Predict Cardio Vascular Disease

Janaarthana Harri Palanisamy(015246205)

Fall 2021 CMPE Lab 1 Report, SJSU

This report contains detailed analysis of the given datasets. It includes plots, predictions and comparison of tasks 1,2 and 3 results.

Dataset

1. Cardio-train
    - No. of features: 13
    - No. of instances: 500
2. Cardio-validation
    - No. of features: 13
    - No. of instances: 500
3. Cardio-test
    - No. of features: 12
    - No. of instances: 500
4. Cardio-complete
    - No. of features: 13
    - No. of instances: 1000

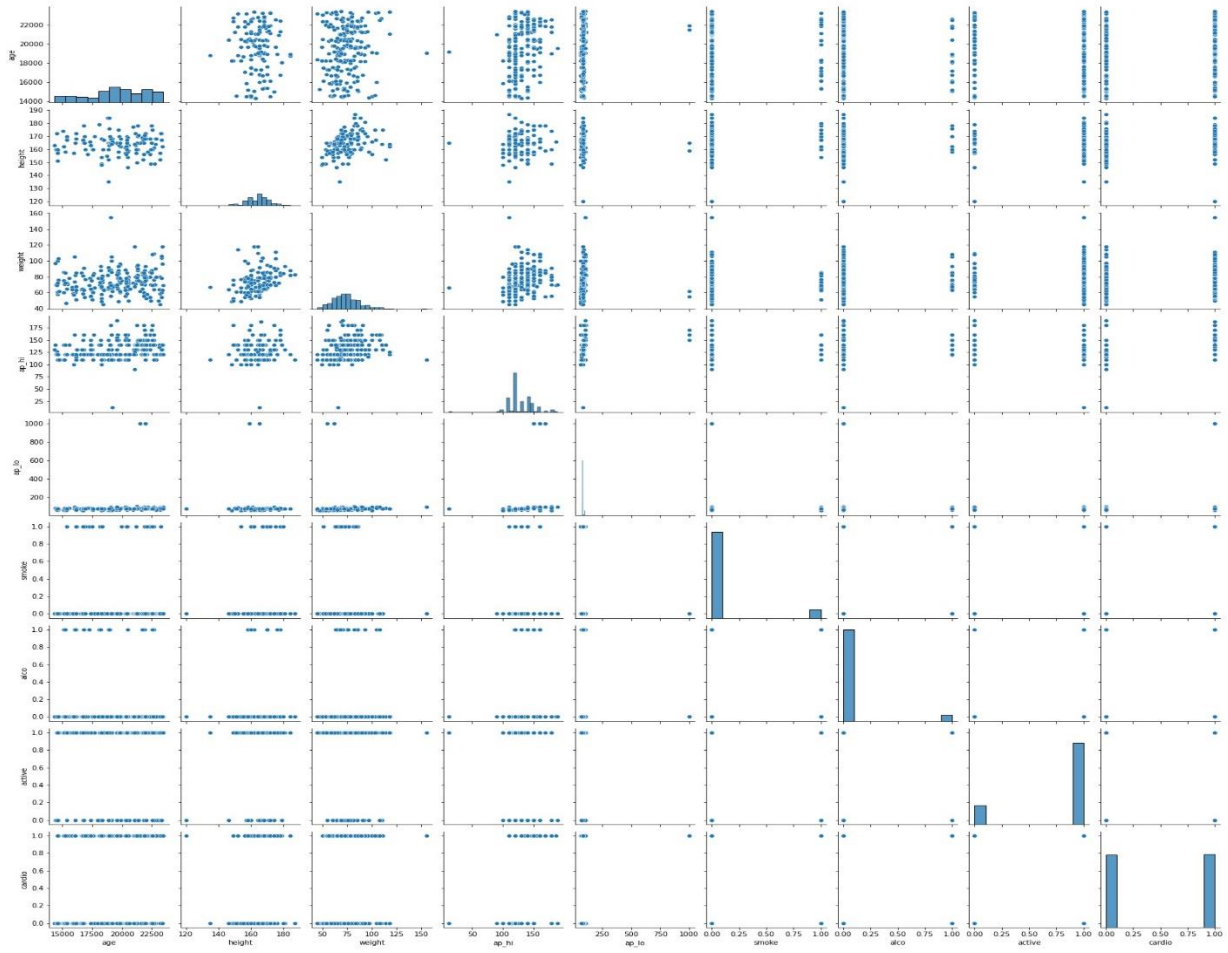
Task 1

Data Analysis

To find relationships between the best features and target variable we can plot correlation matrix with heatmap and scatter matrix (pair plot) which can tell what features are highly correlated with the target.

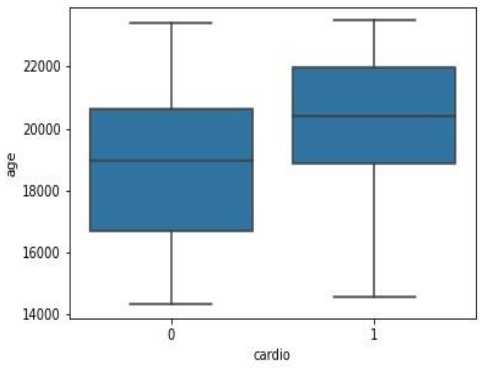
Correlation Matrix:

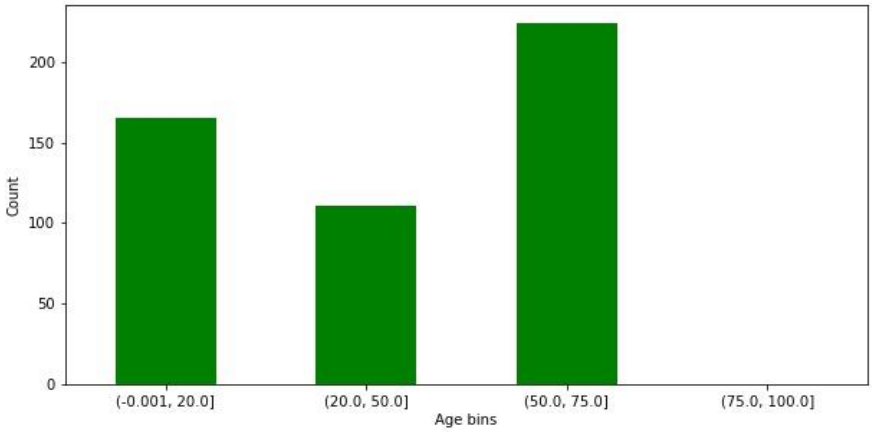
Scatter Matrix:



From the above two plots we can say age (0.28), ap_hi (0.44), weight (0.17) are all positively correlated with the target. With kde, box and bar plots we can analyze the features.

Age:





(Note: For easy understanding I have converted the ages to years)

Observations:

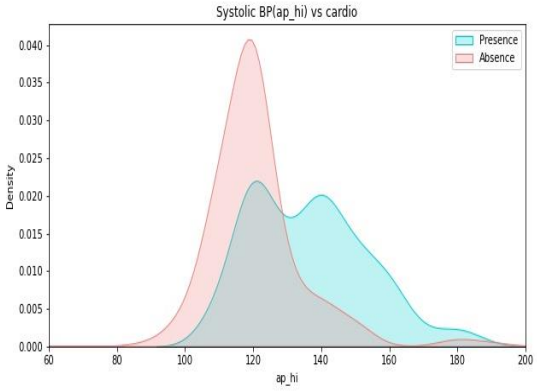
- From kde and box plot we see elderly people are most likely to have cardio disease.
- From age bins we found the split between age groups and their approximate counts.

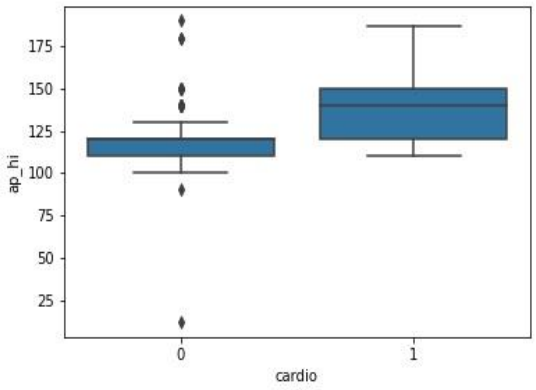
Blood Pressure Stages

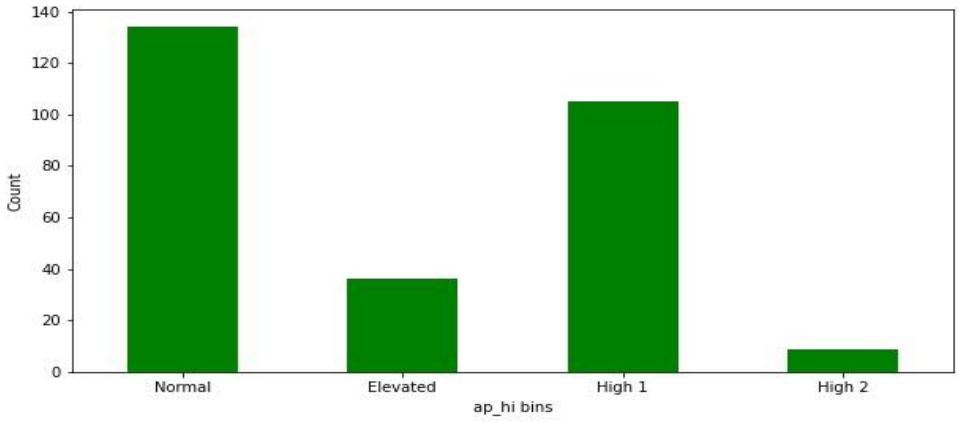
Blood Pressure Category	Systolic mm Hg (upper #)		Diastolic mm Hg (lower #)
Normal	less than 120	and	less than 80
Elevated	120-129	and	less than 80
High Blood Pressure (Hypertension) Stage 1	130-139	or	80-89
High Blood Pressure (Hypertension) Stage 2	140 or higher	or	90 or higher
Hypertensive Crisis (Seek Emergency Care)	higher than 180	and/or	higher than 120

Source: American Heart Association

Ap_hi:



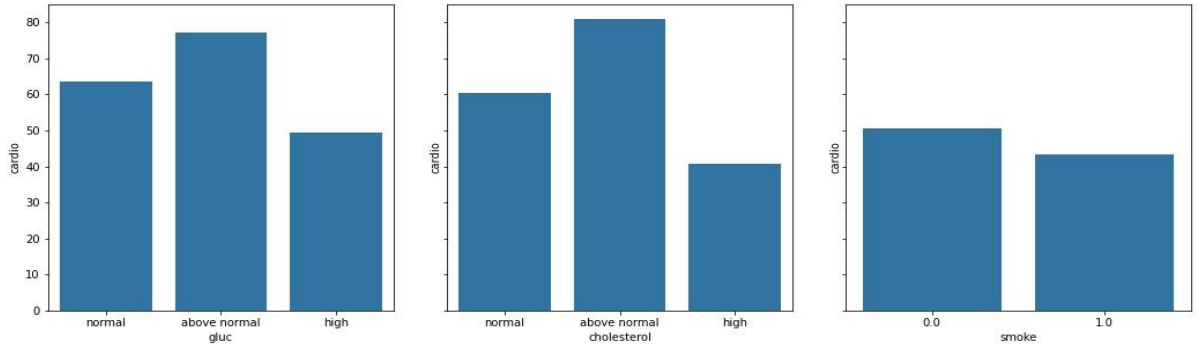


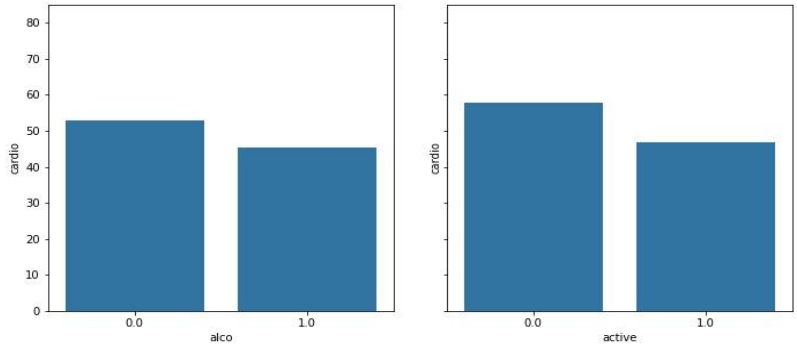


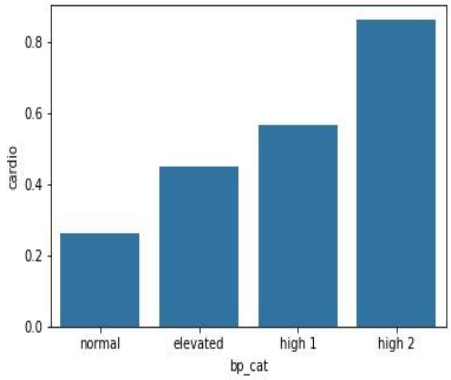
Observation:

- From the plots we can see people with high ap_hi most likely to have cardio disease.
- Bins are arranged with respect to the blood pressure stages.

Other features relations:



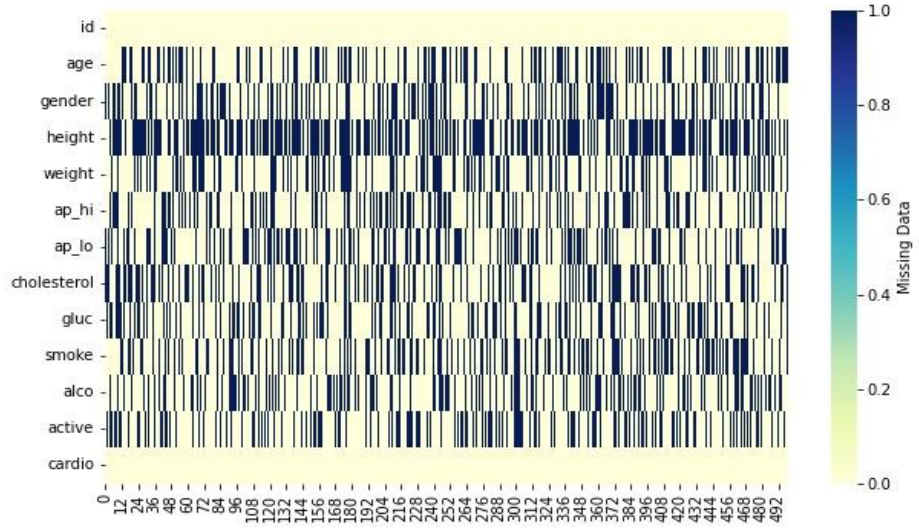


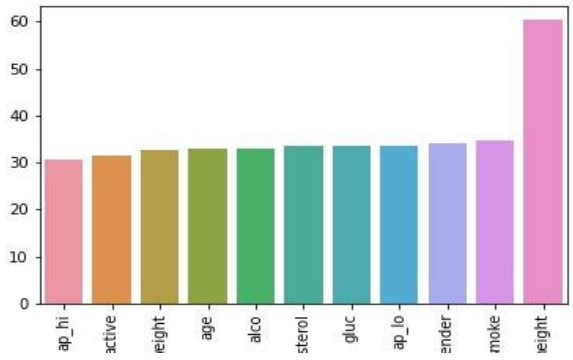


Observations:

- Minor relations found active feature where inactive people might develop cardiovascular disease.
- Bp_cat feature was added for further analysis by combining ap_hi and ap_lo. We see people with high bp are more likely to get cardio disease.
- More than 70% of the people who have above normal glucose and cholesterol levels are more prone to cardiovascular disease.

Missing Data:

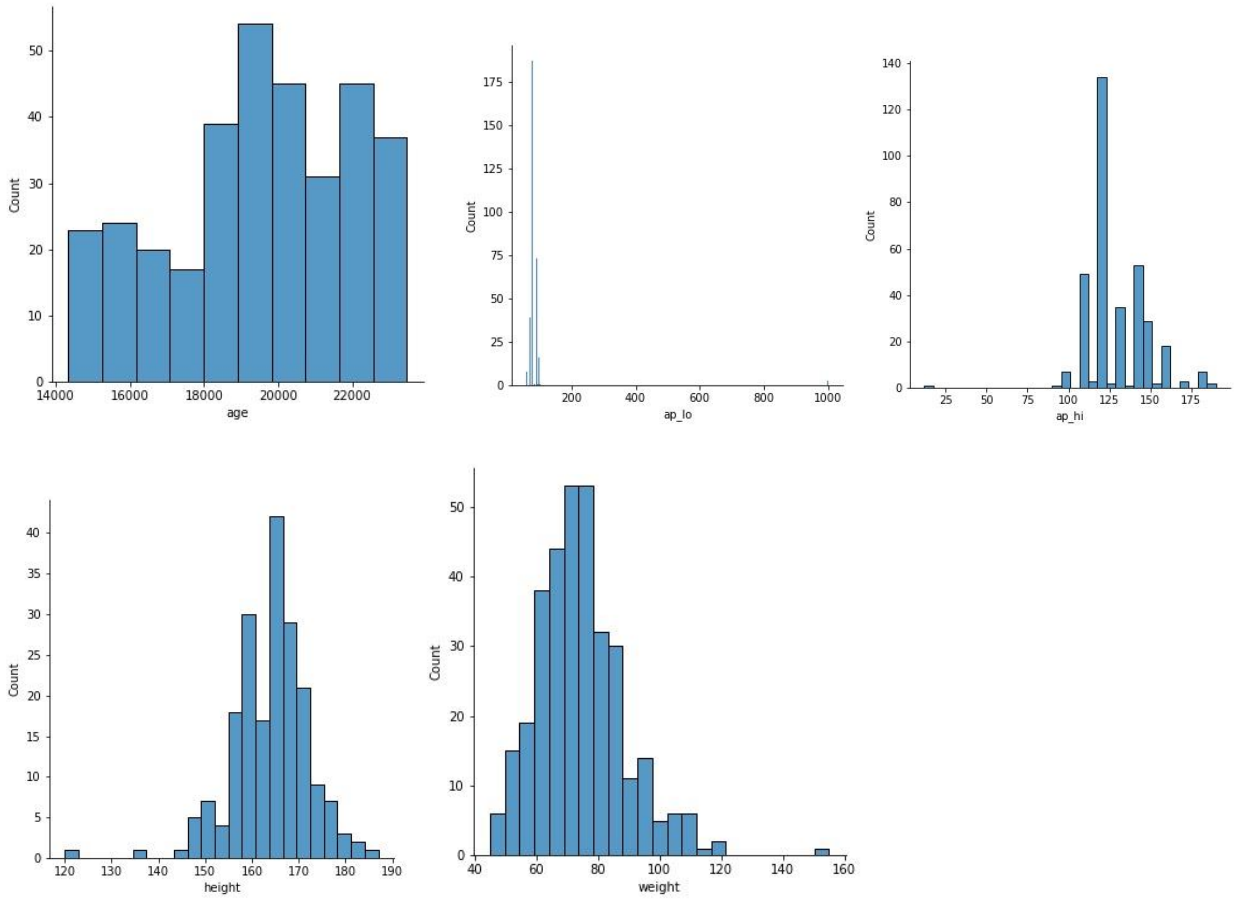




Observations:

- From the heatmap we can see missing data visualization in each variable and the bar plot gives the percent of data.
- Height column has around 60% missing data and other features missing data are on an average of 30%.

Outliers Detection:



Observation:

- From the above displots we can see some data points do not fit in. For example, ap_hi and ap_lo has data points with a value of 15 and 1000 respectively. These are clear outliers and must be eliminated.
- We see the same in height and weight but the gap is not too large and there are people who may fit to those values. So, we can ignore them.
- Age has no outliers.

Filling Missing Values:

Created various datasets with different techniques to handle the missing values. For best results I have concatenated the cardio-validation dataset with cardio-train.

- Dropna_df – Rows which contains missing values are droped.
- Replace_zero_df – filling the numerical features with zero and categorical features with values with min value counts.
- Replace_max_df – filling the numerical features with max values and categorical features with values with max value counts.
- Replace_mean_df – filling the numerical features with mean values and categorical features with values with random values.
- Interpolate_df – Interpolate is a technique which estimates the missing data between two data points. Used built in function to fill values.

Data preparation:

- After handling outliers and missing values, we have the preprocess the dataset.
- Converted the categorical features to numerical using label encoder.
- For better analysis and results split the gender column to male and female.
- Inserted bmi column by calculating height and weight.
- Inserted bp_cat (blood pressure stages) for understanding the data.
- Dropped the id and cardio columns for training the dataset with ML models.

Model Predictions:

1. Logistic Regression:

Datasets	Train accuracy	Kaggle Accuracy
Dropna_df	74.4	67.2
Replace_zero_df	60.4	70.4
Replace_max_df	62	69.6
Replace_mean_df	70.8	70.4
Interpolate_df	68.5	71.2

Observation:

- The best test accuracy was from replace_zero_df and replace_mean_df but it seems replace_zero_df has high bias since train accuracy is low compared to test accuracy.
- So, the best result is from replace_mean_df with 70.4 accuracy.

2. Support Vector Machines:

Datasets	Train accuracy	Kaggle Accuracy
Dropna_df	74.4	69.6
Replace_zero_df	72.4	70.4
Replace_max_df	68.3	68.0
Replace_mean_df	70.5	72.8
Interpolate_df	68.5	69.6

To find the best parameters for the models, Randomized search CV was implemented.

Observations:

- Every dataset seems to perform well with SVC model.
- No underfitting or overfitting can be seen.
- The best result was from replace_mean_df with 72.8 accuracy.

From logistic regression and SVM, we can confidently say replace_mean_df performs better compared to other datasets. So, we will apply different models to replace_mean_df to get the best results.

3. Ridge Classifier:
    - Train accuracy: 71.6
    - Kaggle accuracy: 72

4. Decision Tree:
    - Train accuracy: 71.6
    - Kaggle accuracy: 67.2

5. Random Forest Classifier:
    - Train accuracy: 78.0
    - Kaggle accuracy: 69.6

Result:

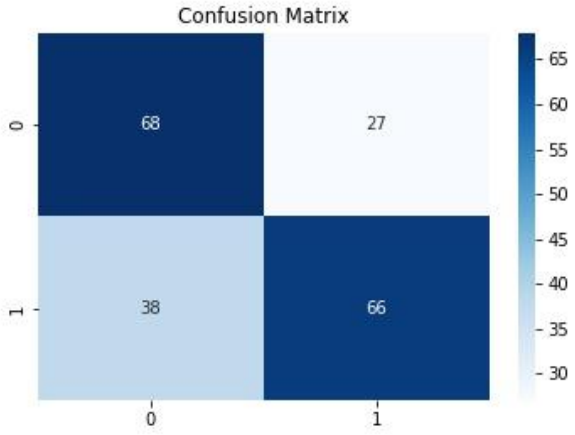
The top two models which has best results are SVM and Ridge classifier with 72.8 and 72 accuracies respectively. So, finally we can conclude SVM as the best model for cardio-train and cardio-validation data.

TASK 2:

Dataset used for this task is cardio-complete. Here we found no missing values. The correlation values of each features had similar relation as that of cardio-train from task 1. So, without any visualizations we applied the similar process from task 1 to obtain the best model. We split the data into train set 80% and test set 20%.

Model Predictions:

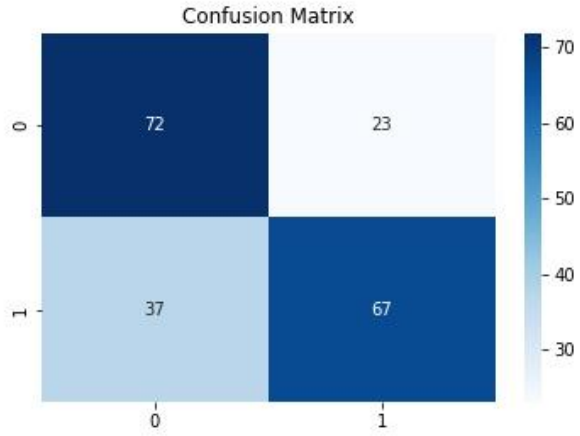
1. Logistic Regression:
    - Train accuracy: 72.6
    - Test accuracy: 67.3
    - Confusion matrix:



    - Classification Report:

	precision	recall	f1-score	support
0	0.64	0.72	0.68	95
1	0.71	0.63	0.67	104
accuracy			0.67	199
macro avg	0.68	0.68	0.67	199
weighted avg	0.68	0.67	0.67	199

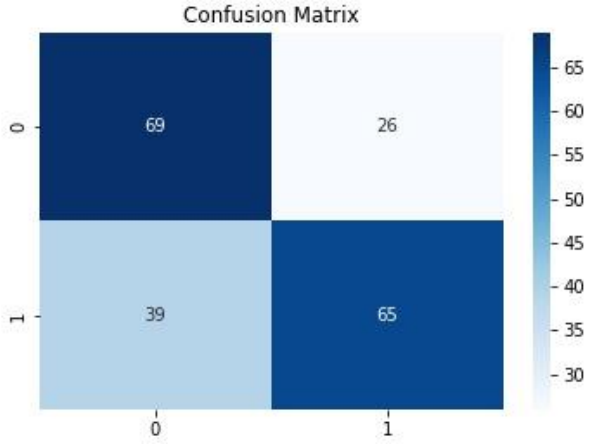
2. SVM:
    - Train accuracy: 72.2
    - Test accuracy: 69.8
    - Confusion matrix:



    - Classification Report:

	precision	recall	f1-score	support
0	0.66	0.76	0.71	95
1	0.74	0.64	0.69	104
accuracy			0.70	199
macro avg	0.70	0.70	0.70	199
weighted avg	0.70	0.70	0.70	199

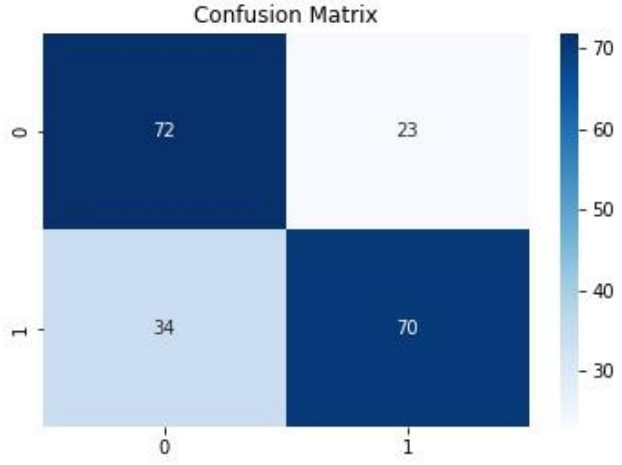
3. Ridge Classifier:
    - Train accuracy: 72.0
    - Test accuracy: 67.3
    - Confusion Matrix:



- Classification Report:

	precision	recall	f1-score	support
0	0.64	0.73	0.68	95
1	0.71	0.62	0.67	104
accuracy			0.67	199
macro avg	0.68	0.68	0.67	199
weighted avg	0.68	0.67	0.67	199

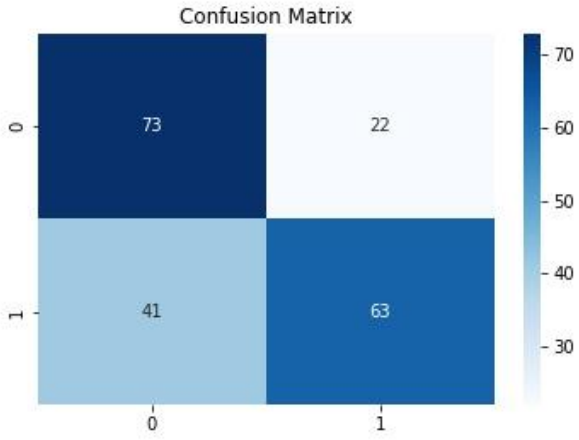
4. Decision Tree:
   - Train accuracy: 73.1
   - Test accuracy: 71.3
   - Confusion Matrix:



- Classification Report:

	precision	recall	f1-score	support
0	0.68	0.76	0.72	95
1	0.75	0.67	0.71	104
accuracy			0.71	199
macro avg	0.72	0.72	0.71	199
weighted avg	0.72	0.71	0.71	199

5. Random Forest:
   - Train accuracy: 73.7
   - Test accuracy: 68.3
   - Confusion Matrix:



   - Classification Report:

	precision	recall	f1-score	support
0	0.64	0.77	0.70	95
1	0.74	0.61	0.67	104
accuracy			0.68	199
macro avg	0.69	0.69	0.68	199
weighted avg	0.69	0.68	0.68	199

Result:

We have tried all the models used in task 1 and came up with best model w hich is decision tree with overall accuracy of 71.3. In task 1 the best result we obtai ned was from SVM model whereas, for the similar dataset the best model in task 2 was decision tree.

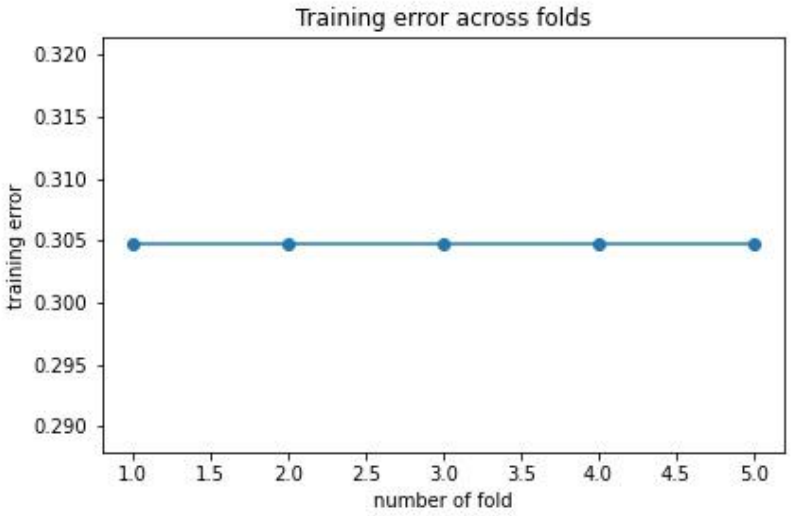
Comparing precision recall and f1-score for SVM and Decision Tree in tas k 2 models. Decision tree model precision was slightly better compared to SVM m odel in both target values (0 and 1). Thought both model's recall of target value 0 was the same decision tree's performance was better for the target value 1. Also, w eighed average f1-score was slightly better with 0.01 percent in decision tree.

As a result, both the models performed well but decision tree had the slight edge and gave the best result for task 2.

TASK 3

In task 3, we are supposed to apply polynomial regression for the features in task 1 data. When SVM model was transformed into polynomial features of degr ee 2 the no. of features in the training set sky rocketed to 105. Then when trained w ith the model, the training accuracy was 99%. When tested with Kaggle accuracy o f 56.7, it clearly showed overfitting. Also we used logistic regression model which had train and Kaggle/test accuracy of 70.5 and 68.8 respectively.

For further classification we used cross validation, in sample error and cros s validation error plot which ultimately showed the model was overfitting. The belo w plot shows mean in sample error for kfold validation set predictions and training set predictions. As every kfold sets predictions was same we got a straight line for t he in same error of the training data.



Comparison of Task 1,2 and 3 results:

Model prediction accuracy for all three tasks is:

- Task 1 – SVM model with 72.8 accuracy
- Task 2 – Decision tree with 71.2 accuracy
- Task 3 – Polynomial Regression of logistic regression with accuracy 68.8

In task1 there were many missing data and it had to filled for model prediction , SVM model's result satisfiable. In task2 we just had to do model predictions and d ecision tree gave the best result. SVM also performed well but decision tree had sli ght edge over SVM. In task3 we used polynomial regression for the task1 data, unf ortunately the data had overfitting issues after transformation.

In task 1 I have also implemented my own logistic regression algorithm using numpy, its results are Train accuracy: 62.5, Test/Kaggle accuracy: 57.6.