

Project Report Template

1.INTRODUCTION

Abstract – College Admission Predictor System is a web based application system in which students can register their marks along with their personal information. This helps to predict their admissions in colleges.

1.1 Overview

The world business sectors are growing quickly and constantly searching generally advantageous information and experience among individuals. Youthful specialists who need to hang out in their positions are continually searching for Higher degrees that can help them in working on their abilities and information. Thus, the quantity of understudies applying for Graduate examinations has expanded in the last decade. One of the principle concerns is getting conceded to their fantasy University. It's seen that understudies actually decide to get their schooling from universities that are known Universally. What's more, with regards to international alumni, the United States of America is the primary inclination of most of them. With most incredibly famous universities, Wide assortment of courses accessible in each order, exceptionally authorize instruction and educating programs, understudy grants are accessible for international understudies .

- Graduate Record Exam1 (GRE) score. The score will be out of 340 focuses.
- Trial of English as a Foreigner Language (TOEFL) score, which will be out of 120 focuses
- University Rating (Uni.Rating) that demonstrates the Bachelor University positioning among different colleges. The score will be out of 5.
- Statement of direction (SOP) which is a record written to show the applicant's life, driven and the inspirations for the picked degree/college. The score will be out of 5 focuses.
- Letter of Recommendation Strength (LOR) which confirms the applicant proficient experience, fabricates validity, supports certainty and guarantees your

ability. The score is out of 5 focuses.

- Undergraduate GPA (CGPA) out of 10.
- Research Experience that can uphold the application, like distributing research papers in gatherings, filling in as examination right hand with college teacher (either 0 or then again 1).

1.2 Purpose

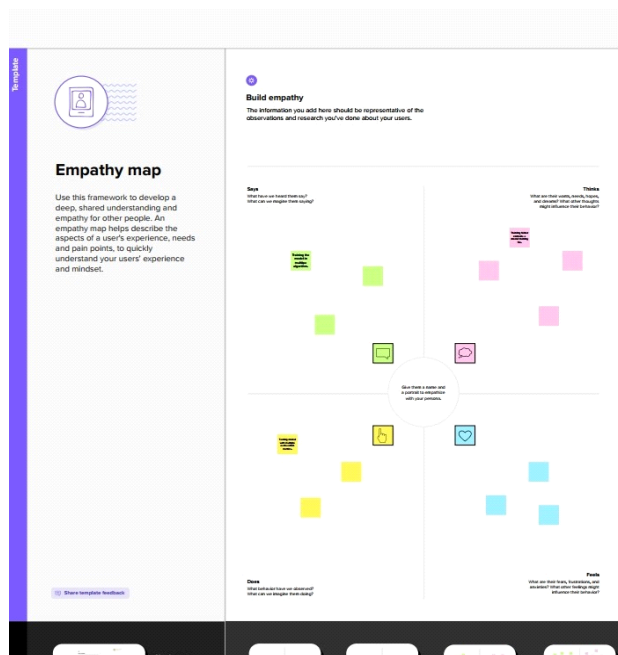
Decision intelligence has three key business benefits: First, it supports data-driven decisions by allowing the full spectrum of relevant data to be analyzed in real time. This enables companies to recognize patterns that would otherwise go undetected as well as potential anomalies that could impact outcomes.

The second major advantage is that decision intelligence empowers companies to reach decisions faster. This is mainly thanks to AI systems, which can process vast volumes of data at hitherto unthinkable speeds, significantly accelerating the entire decision-making process.

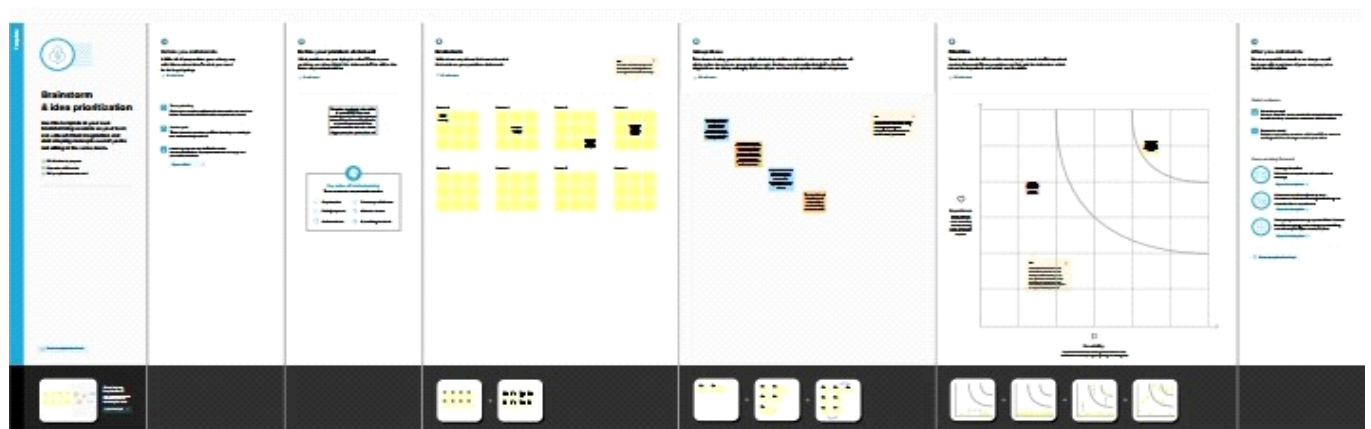
And last but not least, decision intelligence helps eliminate errors and prevent bias. It does this by basing decision on all the available data while avoiding bias by executing programmed algorithms – rather than relying on intrinsically fallible human judgment and gut feeling.

2.Problem Definition & Design Thinking

2.1 Empathy Map



2.2 Ideation & Brainstorming Map



3.Result

As previously stated in the background section, several limitations occurred in the advising process at TU. These limitations can be grouped according to the descriptive or predictive level. In the following sections, we will present some of the contributions that we gain after applying the proposed approach according to these two levels. • **Descriptive Analysis** In the core of AA process, finding the factors that influence the students' final year GPA plays a vital role not only for the students but also for the presidential decisionmakers at the governmental (public) universities level because the increasing number of students who fail to succeed in

their studies on the assigned time adds extra cost and resources. Analyzing the stored data, we found a relative positive correlation between the number of “equivalent contact hours” and students’ ages. Because of the educational regime followed at TU, the departments accept transferred students from other branches as well as from other universities. shows such a correlation. For this reason, we did more investigation to explore the reason behind this correlation. Correlation with GPA. Correlation between student age and amount of equivalent hours. shows the correlation between “student age” and number of “equivalent hours”. Even if this graph does not tell us more information. However, zooming on the age period between we noted that most of the equivalent hours fall in this interval. This can be explained by two reasons: The first reason is the educational regulations followed at Saudi universities that allow students to transfer from other branches or universities. So, when the students find themselves near to fail or hear that the study in other universities is easier than the study in their current university, they -especially fresh and junior students- might want to transfer. The second finding is related to the Deanship of Admission and Registration that tries to count such hours as soon as they can. However, in some cases are delayed and the students start their study even if the equivalence process does not finish yet. In addition to the earlier findings, the head of departments, as well as the deans of colleges, might want to give a general overview of the performance of students. Hence, our tool provides them with such a requirement. presents how is the distribution of students’ grades over the years.

4.ADVANTAGES &DISADVANTAGES

Advantages of Online Admission:-

- **Applicants’ Convenience** – One of the greatest advantages of the online application system is that applicants can choose to submit their applications at their convenience. All that is required is access to a computer and internet connectivity. Messy handwriting, lack of postal connectivity, delay in courier delivery etc. are unlikely to disrupt the application process. This is a great advantage to candidates in rural areas and candidates with disabilities.
- **Logistics** – No more running out of paper application forms, picking the right colour ink pens, illegible prints and wondering if the application has been received at all. The online application process offers university

applicants a uniform platform for filling in their applications and also provides prompts on which fields are mandatory. The acknowledgement is almost immediate and the system user-friendly.

- **Advantage to Universities** – Universities and educational institutions are also at a major advantage when it comes to an online admission process. Quick access to student records and databases, efficient systems for filtering out candidates and processing of applications is possible through the online application process. The costs of processing applications and employing additional manpower during admissions are slashed with the implementation of an online application system.
- **Increases Accuracy and Efficiency** – Those who have seen university officials accepting thousands of paper applications each day at office counters understand that high fatigue and monotony involved in the paperwork is a catalyst for errors. Each error could cost students their academic career and educational prospects. The online admission system is highly reliable and efficient and eliminates chances of such errors.
- **Demolishing Geography** – Another great advantage of the online admission system is that it makes it possible for candidates from across the country and even abroad to apply to Indian universities without any hassles. It eliminates the inconveniences caused by ailments and exigencies, providing deserving candidates a convenience that has never before been available.

Disadvantages of Online Admission:-

- **Computer Literacy and Internet Access** – In India, though Internet penetration is rather high, Internet connectivity and speed issues are major impediments to bring any real advantage to university applicants. Most rural areas experience high blackouts and electricity issues. This means, once again candidates in urban districts and areas are placed at a significant advantage.

- **Low Computer Literacy** – Another major concern is the low rate of computer literacy in India. Current estimates say that only about 6.5 percent Indians are computer savvy. A sudden shift to the online admission process is likely to cause confusion and despondency among a great many applicants.
- **Security Concerns** – In a country like India where security fails of online systems have become increasingly common over the years, online applications make it easier for systems to be breached and for applications or scores to be manipulated. The fear that hackers may target universities and educational institutions is a grave one. Unintentional system failures or server crashes may disrupt the entire admission process of universities and educational institutions. Another important concern is the confidentiality of student information and associated security risks involved in online application processing.
- **Authenticity** – In most manual admission processes, the eligibility of candidates is proved by verification of originals at the time of accepting applications, ensuring that only genuine candidates apply. Online applications make it easier for fraudsters to manipulate the application process and eligibility requirements.
- **Infrastructural Requirements** – Building a robust and secure online admission process is a task that requires financial and infrastructural resources. Many universities and educational institutions may not have the necessary resources and all these costs will ultimately be borne by the students. In a country where higher education is a luxury few can afford, increased costs may be a deterrent for education.

5.APPLICATIONS

Data Collection:

The way toward get-together information relies upon the sort of undertaking, for a ML project, real time information is utilized. The informational index can be gathered from different sources like a document, data set, sensor and different sources and some free informational collections from web can be utilized. Kaggle and UCI Machinelearning Repository are the storehouses that are utilized

the most for information assortment for Machine learning models. Kaggle is quite possibly the most visited sites that is utilized for gathering informational collections [2,3].

Pre-processing:

Information pre-processing is a cycle of cleaning the raw information i.e. the information is gathered in reality and is changed over to a perfect dataset. There are certain steps executed to change over the data into a little clean data collection and make it practical for examination, this piece of the interaction is called as information pre-processing.

The greater part of this present reality information is chaotic, as:

- Missing Data
- Noisy Data
- Inconsistent Data

A portion of the essential pre-handling strategies that can be utilized to change over crude data are:

- Conversion of Data
- Ignoring the missing qualities
- Filling the missing qualities
- Detection of exceptions
- Feature Extraction

At the point when the data to an algorithm is too enormous to ever be processed and it is suspected to be repetitive then it very well may be changed into a diminished arrangement of highlights. Deciding a subset of the initial features is called feature selection. The selected features are required to contain the significant data from the data, with the goal that the ideal assignment can be performed by utilizing this reduced representation rather than the total beginning data. Feature extraction includes reducing the number of assets needed to describe an enormous arrangement of information. When performing investigation of complex

information one of the serious issues comes from the quantity of factors included. Examination with an enormous number of factors for the most part requires a lot of memory and calculation power, likewise it's anything but a grouping calculation to overfit to preparing tests and sum up inadequately to new examples. Highlight extraction is an overall term for techniques for developing blends of the factors to get around these issues while as yet portraying the information with adequate exactness. Many AI experts accept that appropriately improved element extraction is the way to successful model development [4,5].

Model Selection:

Model determination is the way toward choosing one last AI model from among an assortment of applicant AI models for a training dataset. Model selection is a cycle that can be applied both across various kinds of models and across models of a similar sort arranged with various model hyper parameters.

Train and Test Data:

For training a model we at first split the model into 2 segments which are 'training data' and 'Testing data'. The classifier is training utilizing 'training data set', and afterward tests the performance of classifier on inconspicuous 'test data set'.

Training set:

The training set is the material through which the computer learns out how to deal with data. AI utilizes calculations to perform the training part. Training data set is utilized for learning and to fit the parameters of the classifier.

Test set:

A set of unseen data utilized uniquely to evaluate the performance of a completely indicated classifier.

Evaluation Model:

Assessment is an integral of the model development process. It assists with tracking down the best model that addresses the information and how well the picked model will function later on. To work on the model hyper- boundaries of

the model can be tuned and the exactness can be improved. Confusion matrix can be utilized to improve by expanding the quantity of genuine positives and genuine negatives. The output is predicted by investigating the test information as contribution alongside test information output and then the output is shown [6,7,8].

Interface:

A web interface is built to take input and display an output. Flask language is used to build a web interface and pickle library is used to integrate both model and web page.

Data Visualization:

Data visualization is the representation of data in a graph, chart, or other visual format.

6.CONCLUSION

The primary objective of this work is to make a Machine Learning model which could be utilized by understudies who need to seek after their Education. Many AI algorithms were used for this examination. Linear Regression model contrasted with different models gives the best outcome. Understudies can utilize the model to survey their shots at getting induction into a specific University with a normal exactness of 82%. An ultimate objective of examination will be cultivated effectively, as the framework permits understudies to save the parcel of time and cash that they would spend on instructive guides and application charges for schools where they have less shots at getting affirmations. In future this module of expectation can be incorporated with module of robotized handling framework and different models like neural organization. Likewise, segregate investigation can be utilized independently or joined for upgrading dependability and precision forecast. At long last, understudies can have an open-source AI model which will assist the understudies with knowing their opportunity of entrance into a specific college with high exactness.

7.FUTURE SCOPE

From the proposed work we are able to identify only chance to get seat and we are not able to identify which university we are obtaining. So, in future we

can develop a representation, which gives us a list of universities in which we can obtain admission.

8.Appendix

Importing the libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

Read The Dataset

```
data = pd.read_csv('Admission_Predict.csv')
```

Handling Missing Values

```
data.info()
data.isnull().any()
data=data.rename(columns = {'Chance of Admit ' : 'Chance of Admit'})
```

Exploratory Data Analysis

Descriptive statistical

```
data.describe()
```

visual analysis

Univariate analysis

```
Sns.distplot(data['GRE Score'])
```

Bivariate analysis

Pair plot

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

Scatter plot

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

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

```
color=['yellowgreen','gold','lightskyblue','pink','red','purple','orange','gray']
```

```
start = True
```

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

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

```
    plt.subplot2grid((4,2),(1,0))
```

```
    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.subplot_adjust(hspace = 0.7, wspace = 0.2)
```

```
plt.show()
```

Scaling the Data

```
from sklearn.preprocessing import MinMaxScaler
```

```
sc = MinMaxScaler()
```

```
x=sc.fit_transform(x)
```

```
x
```

Splitting data into x and y

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

```
x
```

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

```
y
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x,y,  
test_size=0.30,random_state=101)
```

Let us convert it into classification problem

```
y_train=(y_train>0.5)
```

```
y_train
```

```
y_test=(y_test>0.5)
```

Model Building

Logistic Regression Model

```
from sklearn.linear_model.logistic import LogisticRegression
```

```
cls =LogisticRegression(random_state =0)
```

```
lr=cls.fit(x_train,y_train)
```

```
y_pred =lr.predict(x_test)
```

```
y_pred
```

ANN Model

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import Dense, Activation, Dropout
from tensorflow.keras.optimizers import Adam
model=keras.Sequential()
model.add(Dense(7,activation = 'relu',input_dim=7))
model.add(Dense(7,activation = 'relu'))
model.add(Dense(7,activation = 'linear'))
model.summary()
model.fit(x_train, y_train, batch_size = 20, epochs = 100)
```

Testing the Model

```
model.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics
=['accuracy'])
model.fit(x_train, y_train, batch_size = 20, epochs = 100)
from sklearn.metrics import accuracy_score

train_predictions = model.predict(x_train)
print(train_predictions)

train_acc = model.evaluate(x_train, y_train, verbose=0)[1]
```

```
print(train_acc)
```

```
test_acc = model.evaluate(x_test, y_test, verbose=0)[1]
```

```
print(test_acc)
```

```
print(classification_report(v test.pred))
```

```
pred=model.predict(x_test)
```

```
pred = (pred>0.5)
```

```
pred
```

Performance Testing & Hyperparameter Tuning

Compare the model

Logistic Regression model

```
from sklearn.metrics import
```

```
accuracy_score,recall_score,roc_auc_score,confusion_matrix
```

```
print("\n 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\n" %(roc_uac_score(y_test,y_pred) * 100))
```

```
print(confusion_matrix(y_test,y_pred))
```

ANN Model:Training Accuracy

```
from sklearn.metrics import
```

```
accuracy_score,recall_score,recall_score,roc_uac_score
```

```
print(classification_report(y_test,pred))  
from sklearn.metrics import accuracy_score,recall_score,roc_auc_score  
print(classification_report(y_test,pred))
```

Model Deployment

Save the best model

```
model.save('model.h5')
```

Integrate with Web Framework

Build python code

```
import numpy as np  
from flask import Flask, request, jsonify, render_template  
import pickle  
app = Flask(__name__)
```

```
from tensorflow.keras.models import load_model
```

```
model = load_model('model.h5')
```

```
@app.route('/')  
def home():
```

```

        return render_template('Demo2.html')
@app.route('/y_predict',methods=['POST'])
def y_predict():
    for rendering results on HTML GUI

min1=[290.0, 92.0, 1.0, 1.0, 1.0, 6.8, 0.0]
max1=[340.0, 120.0, 5.0, 5.0, 5.0, 9.92, 1.0]
k=[float(x) for x in request .form.values()]
p=[]
    for i in range(7):
        l=(k=[i]-min1[i])\ (max1[i]-min1[i])
        p.append(1)
prediction = model.predict([p])
print(prediction)
output=prediction[0]
if(output==False):
    return render_template('nochance.html', prediction_text='You Don't
have a chance of getting admission')
else:
    return render template('chance.html',prediction text='You have a
chance of getting admission')
if __name__=="__main__":

```



```
app.run(debug=False)
```