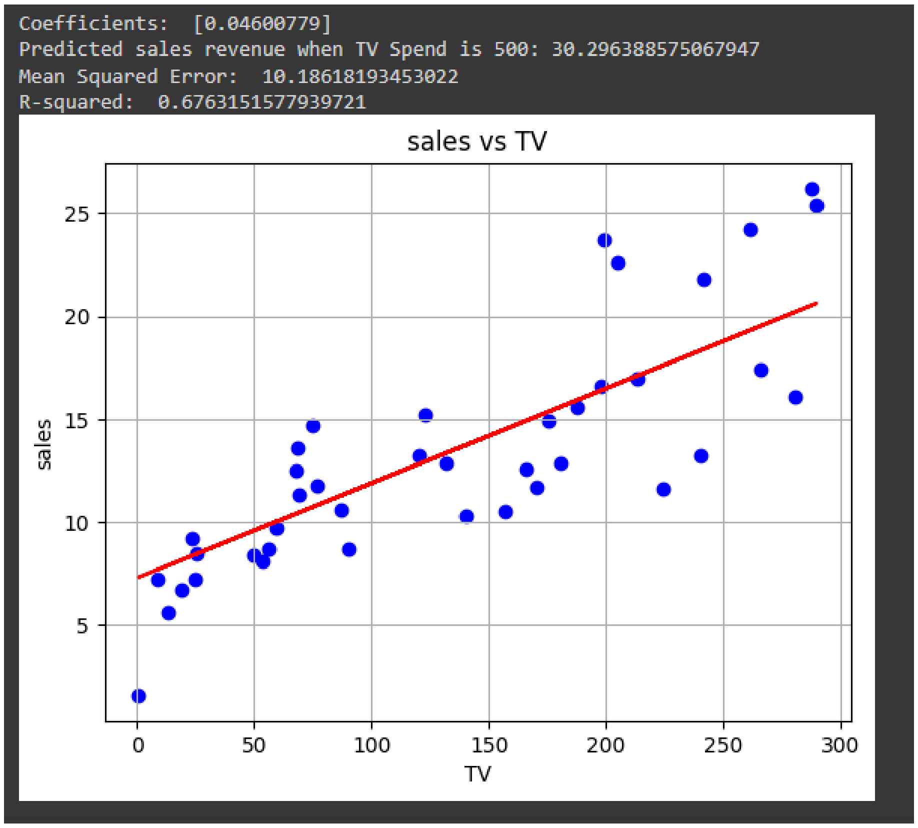
Emerging Technologies 2	**LABORATORY EXAM REPORT**	March 6, 2024
BSCPE32 - S3	Partners: ***Catorce,*** *Mayah Mae* **|** ***Naga,*** *Jamal*	**Engr. Roman Richard**

1.) Singular Linear Regression Model

In creating the Singular Linear Regression model we used to import libraries such **pandas** and **numpy** so that we can load our data in the dataset, also, we used **matplotlib**, **seaborn**, and **sklearn** to visualize the data of our dataset. We chose the dataset named advertising.csv where it has 200 entries with 4 columns such as TV, radio, newspaper, and sales.

We used the heatmap() method in visualizing correlations between variables, where it indicates that sales and TV have a strong positive correlation since these two variables have the highest values than others. After that we select the TV as our independent and sales as dependent variable and we split the data entries into train and test set where the train set has 80% entries in the data set while the test set is 20%. After that, we fit the linear regression to the training set and predict the result of the test set. The result we got in R Squared Score is 0.67 which is approximately 67% and by that we can say that the independent variable (TV) affects the dependent variable {sales) as seen in figure 1.1.



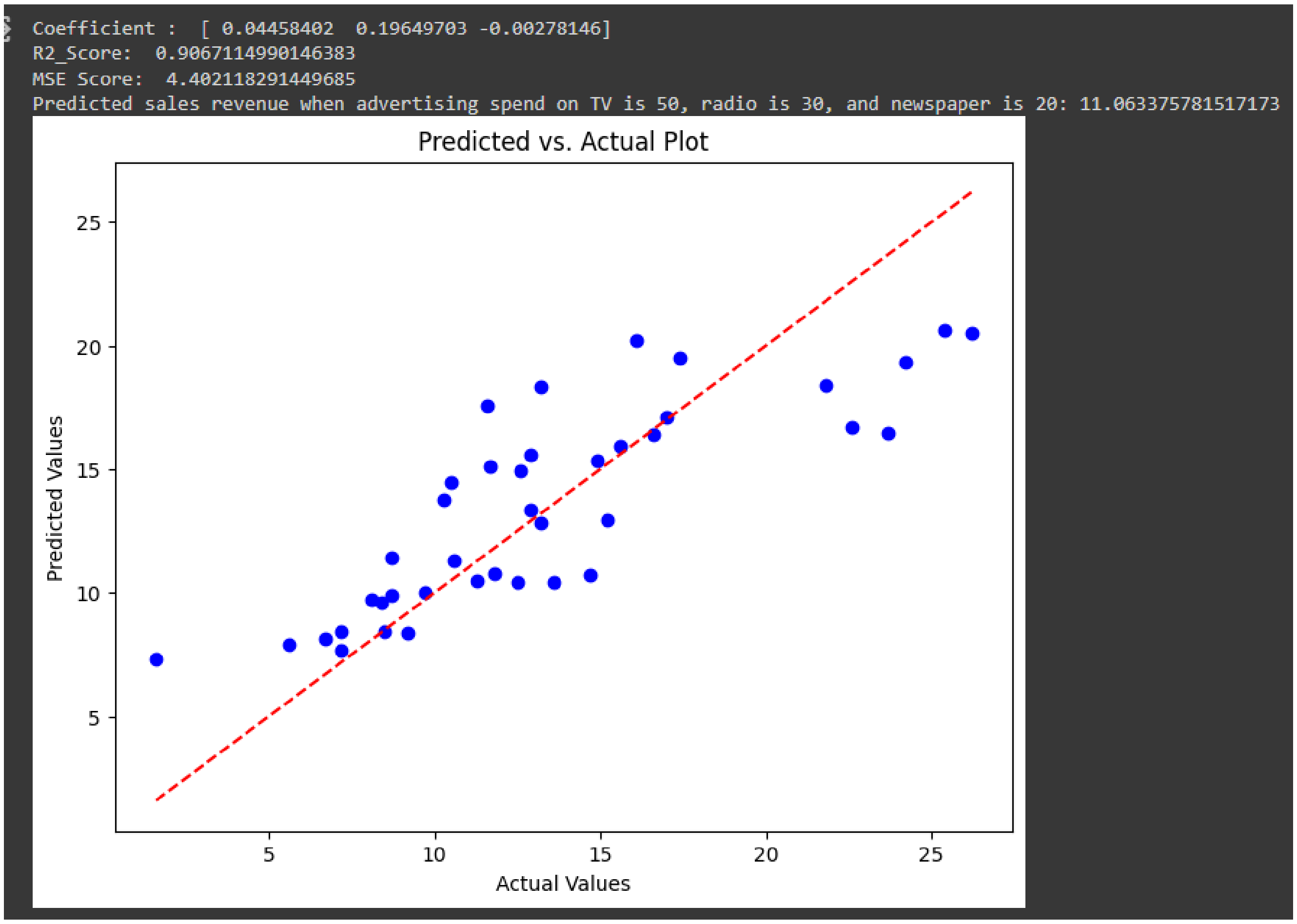
```
Coefficients:  [0.04600779]
Predicted sales revenue when TV Spend is 500: 30.296388575067947
Mean Squared Error:  10.18618193453022
R-squared:  0.6763151577939721
```

Figure 1.1: Singular Linear Regression Model Output

2.) Multiple Linear Regression Model

Same process we've done in the Singular Linear Regression Model we used to import libaries such as **pandas**, **numpy**, **sklearn**, **seaborn**, and **matplotlib**. First, we used the **pairplot()** method so that we can visualize the relationship between Features and Response .

We used 3 features (TV, radio, newspaper) as our independent variables X and 1 feature for dependent variable y (sales) to predict sales revenue based on advertising spending through media such as TV, radio, and newspaper. After that, we split the data entries into a train and test set where the train set gets the 80% and the rest are the test set. We fit the Singular Linear Regression into the train set and we predict the test set result. We got a R squared score of 0.90 which is approximately 90% where we can say that all features in the independent variable X (TV, radio, newspaper) really affect the revenue sales as seen in the figure 1.2.



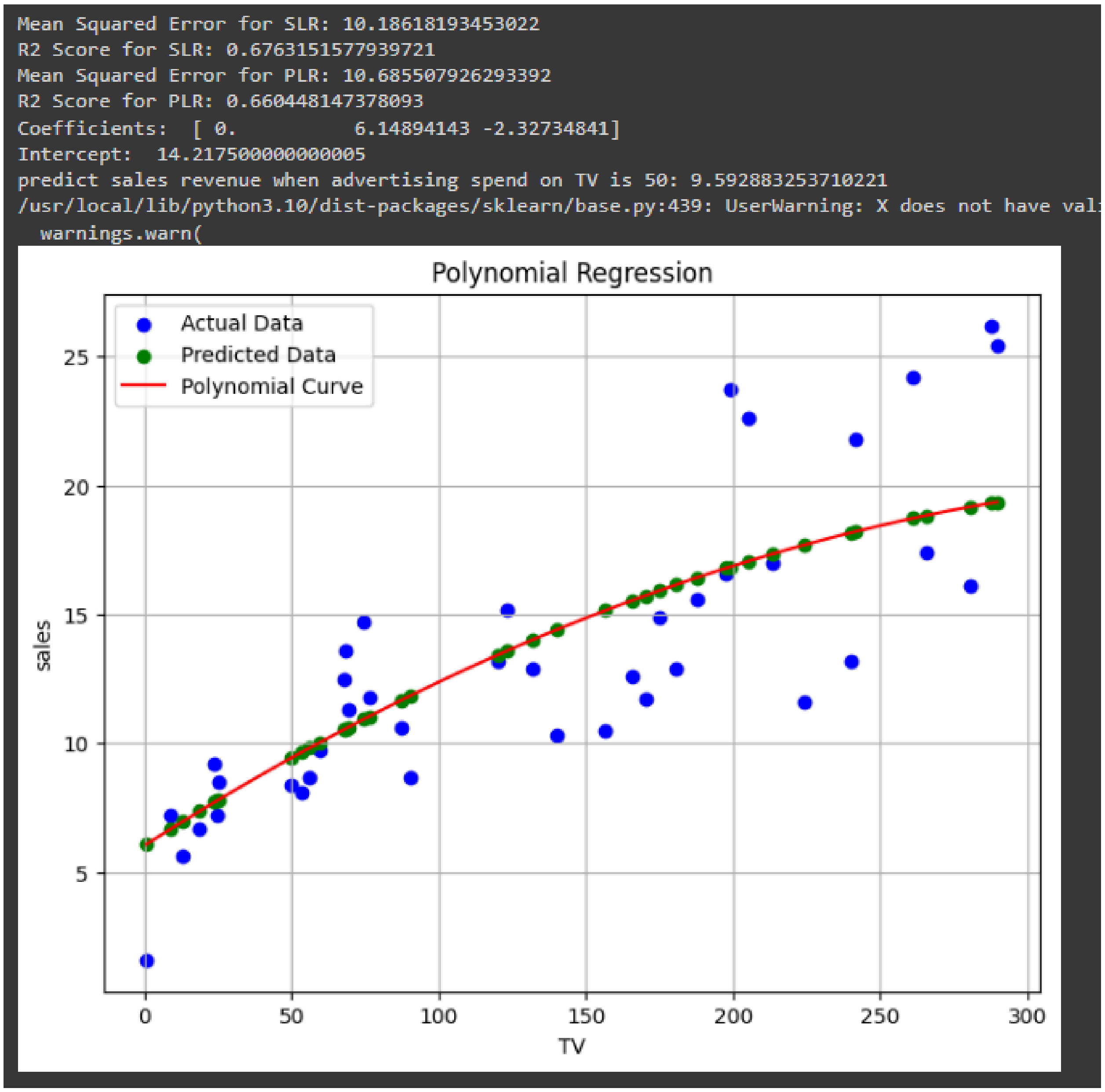
```
Coefficient :  [ 0.04458402  0.19649703 -0.00278146]
R2_Score:  0.9067114990146383
MSE Score:  4.402118291449685
Predicted sales revenue when advertising spend on TV is 50, radio is 30, and newspaper is 20: 11.063375781517173
```

Figure 1.2: Multiple Linear Regression Model Output

3.) Polynomial Regression Model

We still used the same libraries and dataset in getting the Polynomial Regression Model. The independent variable we used here is TV and our dependent variable is still the sales. Same as above we split the values into train and test sets where the train gets 80% and the rest are the test set. In this model we apply the polynomial features and standard scaler where it transforms the original features into polynomial features of degree 2 and then standardizes the scaled polynomial features for both the training and testing sets, ensuring consistent feature scaling across the data. After that, we train the Singular Linear Regression and Polynomial Regression to compare the output of these two models. After that we conclude that the Singular Linear Regression Model is a better model than the Polynomial Regression Model since the R square score of the Singular Regression Model is much higher (0.67 or 67% > 0.66 or 66%). With this result we can say that the Singular Linear Regression Model has better performance in predicting sales revenue based on advertising spending through TV.

Figure 1.3 shows print results and visualization of the Polynomial Regression Model where it allows to capture the non-linear relationships between independent (TV) and dependent (sales) variables. By fitting a polynomial curve to the data it can capture curved patterns and complex relationships.



```
Mean Squared Error for SLR: 10.18618193453022
R2 Score for SLR: 0.6763151577939721
Mean Squared Error for PLR: 10.685507926293392
R2 Score for PLR: 0.660448147378093
Coefficients:  [ 0.          6.14894143 -2.32734841]
Intercept:  14.217500000000005
predict sales revenue when advertising spend on TV is 50: 9.592883253710221
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have val
  warnings.warn(
```

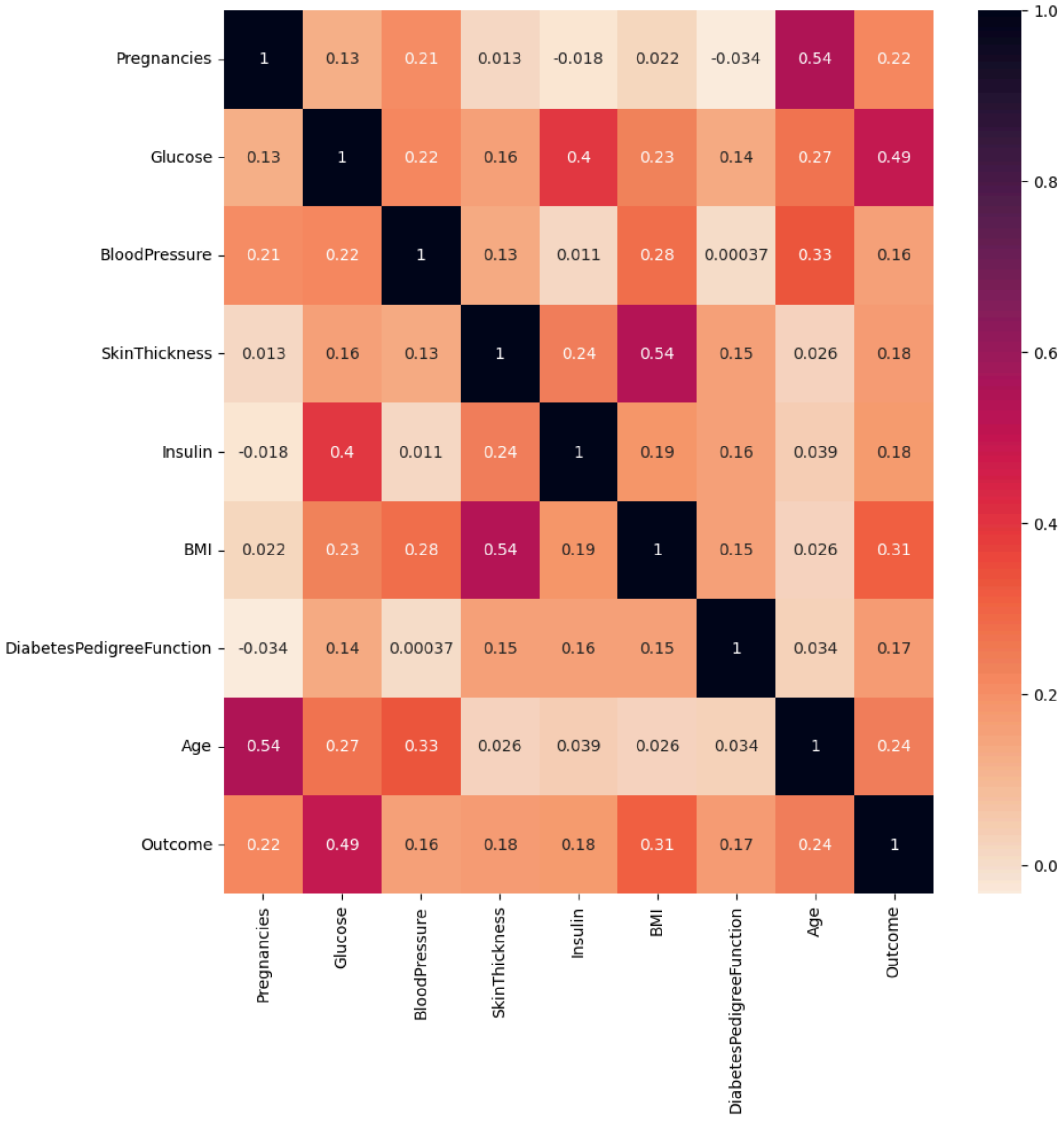
Figure 1.3: Polynomial Regression Model Output

4.) Logistic Regression Model

In implementing the Logistic, Decision Tree and Random forest. We have implemented a generalized data understanding, cleaning, and splitting in the first few cells. With this, we will not be implementing these codes again and again which will lead to much complicated data for analysis. We iterated the libraries that will be used on the first cell and then progressed to the literal implementation of the logistic regression.

First, the implementation of the libraries namely numpy, pandas, matplotlib, seaborn, and sklearn. These are used in showing plot, counting metrics and more. On the first analysis of the data, we have observed that there were columns with zero values which are the features that are not possible to have zero values. That is why we started cleaning the data by checking if there were any null values, or zero values. Luckily we did not find any null values, only the zero values. We therefore proceeded by replacing the zero values with the mean of each features. Then proceeded with the plots and graphs for correlations.

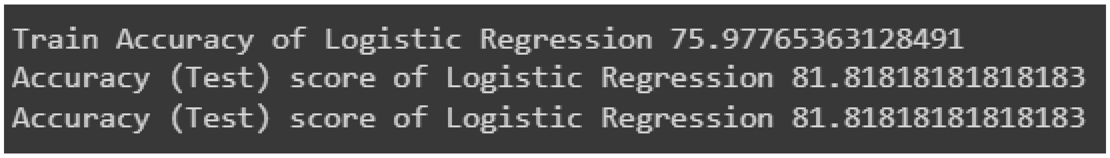
We have observed that there are 4 features that has high correlation with the 'Outcome'. It is the 'Pregnancies', 'Glucose', 'BMI' and 'Age'.



	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
Pregnancies	1	0.13	0.21	0.013	-0.018	0.022	-0.034	0.54	0.22
Glucose	0.13	1	0.22	0.16	0.4	0.23	0.14	0.27	0.49
BloodPressure	0.21	0.22	1	0.13	0.011	0.28	0.00037	0.33	0.16
SkinThickness	0.013	0.16	0.13	1	0.24	0.54	0.15	0.026	0.18
Insulin	-0.018	0.4	0.011	0.24	1	0.19	0.16	0.039	0.18
BMI	0.022	0.23	0.28	0.54	0.19	1	0.15	0.026	0.31
DiabetesPedigreeFunction	-0.034	0.14	0.00037	0.15	0.16	0.15	1	0.034	0.17
Age	0.54	0.27	0.33	0.026	0.039	0.026	0.034	1	0.24
Outcome	0.22	0.49	0.16	0.18	0.18	0.31	0.17	0.24	1

Figure 4.1

We trained the data with these features, having the independent variable y = 'Outcome' and the dependent variables are the remaining features and columns. On the logistic regression it outputted 75.97% on Train Accuracy, while 81% on the Test Accuracy.



```
Train Accuracy of Logistic Regression 75.97765363128491
Accuracy (Test) score of Logistic Regression 81.81818181818183
Accuracy (Test) score of Logistic Regression 81.81818181818183
```

Figure 4.2

Afterwards, we implemented a confusion matrix to describe the performance of the classification model where the true values are known. The output is shown in figure 4.3. This states that there are 141 True Positives and 48 True Negatives, by using this model.

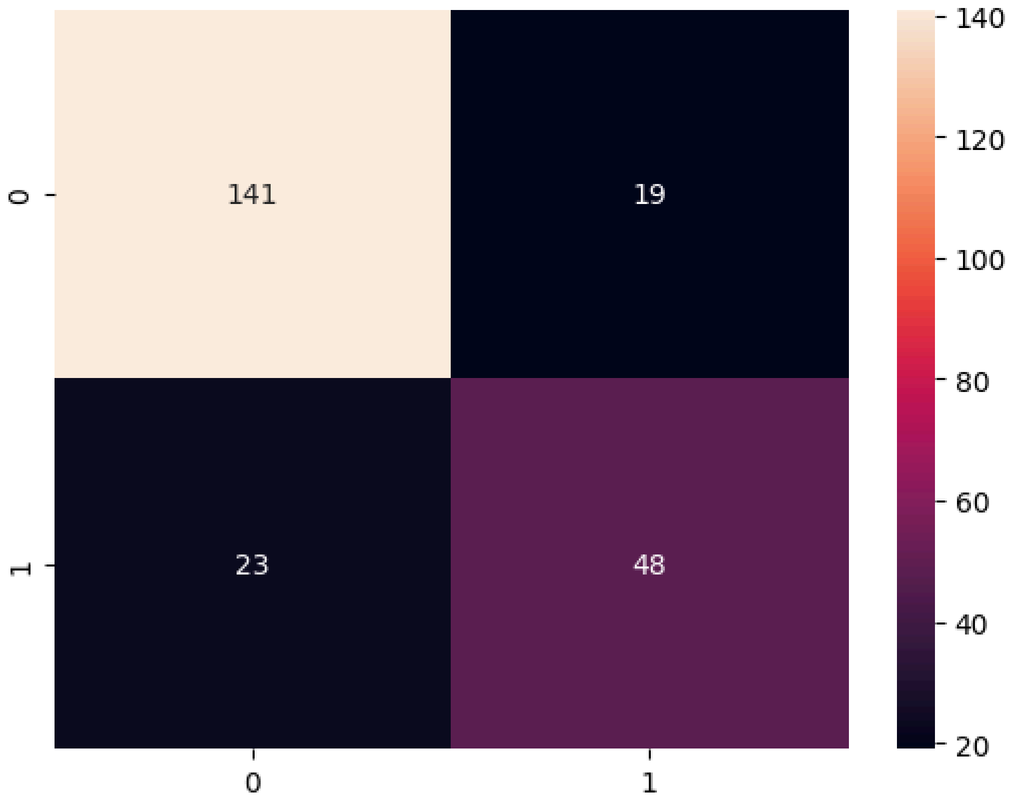
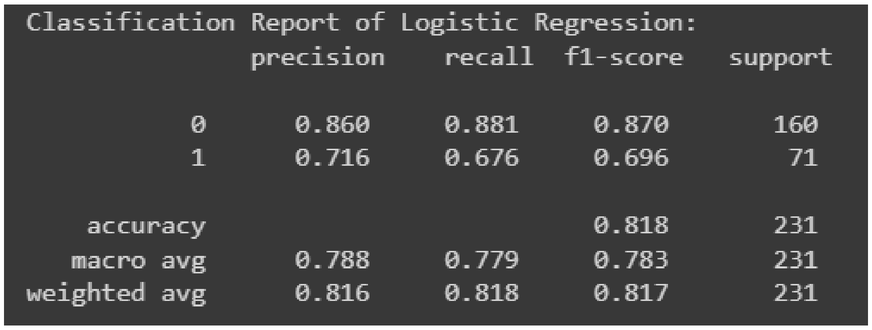


Figure 4.3

Lastly, we implemented a Classification report which printed out the Logistic Regressions classification report on the test set.



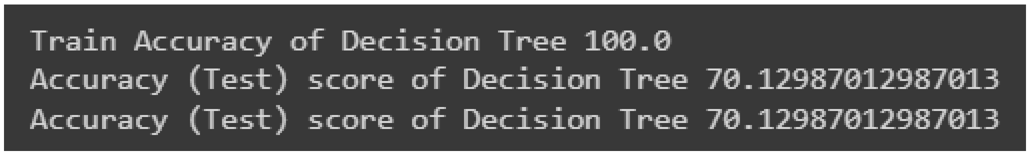
```
Classification Report of Logistic Regression:
              precision    recall  f1-score   support

           0      0.860     0.881     0.870       160
           1      0.716     0.676     0.696        71

    accuracy                          0.818       231
   macro avg      0.788     0.779     0.783       231
weighted avg      0.816     0.818     0.817       231
```

Figure 4.4

This report proves that Logistic regression performs very well by having higher precision, recall, and F1-score for class 0 compared to class 1. But still, this model is having difficulty with its reliance on linear decision boundaries and assumptions of feature independence.

5.) Decision Tree Model

Same as the Logistic Regression, similar dataset was used in this model and also similar dataset understanding, cleaning and splitting were performed. Proceeding with the implementation of the Model Evaluation. It has hit the 100% Train accuracy while 70% on the Test accuracy.



```
Train Accuracy of Decision Tree 100.0
Accuracy (Test) score of Decision Tree 70.12987012987013
Accuracy (Test) score of Decision Tree 70.12987012987013
```

Figure 5.1

We then proceeded with a confusion matrix to describe the performance of the classification model where the true values are known. The output is shown in figure 5.2. This states that there are 123 True Positives and 39 True Negatives, by using this model.

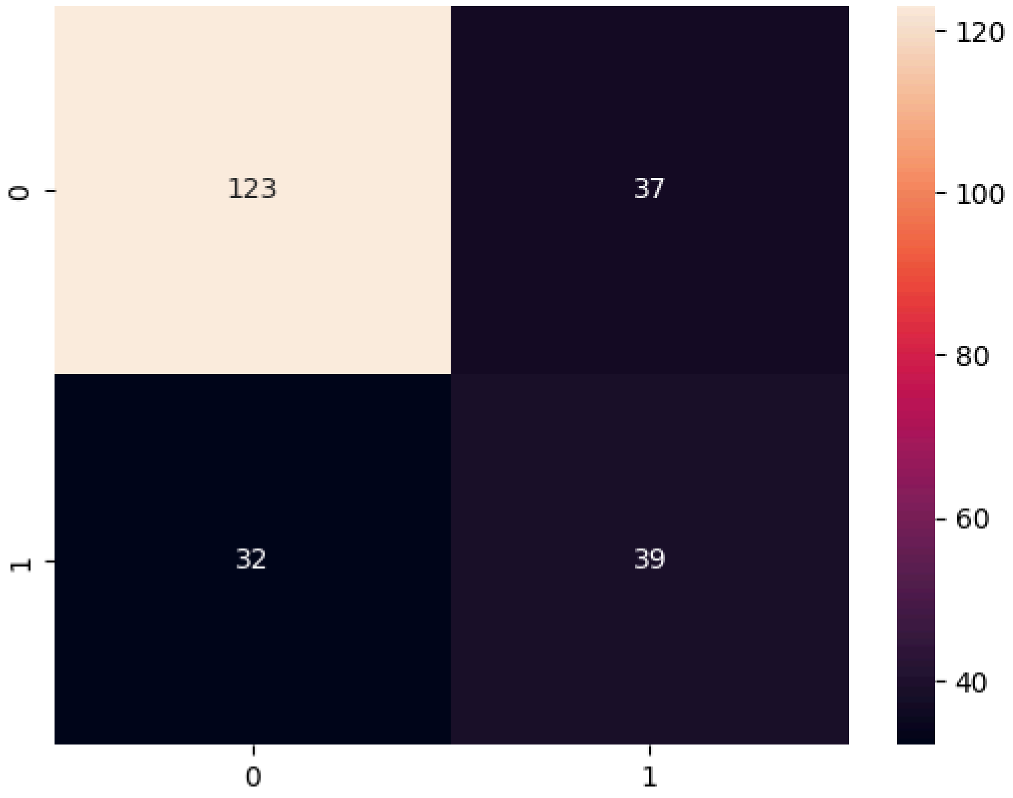


Figure 5.2

By using this Model, we are able to output a Decision tree that picks up every factors that can be deduced in a dataset. It continuously splits up in a way of presenting different decisions and possible outcomes until it reaches the edges where the answers can be observed. This offers interpretability because it generates a tree-like structure where each node represents a decision based on a feature, making it very intuitive to understand the predictive process.

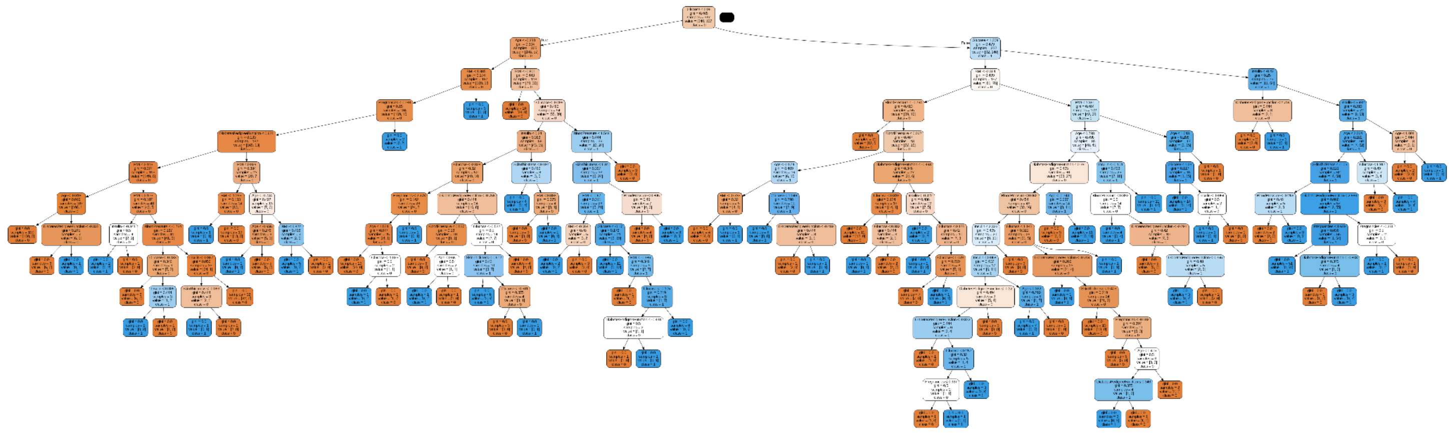
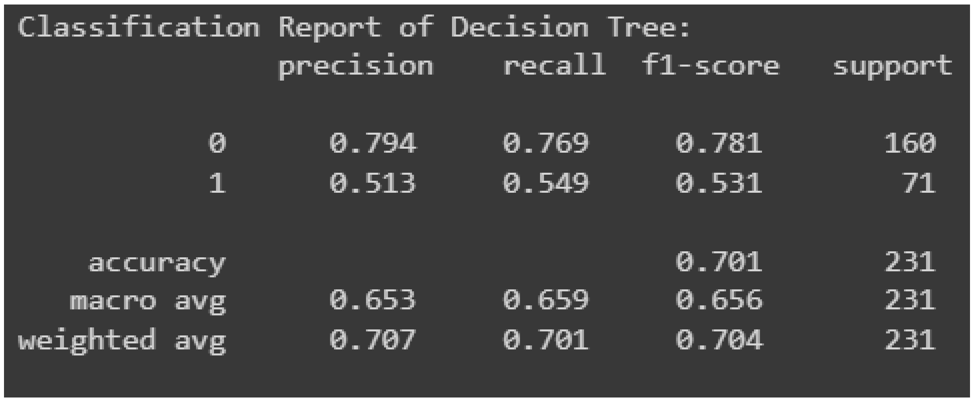


Figure 5.3

On the last part, like on the logistic regression model, we have implemented a classification report for the decision tree.



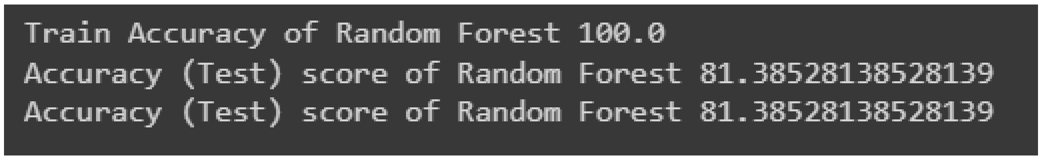
```
Classification Report of Decision Tree:
              precision    recall  f1-score   support

           0      0.794     0.769     0.781       160
           1      0.513     0.549     0.531        71

    accuracy                          0.701       231
   macro avg      0.653     0.659     0.656       231
weighted avg      0.707     0.701     0.704       231
```

Figure 5.4

The classification report indicates that the decision tree model performs less accurately compared to the logistic regression model. Both precision and recall are lower for both classes in the decision tree model, suggesting it has more difficulty correctly classifying instances than the logistic regression model.

6.) Random Forest Model

The same as the Logistic Regression and Decision Tree model, we have implemented similar data training and testing in this model. Shown in the figure below is the Train accuracy of random forest and its Test Accuracy. We can see that it also got 100% Tran Accuracy while 81% Test Accuracy. This implies that the random forest model achieves perfect accuracy on the training data but lower accuracy on new, unseen data, indicating potential overfitting. To address this issue, techniques like cross-validation, hyperparameter tuning, or regularization can help improve the model's ability to generalize to unseen data.



```
Train Accuracy of Random Forest 100.0
Accuracy (Test) score of Random Forest 81.38528138528139
Accuracy (Test) score of Random Forest 81.38528138528139
```

Figure 6.1

Proceeding with the confusion matrix. In Figure 6.2, it is shown that there area 136 True Positives and 52 True Negatives in this model. Which is good because having a high number of true positives and true negatives indicates that the model is making correct predictions for both classes. It's a sign of good performance, especially if compared to false positives and false negatives, which represent instances where the model made incorrect predictions.

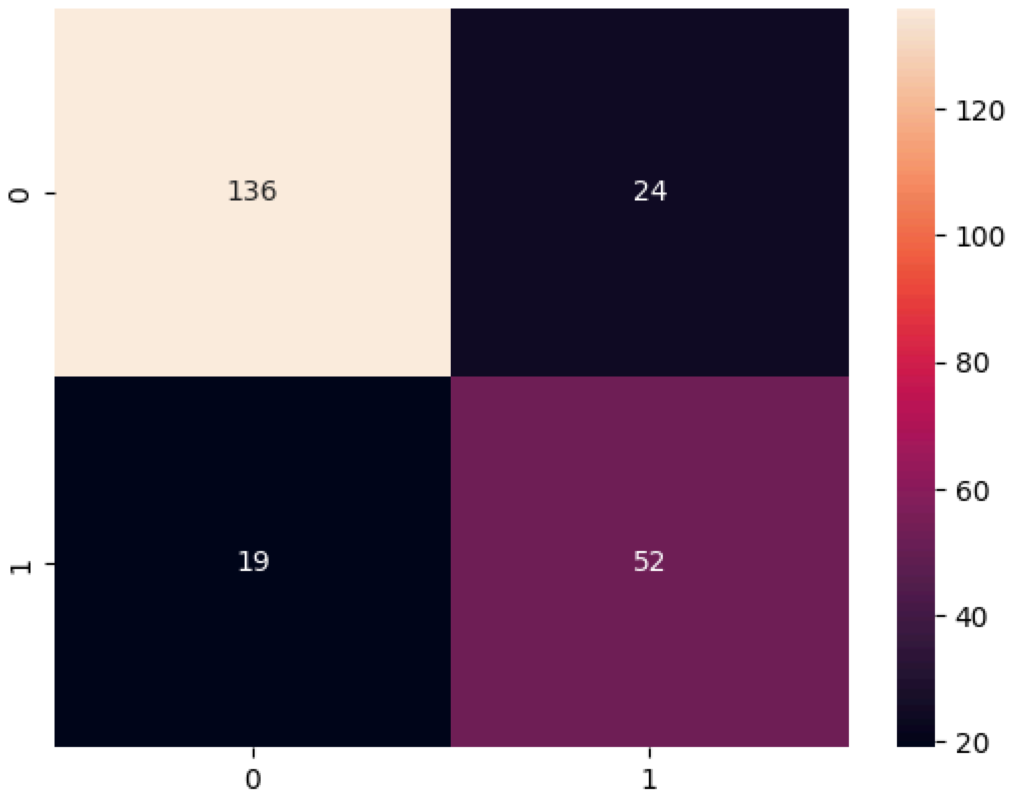


Figure 6.2

Lastly is the Classification report using Random Forest. The report says the random forest classifier works pretty well. It's good at correctly identifying things from both classes, but it's a bit better at one class than the other. But despite the advantages, this regression may still face challenges such as model interpretability and computational complexity.