Capstone Project: Predicting Heart Disease using Classification Algorithm

• Definition

1.1: Project Overview

Heart disease is not just the number one cause of death in Malaysia, it was also the number one cause of death worldwide. It affects both male and female. Everyone knows someone who had a heart issue. Lots of research had been invested into the possible causes of a heart disease. Machine learning could contribute in helping to reduce this number one killer.

Drivendata has hosted a competition in predicting heart disease using Machine Learning. Participant has to submit a submission file and they will be rated based on the submissions.

Reference:-

- Cardiovascular disease Global Facts and Figures: world-heart-federation
- Why cardiovascular disease is the leading cause of death in Malaysia : star2.com
- Heart Disease Diagnosis and Prediction Using Machine Learning and Data Mining Techniques: A Review ripublication.com
- Dataset source: drivendata
- Competition: drivendata Machine Learning with a Heart Competition

1.2: Problem statement

This project will take the dataset, train them using machine learning algorithm and predict the probability of a patient having a heart disease. This is a classification problem since a label has been given in the dataset for training purpose:-

0: Heart disease is not present

1: Heart disease present

1.3: Evaluation metrics

The aim for this model is to get lower score than the benchmark, as the result will then be evaluated using log-loss function. This function used the probability of class prediction and the true class labels to generate a number. A number that is closer to zero is deemed to have a better model, while zero is a number to strive for the perfect model.

- ## Analysis

2.1: Data Exploration

2.1.1: Dataset and Inputs

This dataset is provided courtesy of Cleveland Heart Disease Database from a competition organised by drivendata. The dataset collects various measurements on health and cardiovascular statistics. Patient's identities are anonymous.

The dataset has been split into training data and test data. Training data has 180 values, while the test data has 90 values. There are 14 columns in the training dataset, where patient_id serves as an identifier. Below are the attributes information of the remaining 13 columns.

-- 1. age

-- 2. sex

-- 3. chest pain type (4 values)

-- 4. resting blood pressure

-- 5. serum cholestoral in mg/dl

-- 6. fasting blood sugar > 120 mg/dl

-- 7. resting EKG results (values 0,1,2)

-- 8. maximum heart rate achieved

-- 9. exercise induced angina

-- 10. oldpeak = ST depression induced by exercise relative to rest

-- 11. the slope of the peak exercise ST segment

-- 12. number of major vessels (0-3) colored by flourosopy

-- 13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

--*labels: 1: Heart disease present. 0: Heart disease is not present

There are no missing values in this dataset. The training data has 100 patients with no heart disease and 80 patients with heart disease. There is a slight imbalance in the dataset, as it contains higher number of patients without heart disease.

2.1.2: Statistical Summary

Below are the statistical summary for the training dataset:-

	slope_of_peak_exercise_st_segment	resting_blood_pressure	chest_pain_type	num_major_vessels	fasting_blood_sugar_gt_120_mg_per_dl	resting_ekg_results	serum_cholesterol_mg_per_dl	oldpeak_eq_st_depression	sex	age	max_heart_rate_achieved	exercise_induced_angina
count	180	180	180	180	180	180	180	180	180	180	180	180
mean	1.55	131.3111	3.155556	0.694444	0.161111	1.05	249.2111	1.01	0.688889	54.81111	149.4833	0.316667
std	0.618838	17.01044	0.938454	0.969347	0.368659	0.998742	52.71797	1.121357	0.464239	9.334737	22.06351	0.466474
min	1	94	1	0	0	0	126	0	0	29	96	0
25%	1	120	3	0	0	0	213.75	0	0	48	132	0
50%	1	130	3	0	0	2	245.5	0.8	1	55	152	0
75%	2	140	4	1	0	2	281.25	1.6	1	62	166.25	1
max	3	180	4	3	1	2	564	6.2	1	77	202	1

Figure 1: Statistical Summary

An outlier has been detected under serum_cholesterol_mg_per_dl. The max value is 564, and this value is almost double than the mean value of 249.

2.1.3: Outlier

	slope_of_peak_exercise_st_segment	thal	resting_blood_pressure	chest_pain_type	num_major_vessels	fasting_blood_sugar_gt_120_mg_per_dl	resting_ekg_results	serum_cholesterol_mg_per_dl	oldpeak_eq_st_depression	sex	age	max_heart_rate_achieved	exercise_induced_angina
patient_id													
rv6siv	2	reversible_defect	115	3	0	0	2	564	1.6	0	67	160	0

Figure 2: serum_cholesterol_mg_per_dl outlier details.

The cholesterol level is on high end, especially any value exceeded 240mg/dl is considered 'very high'. However, the highest cholesterol level ever recorded is 3165 mg/dl, Guiness Book of Record, and that value is definitely an outlier. With this information, I shall keep this outlier in the dataset as I accept this result as a genuine result.

2.2: Exploration Visualisation

A correlation heatmap has been created because I wanted to see possible linear relationship between variables. A new feature called ‘age-range’ has been created, and thal value has been encoded into 0 and 1. Now, there are 18 features in the dataset.

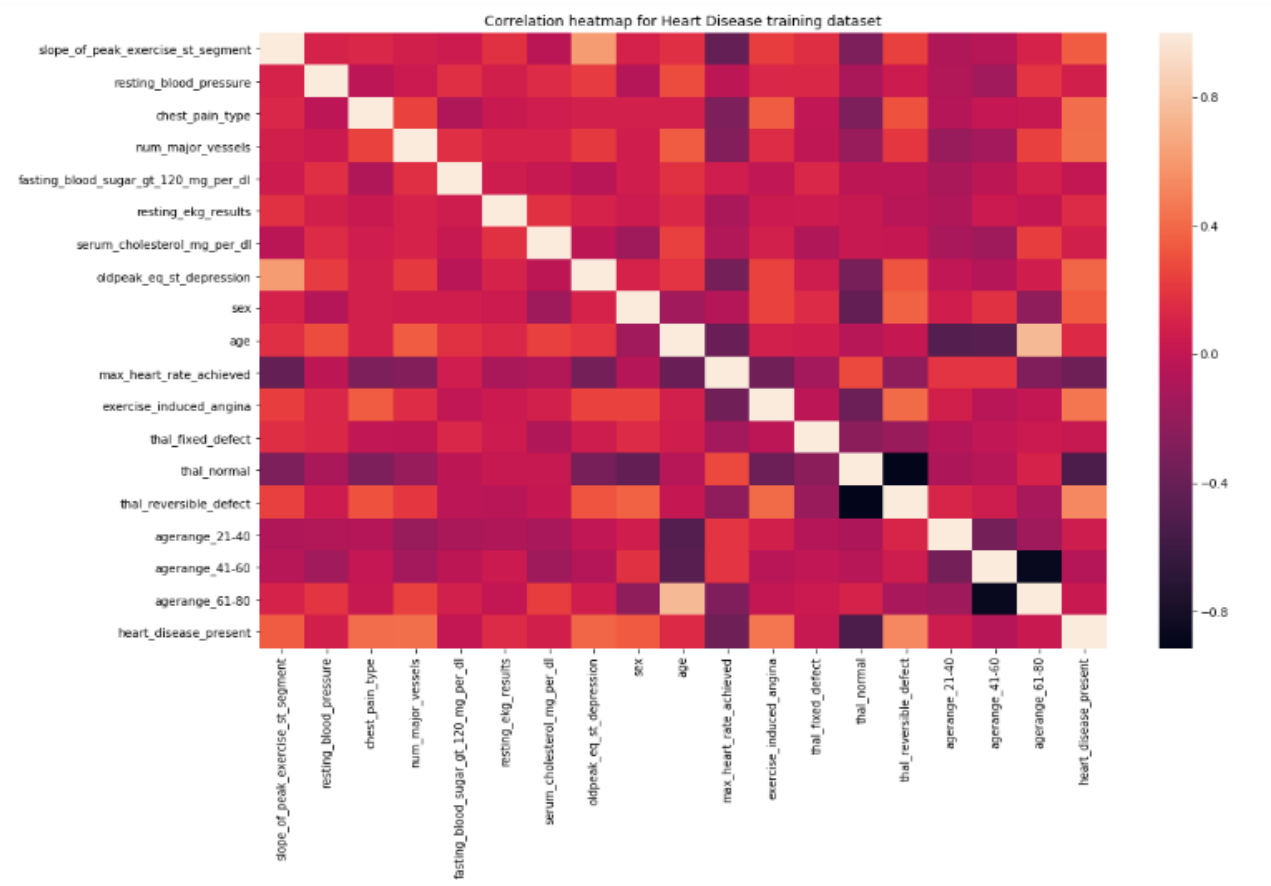


Figure 3: Correlation heatmap for Heart Disease training dataset.

Some observations:-

- Thal_normal and thal_reversible defect has an almost perfect negative linear relationship at -0.913417.
- Age-range 41-60 and age-range 61-80 also almost have a perfect linear relation at -0.869711.
- Some features have a good correlation with heart_disease_present attribute. Below are the Top 5 features with the highest Pearson's correlation score with heart_disease_present:-
    - thal_normal : -0.528811509654
    - thal_reversible_defect : 0.525145374593
    - exercise_induced_angina : 0.448646516812
    - num_major_vessels : 0.421518626048
    - chest_pain_type : 0.412828625366

It seems like the features are not independent of each other, as some linear relationship can be detected amongst the features, as shown in the above correlation findings.

2.3: Algorithm and Techniques

This is a classification problem, so we need to tackle this problem with some classifiction algorithm. The algorithms that were selected are:-

- K-Nearest Neighbors (KNN)
- Decision Tree
- Logistic Regression
- Adaboost.
- Stacking algorithm (by combining KNN, Logistic Regression and Adaboost)

Even though Decision Tree is not normally used for classification task, I would like to include this algorithm as a base to see how well the classification algorithm such as KNN, Logistic Regression, Adaboost is performing vs non-classification algorithms. Stacking algorithm is also introduced to the training as this allows us to combine a few of these algorithms.

KNN was chosen because this models works well with a small dataset. Logistic Regression was chosen because the benchmark is also using Logistic Regression, and I want to see if I am able to improve the benchmark using the same algorithm. And Adaboost is selected because our dataset is pretty clean, and this algorithm performed well on cleaned dataset.

I chose StratifiedKfold as cross-validation algorithm as my labels dataset is slightly imbalance, favouring the non-heart disease patients. For parameters tuning, I selected GridSearchCV algorithm.

2.4: Benchmarks

The benchmark for this problem is 0.5381, using Logistic Regression, as it was one of the most-common default implementation. This benchmark also ensures your model has performed better than chance, as you only have 2 outcomes- either heart disease present or not.

Reference: [drivendata: Benchmark score]

3. Methodology

3.1: Data pre-processing

A new feature called 'age-range' has been created into the train and test database, as I feel age feature we have right now is too specific and our data does not have every single age.

One-hot enconding is being implemented for age-range and thal features. Once it's applied, the features have grown from 13 to 18.

Lastly, minmax scaler is being applied to the dataset, as some of the values like serum_cholesterol and max_heart_rate has three digits numbers while features such as sex is only denoted in 0s and 1s.

3.1: Implementation

Once the pre-processing is done, it's time to implement the models. Below are the steps:-

1. Use StratifiedKfold to split the training data into training and validation (random_state=11).

2. Run the classifier and record the result.

3. Introduce parameters tuning by using GridSearchCV

4. Repeat step 2.

Results:-

Without Features selection:

Classifier	Default Parameter	Parameters Tuning
Decision Tree	9.210	8.854
----------	----------	----------
K-Nearest Neighbour	1.494	4.043
----------	----------	----------
Logistic Regression	0.386	0.438
----------	----------	----------
Adaboost	0.631	0.692
----------	----------	----------
Stacking	0.453	0.489

Figure 4: Log-loss score for both default parameters and tuned parameters without Feature Selection.

3.2: Refinement

I decided to introduced feature selections into the model since it could help in improving the model. I used the RFE model and I chose LogisticRegression as the classifier.

The top 5 features are:-

- 'chest_pain_type',
- 'num_major_vessels'
- 'oldpeak_eq_st_depression'
- 'max_heart_rate_achieved'
- 'thal_normal'

thal_normal, max_heart_rate_achieved and num_major_vessels have also appeared in Top 5 of the Pearson's correlation with Heart Disease feature.

The selected features were being trained again and this time the results are being recorded and compared.

With Features selection:-

Classifier	Default Parameter	Parameters Tuning
Decision Tree	8.826	8.06
----------	----------	----------
K-Nearest Neighbour	2.56	2.549
----------	----------	----------
Logistic Regression	0.422	0.552
----------	----------	----------
Adaboost	0.648	0.692
----------	----------	----------
Stacking	0.464	0.486

Figure 5: Log-loss score for both default parameters and tuned parameters with Feature Selection (using RFE model)

The best performing model is the Logistic Regression with its default parameters. As expected, Decision Tree performed the worst.

Below are the models that performed better than the benchmark (0.5381):-

- Logistic Regression default parameters without Features Selection: 0.394
- Logistic Regression with parameters tuning without Features Selection: 0.438
- Stacking model with default parameters without Features Selection: 0.453
- Stacking model with default parameters without Features Selection: 0.489

- Logistic Regression default parameters with Features Selection (RFE): 0.422
- Stacking model with default parameters with Features Selection (RFE): 0.464

Overall, the models performed worse when we introduced Features Selection into the training dataset.

4. Results

4.1: Model Evaluation and Validation

Below are the default parameters for LogisticRegression()

```
Out[88]: LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                            intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                            penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
                            verbose=0, warm_start=False)
```

Figure 6: Default Parameters for LogisticRegression() classifier

```
Out[122]: LogisticRegression(C=0.5, class_weight=None, dual=False, fit_intercept=True,
                             intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
                             penalty='l2', random_state=None, solver='liblinear', tol=1.0,
                             verbose=0, warm_start=False)
```

Figure 7: LogisticRegression() classifier with parameters tuned.

Logistic Regression actually performed worse with parameters tuning than default parameters. We can see some differences in these parameters, especially the tol value In default parameters, the value is set at 0.0001 and in the tuned paramters, it was set at 1.0. 'tol' is refering to the tolerance for stopping. It is possible that the classifier with higher tol value actually stops its training early. Hence, the model did not get enough training and the algorithm stops before it can converge.

Logistic Regression is a good model to apply since we are using probability prediction as this model produced probabilistic values.

4.2 Justification

I've submitted the submission.csv to Machine Learning with a Heart competition sponsored by drivendata. My score is:-

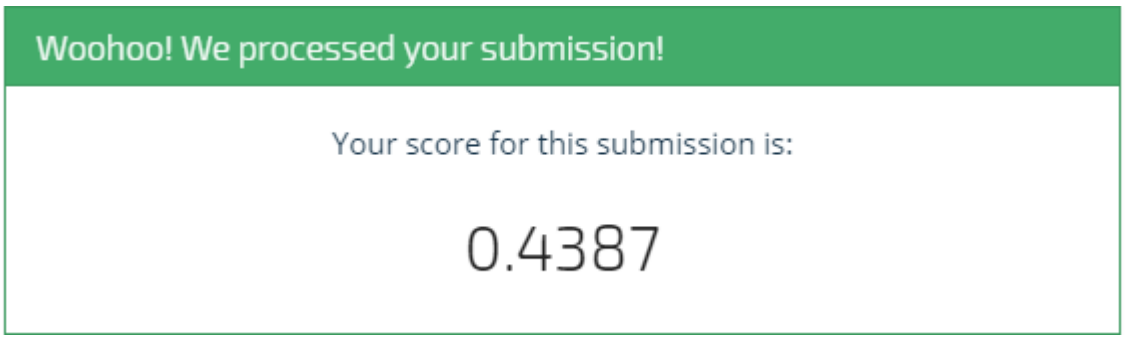
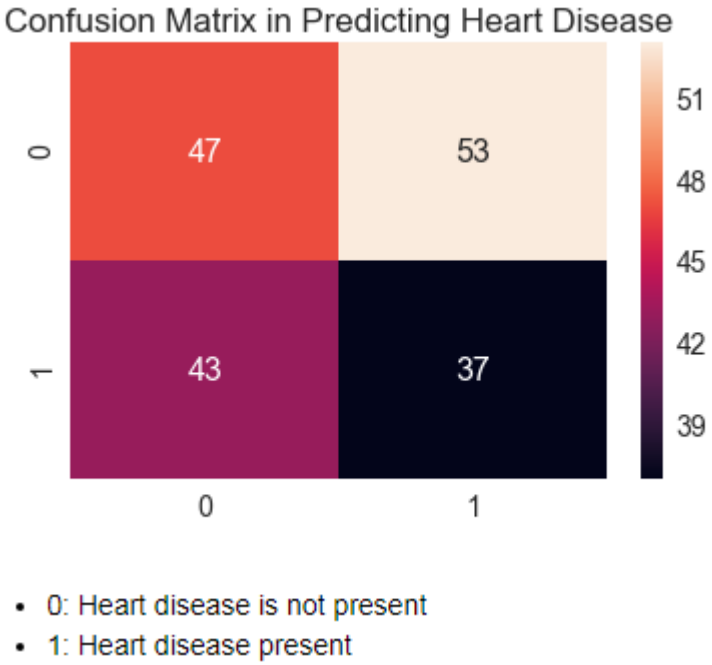


Figure 8: Submission score

The submission score is 0.4387, which is an improvement from the benchmark score of 0.5381. Overall, this model improves the benchmark score by 18.47%. However, it wasn't a significant improvement. This model comes with its own problem which is explained in under Figure 9.

5. Conclusion

5.1: Free-form Visualisation



- 0: Heart disease is not present
- 1: Heart disease present

Figure 9: Confusion Matrix in Predicting Heart Disease

- * 47 true positive prediction, aka, predicted with no heart disease, and they actually has no heart disease
- * 37 true negative prediction, aka, predicted with heart disease, and they actually have no heart disease
- * 53 false positive prediction, aka, we predicted they have a heart disease, but actually, they don't.
- * 43 false negative prediction, where we predict they don't have a heart disease, but actually, they do.

Accuracy, aka the correct predictions: (47 + 37)/180 = 0.467

Precision, aka, who actually has a heart disease? (43 + 37) / 180= 0.444

Recall/Sensitivity, aka the proportion of the patients with heart disease that was diagnosed by the algorithm as having a heart disease: (52+37)/180= 0.528

Overall, based on the confusion matrix above, the model performed quite badly. 43 false negative prediction is a high number to make. If we want to minimise the false negative, the recall value has to be as closed to 100% as possible.

5.1: Reflections

The most difficult part is choosing the right algorithm, as there's a lot of options out there. And not just the right classifiers, but also the right algorithm for Feature Selection and Parameters Tuning.

Also, I am being restricted by the dataset. With only 180 data available for training, I wasn't able to train this data using other classifiers like neural network, in case it ends up overfitting the training dataset.

A reviewer has introduced Stacking into the training method. This is the first time I'm using such method, and I found it interesting and I am seriously considering to use this method in future projects.

5.1: Improvements

A few improvements could be done :-

- More data

The dataset could use a higher number of data. Training with high number of cleaned data can help the algorithm predicts better.

- Better feature engineering

I only created one new feature called 'age-range', and based on RFE Feature Selection model or Pearson's correlation, this new feature is actually one of the 'weaker' feature and it did not contribute much to prediction. It is better to work with heart specialist as they can identify missing feature and creates a new one.

- Local data for local adaptation

This dataset is provided by Cleveland Heart Disease Database. However, it may not work as well in different regions as we need to cater for local adaptation. A population might have a higher than average cholesterol level, but no heart disease is present. In this case it is better to use a local data and re-train the model again.