CSC 177: Data Warehousing and Data Mining

Project 2: Linear Regression and Classification Tree

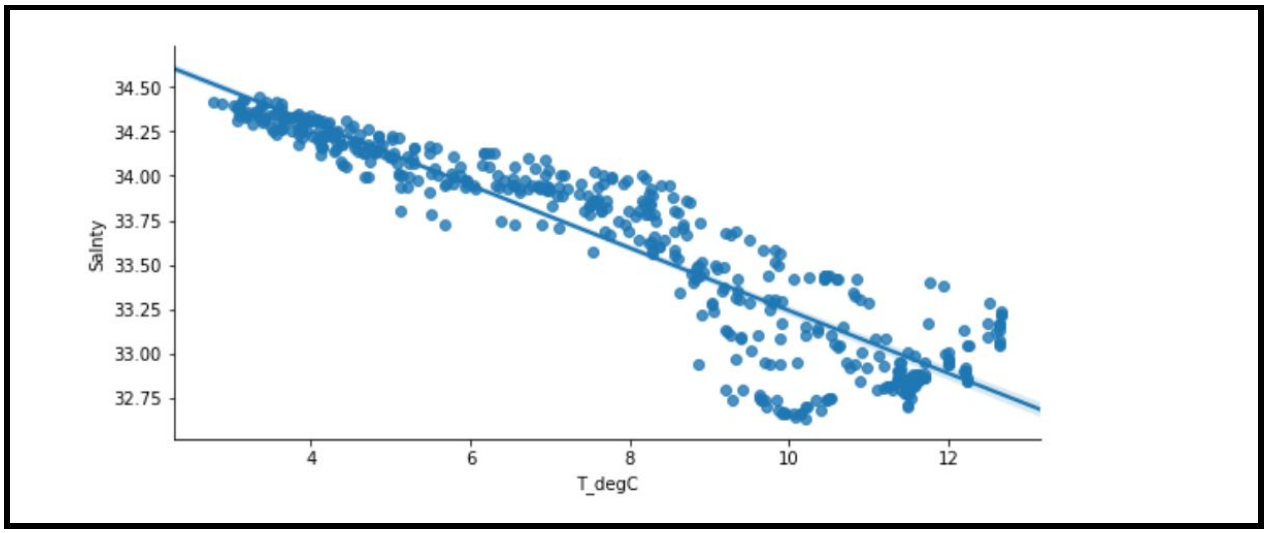
Jagan Chidella

Group: An Lam, Jimmy Le, Tom Amir,
Dianna Melendez, Amrit Singh, Talal Jawaid, Min Li

PART A:

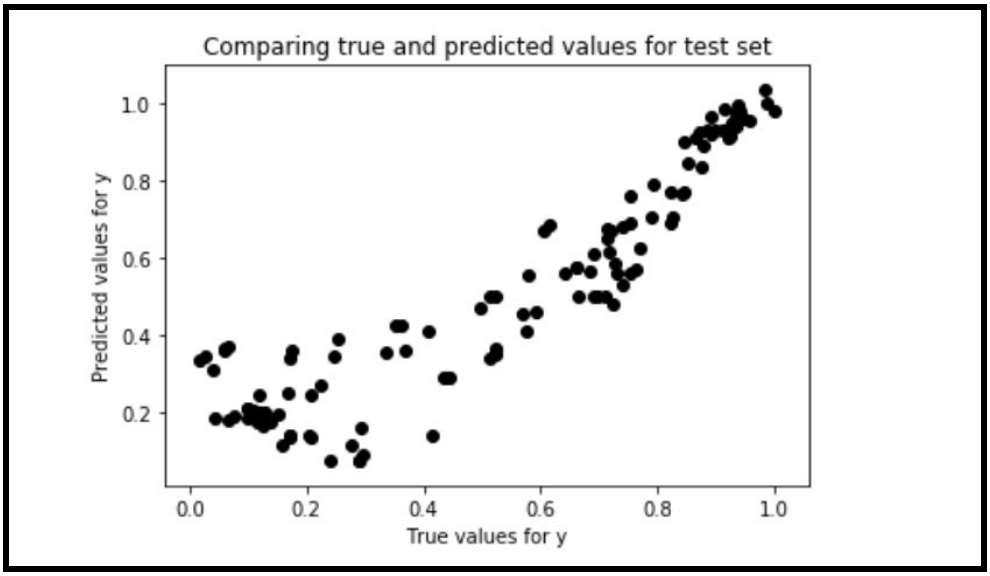
For Linear regression, at first we chose to use the London Air data set. However, when attempting to perform the linear regression on that data set, it was producing a very low regression value. Though we understand the importance of normalizing data, even when we attempted to normalize the data and sample a small section of the data, eliminating outliers and missing values and replacing them with the median, the result was very poor. Due to this, we decided to switch data sets and chose to use a data set which measured the salinity and temperature of seawater. We were able to use linear regression to accurately predict the salinity of the water given the temperature of the water using our training data. We believe the previous London Air dataset did not give us an acceptable output due to the fact that there is no correlation between the date and the number of particulates in the air. This follows the concept of garbage in, garbage out.

Linear regression applied on data set

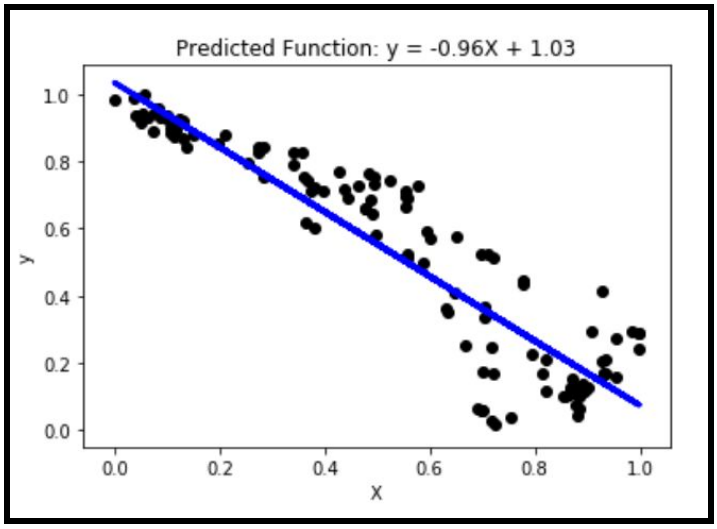


Multiple Regression simply continues from the linear regression process. There are two added steps, where you evaluate the performance of your model on the test set and then a final post-processing step.

Evaluating Model Performance on Test Set



Post-Processing



PART B:

Manually study the data. Make any random observations about the data.

- Serial Numbers in Ascending Order
- TOEFL: Test of English as Foreign Language
- University Rating on a scale of 1 - 5
- GRE Score: out of 350
- SOP: Statement of Purpose
- LoR: Letters of Recommendation
- Research: Either 0 or 1, representing boolean values, presumably 0 if they have not done research and 1 if they have

Regression steps:

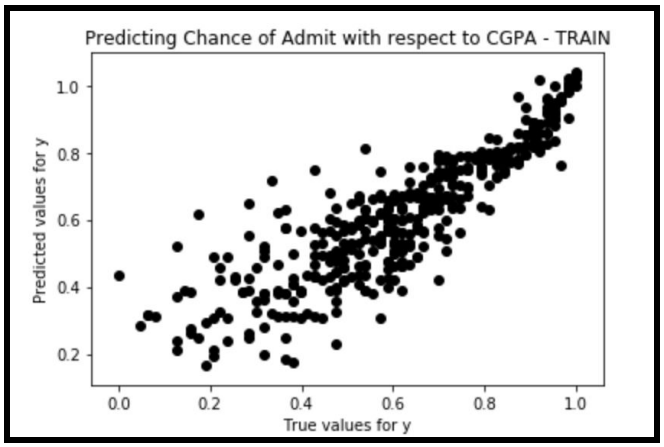
1) Split the data into training data, and test data

   We split the data into the standard 80:20. Since we did not know the distribution of the data in the rows, we performed "Shuffling Dataframes" on the data set before partitioning the data into the training data, and test data.

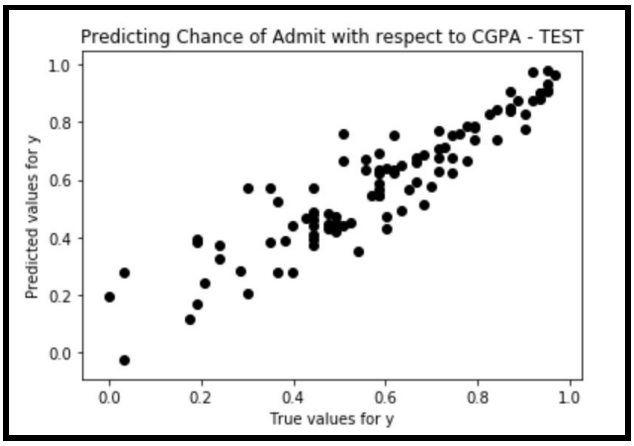
2) Apply Linear Regression & Multiple Regression on both training and test data

   After performing Linear Regression and Multiple Regression on both data sets, here are the results:

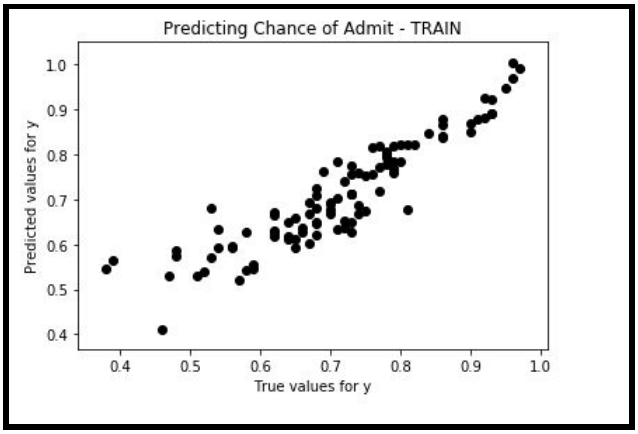
Linear Regression on Training Data:



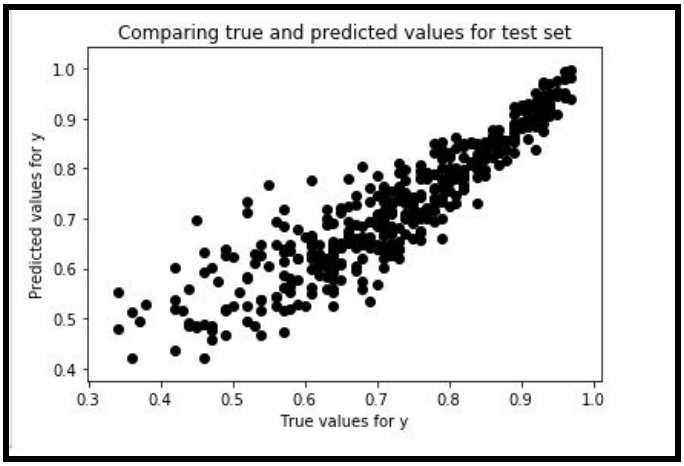
Linear Regression on Test Data:



Multiple Regression on Training Data:



Multiple Regression on Test Data:



3) Observations/Analysis on the Regression performed on the Training Data and Test Data:

   Simple Regression Analysis: When performed a simple regression on both the training data, and the test data, it can be seen that the curve that is created is very similar. There are more data points seen in the training data curve, but that is because it has 4x the amount of data points. It is significant that the curves look very similar, because what can be seen is that with more data points, the more accurate a linear regression line becomes.

Classification:

Process:

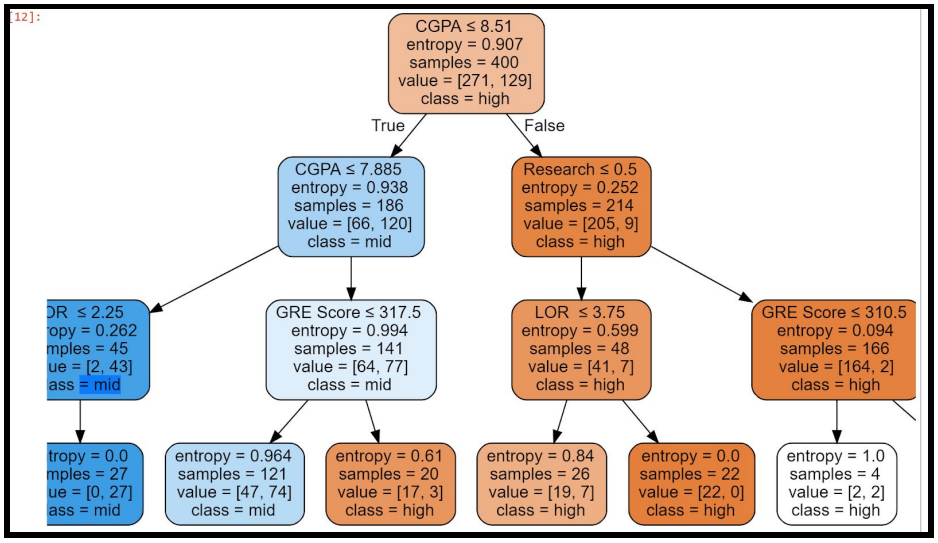
1) We discretize the last column "Chance of Admit" into three classes after shuffling. We classify 0-0.33 is low, 0.34-0.66 is mid, and 0.67-1.00 is high.

	Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
0	267	312	105	2	2.0	2.5	8.45	0	high
1	358	301	104	2	3.5	3.5	7.89	1	high
2	433	324	112	4	4.5	4.0	9.22	1	high
3	155	326	108	3	3.0	3.5	8.89	0	high
4	300	305	112	3	3.0	3.5	8.65	0	high
5	310	308	110	4	3.5	3.0	8.60	0	high
6	222	316	110	3	3.5	4.0	8.56	0	high
7	70	328	115	4	4.5	4.0	9.16	1	high
8	478	309	105	4	3.5	2.0	8.18	0	mid

2) We split the data 80/20 for training/testing. First we drop the Serial No. column because it is useless. The column we want to predict is Chance of Admit so we set the column to Y, and the rest to X. The tree's criterion is entropy with max depth of 3.

```
Out[6]: DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=3,
                               max_features=None, max_leaf_nodes=None,
                               min_impurity_decrease=0.0, min_impurity_split=None,
                               min_samples_leaf=1, min_samples_split=2,
                               min_weight_fraction_leaf=0.0, presort=False, random_state=None,
                               splitter='best')
```

3) We can also display the tree with graphviz.



4) Finally, we can try predicting the classes based on our test data from the original dataset. The accuracy on the test data is 73%.

```
Accuracy on test data is 0.73
```

Out[64]:

	Chance of Admit	Predicted Class
0	high	high
1	mid	high
2	mid	mid
3	mid	high
4	high	high
5	mid	mid
6	high	high
7	high	high
8	high	high
9	mid	high

5) Conclusion: We found out that with higher number of max depth on the decision tree the prediction accuracy becomes better. For example, we tried max depth of 6 and the accuracy jumps up to 85% with no other changes. However, the accuracy doesn't scale with max depth. We tried max depth of 24 and the accuracy went down to 84%.

PART C:

Part C – I

	color	shape	size	Pattern (new attribute)	class
1	red	square	big	checked	+
2	blue	square	big	striped	+
3	red	round	small	dotted	-
4	green	square	small	checked	-
5	red	round	big	striped	+
6	green	round	big	dotted	-
Red					
1	red	square	big	checked	+
3	red	round	small	dotted	-
5	red	round	big	striped	+
Checked					
1	red	square	big	checked	+
4	green	square	small	checked	-

$Entropy(t) = -\sum p(j|t) \log 2\ p(j|t)$

1. Calculating initial entropy

   P(+) = -($\frac{3}{6}$) * $\log_2$($\frac{3}{6}$) = 0.5
   P(-) = -($\frac{3}{6}$) * $\log_2$($\frac{3}{6}$) = 0.5
   $Entropy(t) = E(t)$ = 0.5 + 0.5 = 1

2. Calculating entropy and gain for each attribute color

   $E(Color{=}red)$= –($\frac{2}{3}$) * $\log_2$ $\frac{2}{3}$ - $\frac{1}{3}$ * $\log_{\frac{1}{3}}$ ≈ 0.92

   $E(Color{=}$blue)= –($\frac{1}{1}$) * $\log_2$ $\frac{1}{1}$ - 0 = 0

   $E(Color{=}$green)= –(0) – $\frac{2}{2}$ * $\log_2$ $\frac{2}{2}$ = 0

   Average Entropy = $\frac{3}{6}$ (0.92) + $\frac{1}{6}(0) + \frac{2}{6}(0) = 0.46$
   Gain (Outlook) = 1 – 0.46 = 0.54

3. Calculating entropy and gain for each shape

   E(shape=square)= $-\frac{2}{3} * \left(\frac{2}{3}\right) \ -\frac{1}{3} * log2\frac{1}{3}$ ≈ 0.92

   E(shape=round)= $-\frac{1}{3} * log2\left(\frac{1}{3}\right) - \frac{2}{3} * log2\left(\frac{2}{3}\right) = 0.92$

   Average Entropy = $\frac{3}{6}(0.92) + \frac{3}{6}(0.92) = 0.92$
   Gain(Outlook) = 1 – 0.92 = 0.08

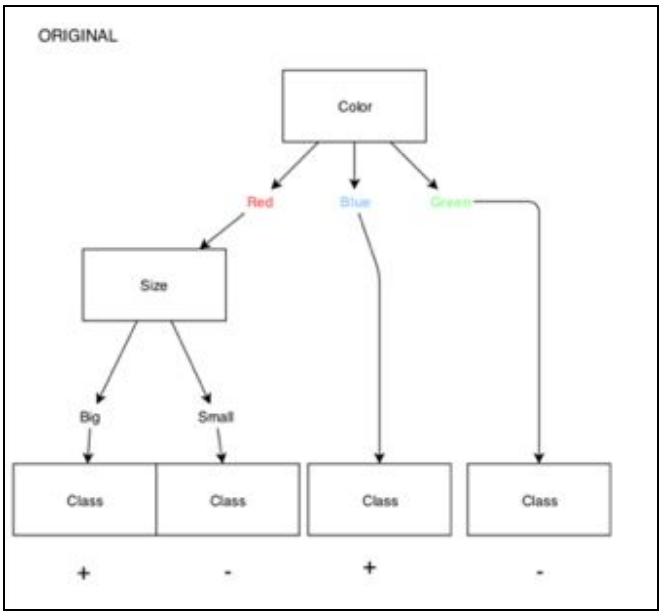
4. Calculating entropy and gain for attribute size

   E(size=big) = $-\frac{3}{4} * log2\ \left(\frac{3}{4}\right) - \frac{1}{4} *\ log2\left(\frac{1}{4}\right)$ ≈ 0.81

   E(size=small) = $0 - \frac{2}{2} * log2\left(\frac{2}{2}\right) = 0$

   Average Entropy = $\frac{4}{6}(0.81) + \frac{2}{6}(0) = 0.54$
   Gain(Outlook) = 1 – 0.54 = 0.46

5. Original Decision Tree



6. <u>AFTER ADDING ATTRIBUTE PATTERN</u>

   E(Pattern=checked) = $-\frac{1}{2}\ log2\left(\frac{1}{2}\right) - \frac{1}{2}\ log2\left(\frac{1}{2}\right) = 1$

   E(Pattern = striped) = $-\frac{2}{2}\ log2\left(\frac{2}{2}\right) - \frac{0}{2}\ log2\left(\frac{0}{2}\right) = 0$

   E(Pattern = dotted) = $-\frac{0}{2}\ log2\left(\frac{0}{2}\right) - \frac{2}{2}\ log2\left(\frac{2}{2}\right) = 0$

Average Entropy = $\frac{2}{6}\ (1) + \frac{2}{6}\ (0) + \frac{2}{6}\ (0) = 0.33$
Gain(Outlook) = 0.985 – 0.33 = 0.655

7. Recalculation of Color Shape and Size after adding new <u>CHECKED PATTERN</u> attribute

E(color = red) = $-\frac{1}{1}\ log2\left(\frac{1}{1}\right) - 0 = 0$

E(color = green) = $0 - \frac{1}{1}\ log2\left(\frac{1}{1}\right) = 0$

Average Entropy = 0
Gain (Outlook) = 0.985 – 0 = 0.985

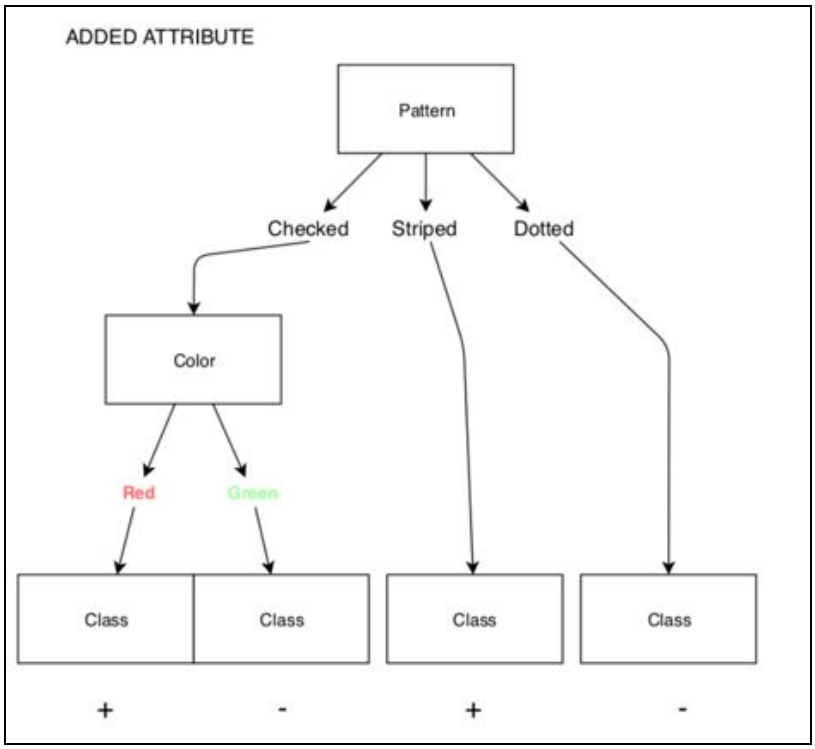
$E(\text{Shape} = \text{square})$ = $-\frac{1}{2}\ log2\left(\frac{1}{2}\right) - \frac{1}{2}log2\left(\frac{1}{2}\right) = 1$

Average entropy = 1
Gain(Outlook) = 0.985 – 1 = - 0.015

E(Size = big) = $-\frac{1}{1}\ log2\left(\frac{1}{1}\right) - 0 = 0$
E(size = small) = $0 - \frac{1}{1}\ log2\left(\frac{1}{1}\right) = 0$

Average Entropy = 0
Gain (Outlook) = 0.985 – 0 = 0.985

8. Decision Tree After Added attribute



PART C – II

- Understand what impact may happen to your created tree, if you later add a new missing attribute after creating the tree?
    - In our original decision tree, the highest gain was the color attribute of 0.54. From our original tree we start with the color attribute to give us the most information on whether the shirt is a (+) or a (-). After adding the pattern attribute the color no longer held the highest gain. The pattern attribute had a gain of 0.655. Thus, the second decision tree started with the pattern attribute. This allowed us to more easily identified if the shirt would be a (+) or a (-).
- What are some of the different possible changes you may expect to see on the classification decision tree you just created?
    - By adding a pattern attribute, it will allow us another pathway to identify if the shirt will be (+) or (-). Having this new attribute will allow us to determine with more confidence (more gain). Moreover, by adding a new attribute we can cross reference the new decision tree against the original decision tree to verify against our results.
- What if a data scientist provided his or her results with high confidence, by missing this attribute altogether?
    - By missing this new attribute altogether, it creates a certain level of ambiguity due to the missing data. For example, if the scientist was missing the size attribute, having the color attribute alone would not be sufficient in determining if the shirt is a (+) or (-). Having the new attribute creates another level of certainty on the results.
- What if his or her results are used for decision making on how many million more shirts to produce for next year?
    - On a much larger scale of how many more million shirts to produce this decision would be catastrophic. If the new added attribute was missing there would be a certain level of uncertainty.
- Do you think the data scientist is elated when he or she discovers the new attribute and gets more reliable results?
    - Yes, as a data scientist with the new added attribute, the scientist can easily determine if the shirt is a (+) or (-) in multiple pathways. Having a new attribute allows the scientist to split at multiple attributes and allow for a greater number of certainties in determining the result.