**THIRUVALLUVAR UNIVERSITY**

PERIYAR ARTS COLLEGE

CUDDALORE-607 001

COLLEGE CODE 105

**DEPARTMENT OF COMPUTER APPLICATION**

## PROJECT TITLE:

# Intelligent Admissions: The Future of UniversityDecision Making with Machine Learning

**TEAM LEADER :** JAYARAJ C

**TEAM MEMBERS :** YUVARAJ R

NARENDIRAN P

HARIDOSS P

# Introduction

## Overview

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.

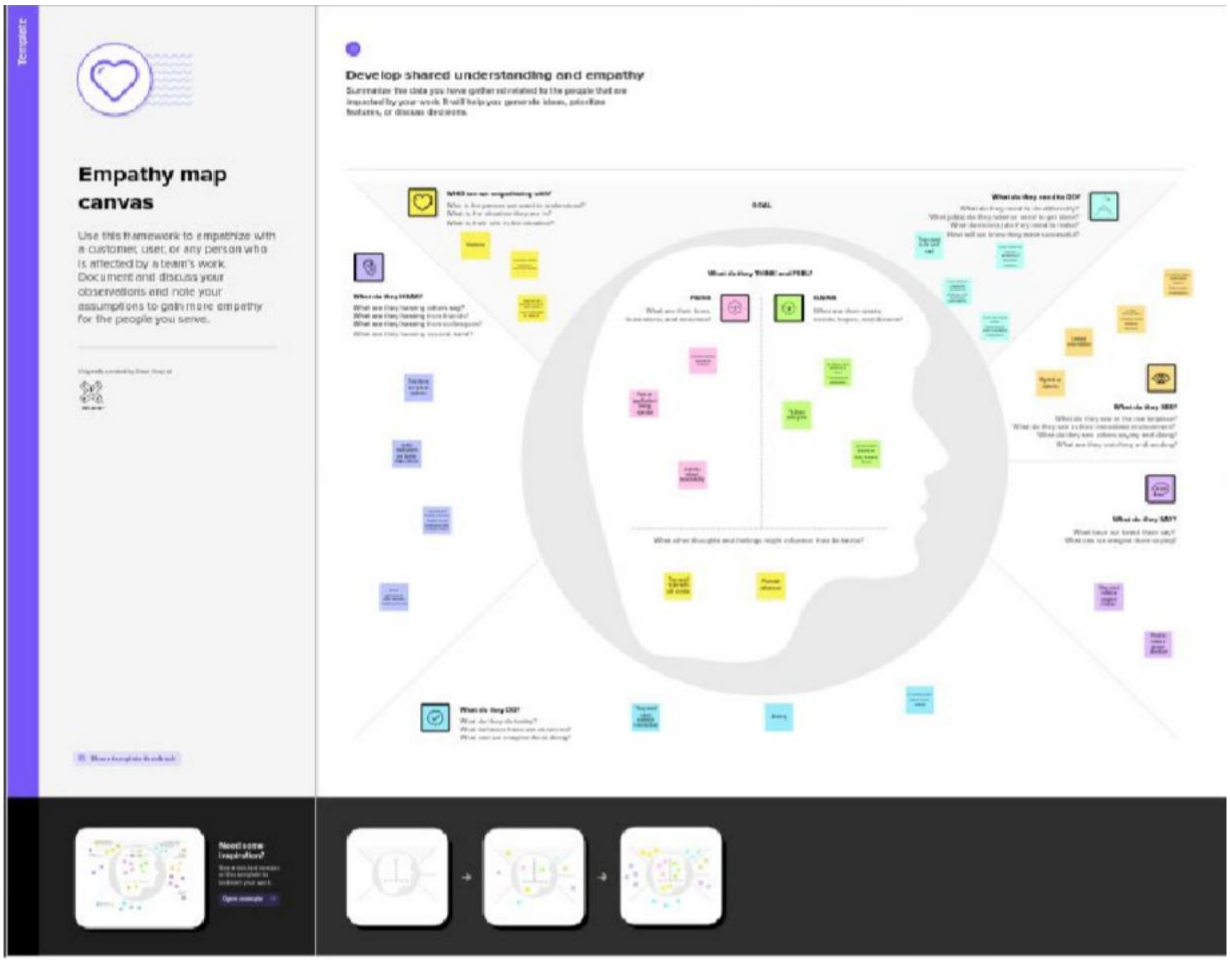
With this project, students can make more informed decisions about which universities to apply to, and universities can make more efficient use of their resources by focusing on the most promising applicants.The predicted output gives them a fair idea about their admission chances in a particular university. This analysis should also help students who are currently preparing or will be preparing to get a better idea.

## Purpose

This project can help students to predict their chance of admission in universities.The project is employed with a well trained model which can predict accurate results. With proper user(student) input, the flask web app can predict the result with the trained model.

# Problem Definition and Design Thinking

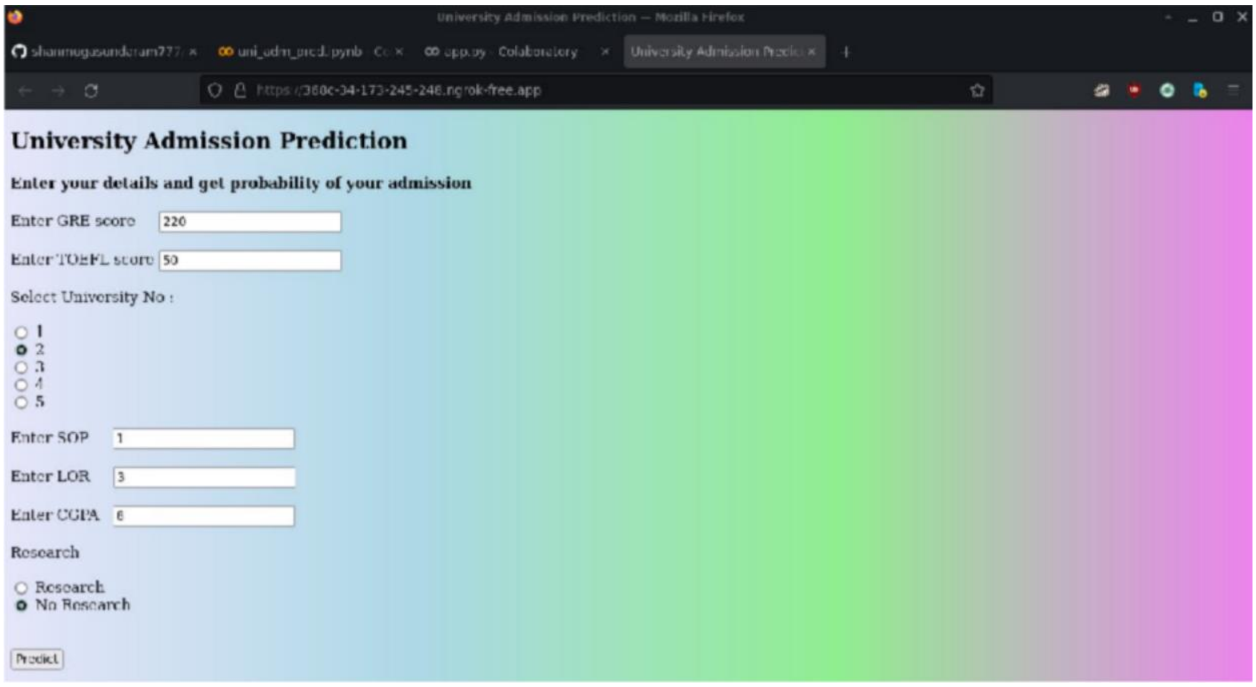
## Empathy Map


Template
Empathy map canvas
Use this framework to empathize with a customer, user, or any person who is affected by a team's work. Document and discuss your observations and note your assumptions to gain more empathy for the people you serve.
Develop shared understanding and empathy
GOAL
What do they THINK and FEEL?
What do they HEAR?
What do they SEE?
What do they SAY?
What do they DO?
Need some inspiration?
Open example

## Brainstorm and Idea Prioritization

# Result

### Homepage


University Admission Prediction — Mozilla Firefox
shanmugasundaram777/
uni_adm_pred.ipynb - Co
app.py - Colaboratory
University Admission Predic
https://360c-34-173-245-246.ngrok-free.app

**University Admission Prediction**

**Enter your details and get probability of your admission**

Enter GRE score 220

Enter TOEFL score 50

Select University No :

○ 1
[✓ 2]
○ 3
○ 4
○ 5

Enter SOP 1

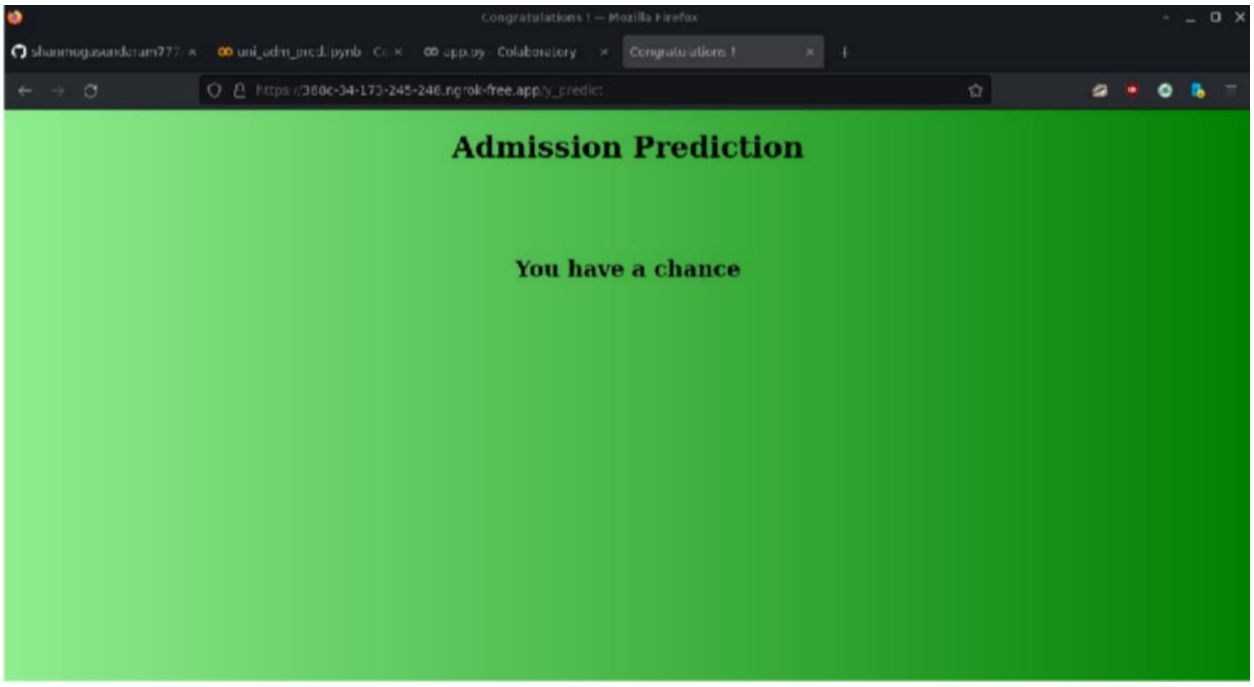
Enter LOR 3

Enter CGPA 6

Research

○ Research
[✓ No Research]

### Prediction


Congratulations ! — Mozilla Firefox
shanmugasundaram777/
uni_adm_pred.ipynb - Co
app.py - Colaboratory
Congratulations !
https://360c-34-173-245-246.ngrok-free.app/y_predict

**Admission Prediction**

**You have a chance**

"Chance of Admit" depends on CGPA,GRE,TOEFEL.The columns SOP,LOR and Reserach have less impact on university admission.

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

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

## Advantages and Disadvantages

### Advantages

The project takes into account of all the necessary variables that determine the admissionof students in universities.

The variables are absolutely bare minimal and is strictly required to perfrom accurate predictions.

It help students to get a preliminary prediction of how their profile/score may perform on the university prediction process.

It enables students to get their overall image on university shortlistings...

### Disadvantages

This projects only takes into account of minimal variables and special edge cases are not considered.

It omits certain special cases which the students and universities may have come across.

The dataset used is merely adequate not dense enough to train high or world class models that is capable of predicting complex inputs while producing accurate results.

Currently, this project makes uses of web technologies such as Google Colab and Ngrok.

Due to poor hardware at our end we have used the web technologies, to deploy the code in other environments and produce a valid webapp, the code must be modified to be run locally on the development server.

## Applications

This project unfortunately can only be applied to Education Fields especially, Universities.

And it is only useful for students to assess their profile performance.

The project may be further modified to fit other educational fields such as schools and other educational bodies, that assess student’s past performance to allow them in their institutions.

## Conclusion

This project is very useful for students to assess their overall profile performance that is necessary to get a overall preliminary of their profile before applying applications to various universities.

It helps them to filter out universities that fit their profile and apply to selected universities that has high rate of being accepted.

## Futute Scope

Currently, the project uses a bare minimal dataset that has entries of about 400 rows.

Although this dataset is enough, it is not feature rich.

To further enchance the project, a more dense and feature rich dataset is to be used.

The additional complex features helps in addressing edge or special cases that the students and universities may encounter during their academical span.

Additional algorithms will be used to proof the accuracy of the dataset split. And feature rich data helps in creating and training a model that can generate much precise and accurate results while handling special or edge cases.

The code needs to be further enhanced to get additional variable inputs from user that are helpful in overall prediction with much specialised and accurate results.

# Appendix

## Source code

***uni_adm_pred.ipynb***

```
# -*- coding: utf-8 -*-
"""uni_adm_pred.ipynb


Automatically generated by Colaboratory.


Original file is located at
    https://colab.research.google.com/drive/1ICJRuL_-58wSg5HO7dbBhZ3pdeaCNByc
"""


# Commented out IPython magic to ensure Python compatibility.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# %matplotlib inline


# Commented out IPython magic to ensure Python compatibility.
# mount google drive and copy dataset
from google.colab import drive
drive.mount('/content/drive')


# %cp '/content/drive/My
Drive/university_admission_prediction/dataset/Admission_Predict.csv' '/content/'
```

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


data.head()


data.drop(["Serial No."],axis=1,inplace=True)
data.head()


data.describe()


data.info()


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


data.isnull().any()


data.corr()


sns.heatmap(data.corr(),annot=True,cmap="RdYlGn")


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


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), (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:-1].values
y = data['Chance of Admit'].values

from sklearn.preprocessing import MinMaxScaler

sc = MinMaxScaler()
x = sc.fit_transform(x)

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_state=42)
```

```
y_train = (y_train>0.5)

y_test = (y_test>0.5)


from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, recall_score, roc_auc_score,
classification_report, confusion_matrix


cls =  LogisticRegression(random_state=0)

lr = cls.fit(x_train, y_train)

y_pred = lr.predict(x_test)


print("Logistic Regression")

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("Confusion Matrix \n", confusion_matrix(y_test, y_pred))


import keras

from keras.models import Sequential

from keras.layers import Dense


model = Sequential()


model.add(Dense(units=7, activation='relu', input_dim=7))


model.add(Dense(units=7, activation='relu'))
model.add(Dense(units=1, activation='linear'))


model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])


model.fit(x_train, y_train, batch_size=20, epochs=100)


ann_pred_tr = model.predict(x_train)

print(ann_pred_tr)


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)


ann_pred = model.predict(x_test)

ann_pred = (ann_pred>0.5)

ann_pred_tr = (ann_pred_tr>0.5)


ann_pred


print("ANN Model \n")


print("Train Prediction \n")

print("Confusion Matrix \n", confusion_matrix(y_train, ann_pred_tr))

print("\n Classification Report \n", classification_report(y_train, ann_pred_tr))
print("Prediction \n")

print("Confusion Matrix \n",confusion_matrix(y_test, ann_pred))

print("\nClassification Report \n", classification_report(y_test, ann_pred))


# Commented out IPython magic to ensure Python compatibility.

# save and copy the saved model to mounted google drive

model.save('model.h5')


# %cp '/content/model.h5' '/content/drive/My
Drive/university_admission_prediction/training/'
```

***app.py***

```
# -*- coding: utf-8 -*-

"""app.py


Automatically generated by Colaboratory.


Original file is located at

    https://colab.research.google.com/drive/1a2qzmOmvcbmvJXqsR0ouQ47MPgYnBFhW

"""


import numpy as np

from flask import Flask, request, jsonify, render_template


# install pyngrok

!pip install pyngrok
```

```
# import ngrok
from pyngrok import ngrok


from tensorflow.keras.models import load_model


# Commented out IPython magic to ensure Python compatibility.
from google.colab import drive
drive.mount('/content/drive')


# %cp '/content/drive/My Drive/university_admission_prediction/training/model.h5'
'/content/'
# %cp -r '/content/drive/My Drive/university_admission_prediction/flask/templates'
'/content/'


!killall ngrok


app = Flask(_name__)


ngrok.set_auth_token("2NejXWwLPdVu3YU8rQUBjbIPzT5_6kpmTTTf41ZBxUDwtDtso")


public_url = ngrok.connect(5000)
print("URL : ", public_url)


model = load_model('model.h5')


@app.route('/', methods=['GET', 'POST'])
```

```
def home():
    return  render_template('home.html')


@app.route('/y_predict', methods=['POST'])
def y_predict():


    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('no_chance.html')
    else:
        return  render_template('chance.html')


app.run()
```