=True

**for** i **in** np.arange(4):

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

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

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

plt.title(category[2\*i])

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

data[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()

x=data.iloc[:,0:7].values

x

y=data.iloc[:,7].values

y

**from** **sklearn.preprocessing** **import** MinMaxScaler

sc=MinMaxScaler()

x\_sc=sc.fit\_transform(x)

x\_sc

**from** **sklearn.model\_selection** **import** train\_test\_split

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.3,random\_state=101)

In [44]:

y\_train=(y\_train>0.5)

y\_train

y\_test=(y\_test>0.5)

y\_test

**from** **sklearn.linear\_model** **import** LogisticRegression

lr=LogisticRegression(random\_state=0)

l=lr.fit(x\_train,y\_train)

y\_pred=lr.predict(x\_test)

l

y\_pred

**import** **tensorflow** **as** **tf**

**from** **tensorflow** **import** keras

**from** **tensorflow.keras.layers** **import** Dense, Activation,Dropout

classifier=keras.Sequential()

classifier.add(Dense(7,activation='relu',input\_dim=7))

classifier.add(Dense(7,activation='relu'))

classifier.add(Dense(7,activation='linear'))

classifier.summary()

loss\_1=tf.keras.losses.BinaryCrossentropy()

classifier.compile(loss=loss\_1,optimizer="Adam",metrics=['accuracy'])

classifier.fit(x\_train,y\_train,batch\_size=20,epochs=100)

**from** **sklearn.metrics** **import** accuracy\_score

t\_p=classifier.predict(x\_train)

**print**(t\_p)

tr\_acc=classifier.evaluate(x\_train,y\_train,verbose=0)[1]

**print**(tr\_acc)

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

**print**(test\_acc)

**from** **sklearn.metrics** **import** accuracy\_score,recall\_score, roc\_auc\_score, confusion\_matrix,classification\_report

**print**("Accuracy Score : **%f**" %(accuracy\_score(y\_test,y\_pred)\*100))

**print**("Recall Score : **%f**" %(recall\_score(y\_test,y\_pred)\*100))

**print**("ROC Score : **%f**" %(roc\_auc\_score(y\_test,y\_pred)\*100))

**print**("**\n**Confusion Matrix")

**print**(confusion\_matrix(y\_test,y\_pred))

**print**("**\n**Classification Report")

**print**(classification\_report(y\_test,y\_pred))