COMP40370 Practical 4

[redacted]

October 22, 2020

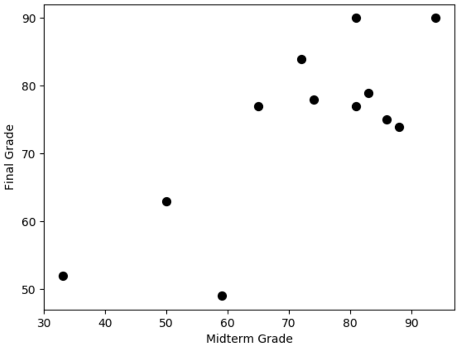
1 Question 1

Question 1 - 1

There are 12 Data Points. I Plotted the Data in a Scatter Plot. While There is a vague Linear Nature to the data there's a lot of scatter around the top.

If we have a look at the correlation We see the **Correlation Coefficient is 0.782** which means that the Final and Midterm are ***Slightly Weakly Correlated***.

Figure 1: Marks plot



Question 1 - 2

I generated a Linear Model Using SKLearn's Linear Regression Class. This model takes in a number and will return the best estimate. Linear Models Take on the form of:

$$y = \beta_0 + \beta_1 x + \epsilon \tag{1}$$

This is the General form where β_0 is the y Intercept. i.e. the y value at the position $x = 0$. For our model $\beta_0 = 32.0278$ and $\beta_1 = 0.58169$. (We ignore ϵ as we assume its 0 for Simple Linear Regression) So we end up with the equation:

$$y_i = 32.0278 + 0.58169 * x_i \tag{2}$$

Equation 2 is essentially what our Linear model is. When we predict a value we sub in for x. This model contains information. The model Tells us that if someone has a midterm score of 0 their predicted final score will be 32. It also tells us that for every 1 point increase in midterm score their final score increases by roughly 0.6 points.

Obviously we don't need to worry about this though as its all hidden from us by SKlearm. but I accessed these numbers using the **intercept_** and **coef_** properties of the model.

Question 1 - 3

As per the Linear Model we can Predict the Final Result of a Student who scored 86 on the midterm by applying **Equation 2** which gives us **82.045...**. But in python we simply need to call:

```
prediction = regression_model.predict(np.array([[86]])).item()
```

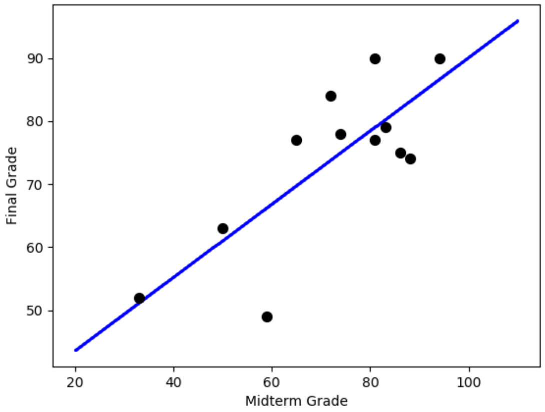


Figure 2: Fitted Regression Line

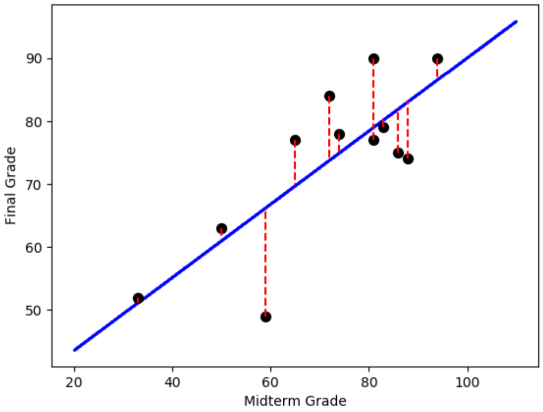


Figure 3: Fitted Regression Line with Residuals

2 Question 2

Question 2 - 1

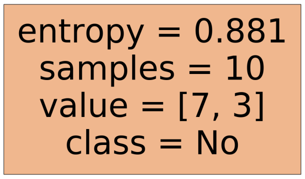
This is as simple as *popping* TID off of the DataFrame.

Question 2 - 2

So we Created a Decision Tree with a Minimum Impurity Decrease of 0.5.

The resulting Tree ends up being 1 Leaf node (See Figure 4) of *Class 'No'*. This means this Decision Tree will classify everything as *'No'*. **The Classification result is very poor.**

Figure 4: DecisionTree (0.5)



Question 2 - 3

Since our last Decision Tree classifier was so poor we Create a new one but this time we Try a Minimum Impurity Decrease of 0.1

This Time we get a much more promising result. We were able to obtain a Decision Tree where all our leaf nodes are all *Pure*. This means that our Classifier is likely to perform well at its task which would make **The Classification Result Accurate**.

Compared to the last tree that would classify everything as *'No'* our new Tree actually has *"Decisions"* and **has managed to split all the training data correctly**.

Question 2 - 4

I will now Discuss the Generated Decision Tree Models. Why did our Initial Tree (Figure 4) perform so badly?

Well A Minimum Impurity Decrease **(referenced as: MID)** decides whether A node will be split if this split induces a decrease of the impurity greater than or equal to this value. *(Note the we can think of Impurity as a measure of* How Mixed *the classes are in a node)*

And the initial Models MID was 0.5 meaning that if a split didn't decrease the Impurity by at least 0.5 then the node wouldn't split and that is what happened resulting in the one leaf Tree.

The Second Trees MID was 0.1 (Figure 5) meaning that a split would only need to decrease the impurity by 0.1 for the split to happen. We can see this in effect with all nodes in that in the first node the *"Mix"* of classes is 7:3 which is a massive impurity. then when we move to the next node after the split the *"Mix"* is only 4:3. Still very impure but its decreased more than 10% since the node before it!

From the result we can see that the Nodes were able to reduce the impurity to split enough times to end up with all pure leafs.

I also, out of curiosity, looked at the **Figure 5 importance of features:**

Homeowner	Marital Status	Annual Income
0.41433362	0.1412828	0.44438358

And this pretty much matches what we see In the figure. Annual Income breaks up samples twice. HomeOwner only splits data once but it correctly filters 3 samples.

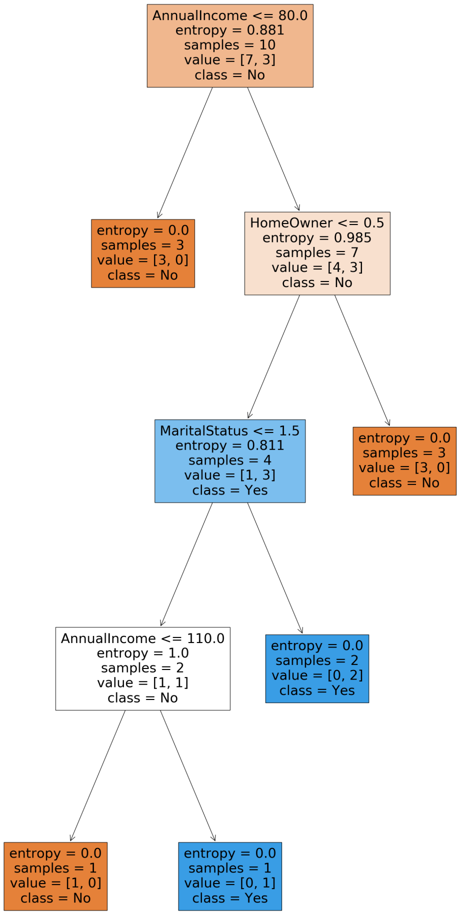


Figure 5: DecisionTree (0.1)