Assignment – 4

Problem – 1

Team LA

Abstract

Linear Regression performs the task to predict a dependent variable(y) based on a set of an independent variable(x). Plotted in 2D, this gives us a straight line which best fits the given data – points. Our job is to find the most appropriate m & c in the following :

$$y_{fit} = mx + c$$

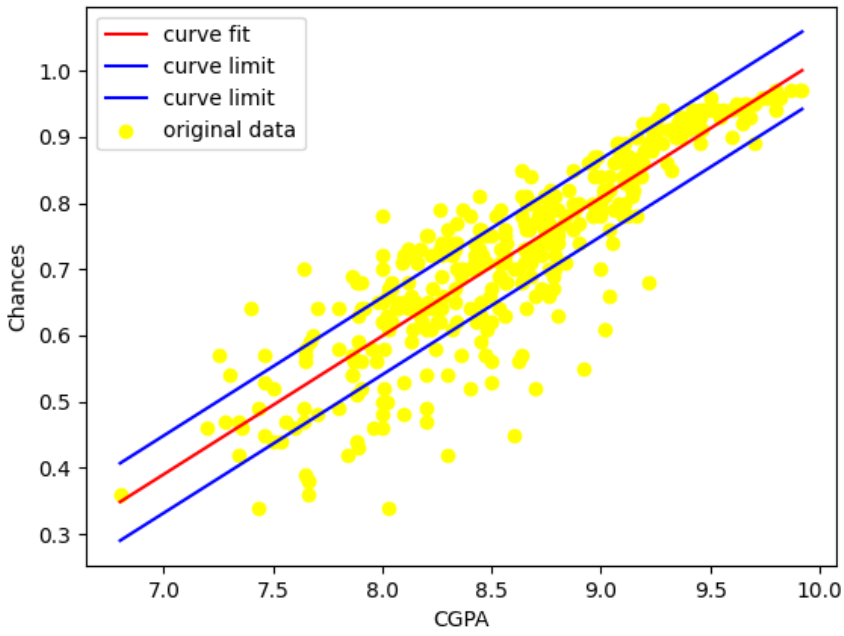
Our analysis :

We performed an exploratory data analysis for the first problem and stuck to using data visualisation as our primary tool as much as we could. So, does CGPA really matter? Well, the ans is a mix of yes and no. The Universities seem quite flattered with a very good CGPA (something above 8.5) but that simply does not mean that an average or a below average student just can't make it in. Refer to our graphs and data below.

Graphs and Data :

Contrary to what is prevalent in the data science ecosystem, we have used the scipy **lingress** method to keep it as simple as possible.

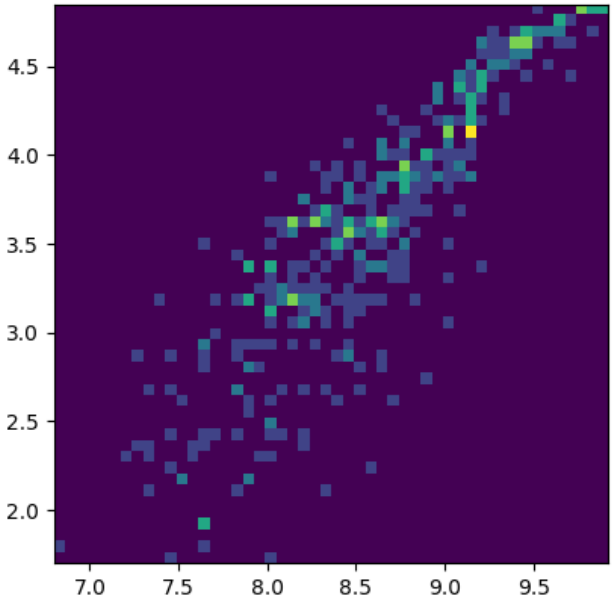
1.We present our first plot :



The yellow scattered plot is the original "Chance probability vs CGPA"
The red line is the best fit line.
The blue lines accompanying it are at an arbitrary distance 0.05 above and below it

Roughly around the 0.8 admission chance, the plots begin to streamline. This motivated us to validate our statements using yet another beautiful data visualisation plot : The heat map.

The Heatmap wasn't fitting quite well, so we multiplied the chances by 5. Nonetheless, this doesnt disturb the trend and displays an accurate clustering and streamlining of data points at the end.



1. $m_{fit} \approx 0.20885$
2. $c_{fit} \approx -1.07151$
3. RMS error of the entire fit is : 2.9602
4. RMS error of the last 10 values is : 0.02046 (less than a 100 times)

Conclusion :

A good CGPA is a solid advantage but something less than that must be well padded with projects, LORs, and a well written SOP.

The Code :

Before executing it, sort CGPA in ascending order and change column name " Chance of Admit " to " ChanceofAdmit " (without spaces anywhere).

```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
from scipy import stats
from sklearn import metrics

path = "Admission_Predict.csv"
df = pd.read_csv(path)

y = np.array(df['ChanceofAdmit'])
x = np.array(df['CGPA'])

y_end = y[390:]
print(len(x))
slope, intercept, r_value, p_value, std_err = stats.linregress(x, y)
```

```
y_fit = slope*x + intercept
std_err *= 10
y_max= y_fit + std_err
y_min = y_fit - std_err

y_fit_end = y_fit[390:]

plt.scatter(x, y, color='yellow', label='original data')

plt.plot(x, y_fit, color='red', label='curve fit')
plt.plot(x, y_max, color='blue', label='curve limit')
plt.plot(x, y_min, color='blue', label='curve limit')

plt.xlabel('CGPA')
plt.ylabel("Chances")
plt.legend()
plt.show()

y = y*5

heatmap, xedges, yedges = np.histogram2d(x, y, bins=50)
extent = [xedges[0], xedges[-1], yedges[0], yedges[-1]]

plt.clf()
plt.imshow(heatmap.T, extent=extent, origin='lower')
plt.show()

total_MS = metrics.mean_squared_error(y, y_fit)
print(total_MS)
print(f"RMS error of the fit is: {np.sqrt(total_MS)}")

end_ms = metrics.mean_squared_error(y_end, y_fit_end)
print(f"rms error for the last 10 values: {np.sqrt(end_ms)}")
```