A comparison of different machine learning models for predicting heart attack

Jummy F. David[1,2]

1 Department of Mathematics and Statistics, York University, Toronto, Ontario, Canada.
2 Laboratory for Industrial and Applied Mathematics (LIAM), York University, Toronto, Ontario, Canada.

*__Correspondence:__ jummy30@yorku.ca

Abstract

Heart attack occurs as a result of sudden blockage of blood flow to the heart. Several machine learning models have been used to predict a chance of heart attack. This study uniquely explores different machine learning and neural network models, and compares their prediction ability using confusion matrix, R2 scores, accuracies and errors. Of all models examined (Random Forest, Support Vector Machine (SVM), K-Nearest Neighbour, Decision Tree, Logistic Regression, Ensemble by stacking, Multilayer Perceptron Neural Network), SVM performed best at predicting a heart attack, with the overall accuracy of 88% for SVM with polynomial kernel. Therefore, SVM model is recommended for medical decision making and as a useful tool for heart attack prediction with similar dataset.

Keywords: Heart Attack, Machine learning models, Neural Networks, MLP, SVM, Classification, Prediction.

1 Introduction

A heart attack (also called myocardial infarction) occurs when the blood flow to a part of the heart muscle is blocked. Based on WHO's report, out of the estimated 17.9 million people who died from Cardiovascular diseases (CVDs) in 2019, 85% were due to heart attack and stroke [15]. The more the delay in treatment, the greater the damage to the heart muscle. Coronary artery disease (CAD) is known to be the main cause of heart attack. Different machine learning algorithms have been used for predicting the quality of surface water [4], predicting and mapping the susceptibility of gully erosion [12], predicting cancer survival [11], predicting the risk of critical COVID-19 [2], time series forecasting [1], and groundwater mapping [10], to mention a few.

While Hasan et al. [7] compared two different models (support vector machine and convolutional neual network) for hyperspectral image classification task and James et al. [9] introduced SVM to determine the beginning and end of recessions in real time, the present study aimed at comparing different machine learning models such as Random Forest, Support Vector Machine (SVM with polynomial kernel), K-Nearest Neighbour, Decision Tree, Logistic Regression, Ensemble by stacking, and Multilayer Perceptron Neural Network) based on programming-based methodology. The proposed models and methodology comprised different steps: Heart attack classification using different models from the confusion matrix, the comparison between the accuracy scores, R2 scores assessment and root mean square errors. The next section briefly describe the data, problem and approach. Section 3 gives details of the data and exploration, while Section

4 explains the proposed models, algorithm and outcome. Results and findings are elaborated in Section 5, while Section 6 concludes the study with some important implications and limitations.

2 Data and problem definition

2.1 Brief on dataset

The dataset imported from kaggle [3] includes medical information and the chances of some patients getting a heart attack. The dataset [3] includes 14 attributes such as **target** (target: 0= less chance of heart attack 1= more chance of heart attack), **age** (ages 29-77 years), **sex** (sex: 1 = male; 0 = female), **chest pain** (cp: 0 = typical angina; 1 = atypical angina; 2 = non-anginal pain; 3 = asymptomatic), **resting blood pressure** (trestbps: 94-200), **serum cholestoral in mg/dl** (chol: 126-564), **fasting blood sugar ¿ 120 mg/dl** (fbs: 1 = true; 0 = false), **resting electrocardiographic results (values 0,1,2)** (restecg: 0 = normal; 1 = having ST-T wave abnormality; 2 = showing probable or definite left ventricular hypertrophy by Estes' criteria), **maximum heart rate achieved** (thalach: 71-202), **exercise induced angina** (exang: 1 = yes; 0 = no), **oldpeak = ST depression induced by exercise relative to rest** (oldpeak: 0-6.2), **the slope of the peak exercise ST segment** (slope: 0 = upsloping; 1 = flat; 2 = downsloping), **number of major vessels (0-3) colored by flourosopy** (ca:), and **thal** (thal: 0 = normal; 1 = fixed defect; 2 = reversable defect). Further exploration of the data will be discussed in Section 3. In addition, the complete attribute documentation can be found at [14], and several works done using the dataset can be found on kaggle at [3].

2.2 Problem definition and approach

Using the medical information given in the dataset, we want to classify the target variable with some machine learning models and show the algorithms that are best at predicting the possibility of patients getting a heart attack. For the prediction, we examined models such as *Random Forest Classifier, Support Vector Machine, K-Nearest Neighbour, Decision Tree Classifier, Logistic Regression, Ensemble: Stacking models, and Multilayer Perceptron Neural Network*. We made comparison based on their confusion matrices, accuracies, R2 scores and root mean square errors, and conclude that the model with the lowest number of misclassified instances, highest accuracy and lowest error is the best at correctly predicting heart attack in patients.

3 Data exploration and description

From our data exploration, the first five rows of the dataset is given in Table 1.

Table 1: ***The first*** 5 ***rows and*** 14 ***features of the heart dataset.***

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

The dataset which is of 303 rows and 14 features including the target, has no empty rows. From Figure 1 below, we observe that **cp**, **thalach**, **slope** are the three most positively correlated features, while **exang**,

oldpeak, **ca** and **thal** are the four most negatively correlated features, with the name of each features given as in Section 2.

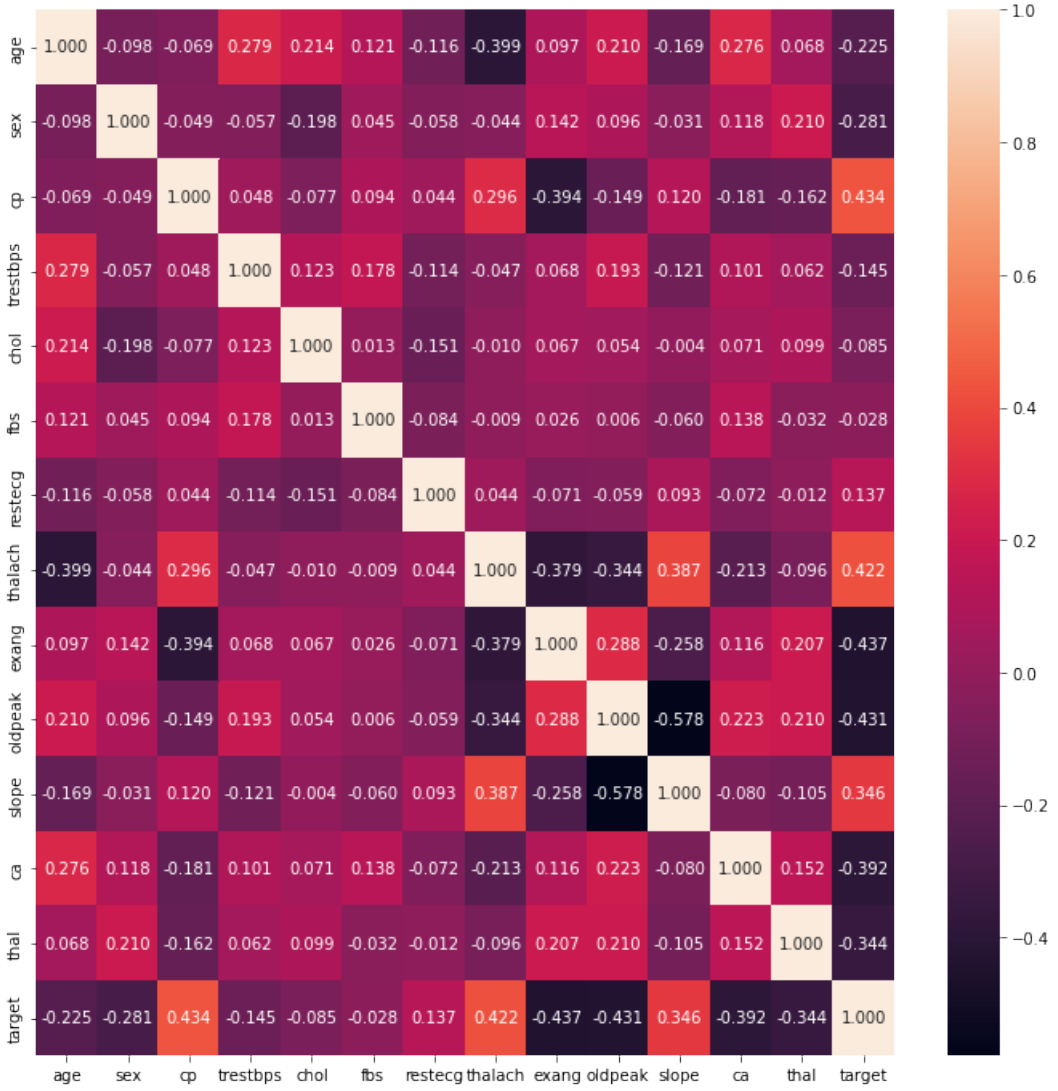


Figure 1: ***Correlation matrix of the heart attack dataset. The colorbar denote the weight and the position of the correlation of each of the features. The lower and higher values respectively denote the strength of the correlation, while the positive and negative sign denote the direction of the correlation. For example, cp is strongly and positively correlated to the target variable with a value of*** 0.434***.***

Furthermore, we checked for multicollinearity using *Variance Inflation Factor* (**VIF**) and dropped the feature **trestbps** with the value of VIF = 58.56. After solving the multicollinearity problem, we split the data into training and testing set with 20% used for testing and a random state of 101. We normalized the testing and training set using the MinMaxScaler library in Scikit Learn with feature range of (0,1) [13]. Detail of algorithm used in generating all outputs and results is available on my GitHub repository [5].

4 Model and algorithm

Different machine learning models are randomly selected based on their good training ability for a classification task. Classification can be explained as a process of categorizing a dataset into classes. The following subsections briefly explain different models, algorithms and approaches known for training a binary classifi-

cation task with two outcomes (heart attack:0/1, i.e 0 for less chance and 1 for more chance). These models are chosen because they support and are great for a binary classification problem of this nature.

4.1 Random Forest Classifier

This is one of the most powerful Machine Learning algorithm in Kaggle competitions since it often work well with no tuning. Random Forest algorithm adds more randomness and the algorithm usually result in greater tree diversity, which trades a higher bias for a lower variance. In fact, it works better with small ensembling. It is basically a collection of several decision tree models. Using the grid search with 5-fold cross-validation (with the help of **Grid Search CV** class) for various values of max_depth, min_samples_leaf and max_leaf_nodes in the range of $0 - 200$, good *hyperparameter* values for a RandomForestClassifier are found. In addition, I cross validate with *recursive feature elimination* (RFECV) using the different number of features and have the optimal number to be 11. The effect on the test R2 score and root mean square error (RMSE) is checked, with the performance of the model measured on the test set. The model with the normalised data and hyperparameter tuning gives a test score and RMSE of 83.6% and 0.40 respectively.

4.2 Support Vector Machine

Support Vector Machine (SVM) were a power house machine learning before the advent of neural networks. SVM is very useful especially with limited amount of data since it has a self-regularizing feature that reduces overfitting. One of the useful features of SVM that differentiate it from most of the other models is the built-in, in-sample metric used to characterize the out-of-sample performance [6, 9]. This algorithm was selected due to its outstanding generalization capability and reputation in achieving high accuracy. Furthermore, SVM has proven to be a great useful tool in capturing non-linearities. The SVM algorithm

```
1 svm_linear = svm.SVC(kernel='poly')
```

is implemented using a *kernel trick* (a form of hyperparameter tuning, which is able to project data into a higher dimensional space such that it is able to create hyperplane that is inclined with the dimension of the newer space so that the created hyperplane in the new space is used to classify the data point effectively [6]). The goal of SVM algorithm is to find the hyperplane that have the most significant margin from the support vector. The polynomial kernel used gives a highest accuracy compared to other kernels explored (linear, rbf, sigmoid) in modelling the nonlinearity implemented by the algorithm. The SVM model has a classification error of 0.34 and an accuracy of 88.5%.

4.3 K-Nearest Neighbours

Here, K-Nearest Neighbour (**KNN**) is used, and the idea behind the algorithm is to find the nearest input

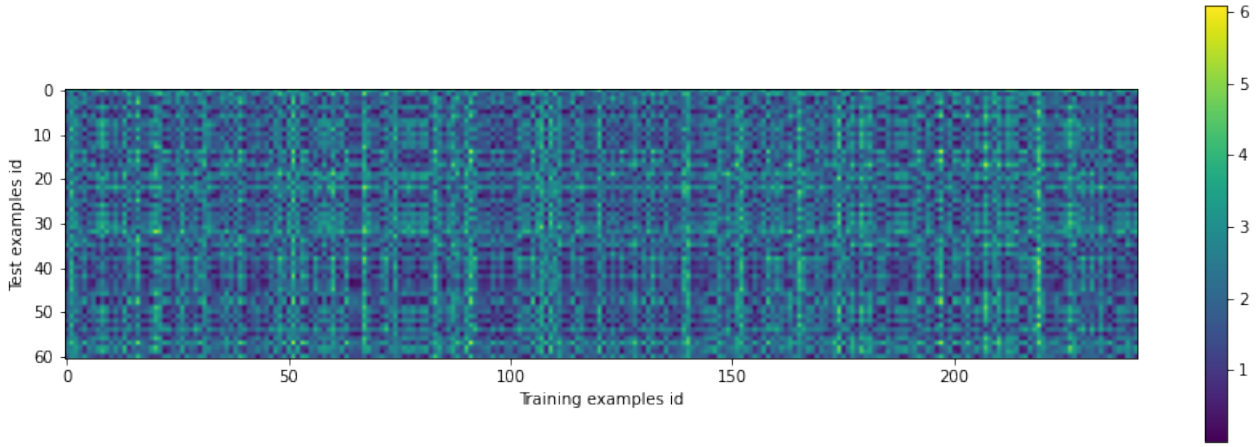


Figure 2: ***The distance matrix showing the closeness between the testing and training examples.***

vector(s) to a novel input vector x in the training set using the euclidean distance equation (4.1) below

$$\sqrt{\sum_{i=1}^{k}(x_i^{(a)} - x_i^{(b)})^2} \equiv \sqrt{(x_1^{(a)} - x_1^{(b)})^2 + \ldots + (x_k^{(a)} - x_k^{(b)})^2}[8], \tag{4.1}$$

and then copy the label(s). The euclidean equation is used to compute the distance between the data and other neighbors (data). Using the normalised data, I implements and train a KNN classifier on the complete dataset and calculate the euclidean distance for all the pairs of test examples as in Figure 2 using the code

```
knn = KNeighborsClassifier(n_neighbors=7).
```

The colorbar from Figure 2 shows that the testing and training data are not too farther apart (more of the deeper blue colors and less visible brighter colors), which means that the data is good for training since there are small euclidean distances between the points. As a form of hyperparameter tuning, I calculated the test accuracy with respect to different choices of K and found that $K = 7$ gives the best test set accuracy of 83.6% and RMSE of 0.40. It is possible to increase the performance of the model and capture regularities by increasing the training samples. I calculated the test accuracy with the best K and reported the R2 score, test accuracy, root mean square error (RMSE) in Section 5.1 and confusion matrix 5.2.

4.4 Decision Trees Classifier

Decision trees Classification model make predictions by recursively splitting on the attributes in the dataset according to a tree structure. Similar to Random Forest, the grid search with 5-fold cross-validation is used to find good hyperparameter values for max_depth, min_samples_leaf, criterion, splitter and max_leaf_nodes in the range $0 - 200$ using the code

```
tuned_params_model = DecisionTreeClassifier(max_depth=5, criterion = 'entropy',
    min_samples_leaf=10, min_samples_split=25, splitter='random', random_state
    =101)
```

The effect on the test R2 score and root mean square error (RMSE) is checked, with the performance of the model measured on the test set. The model gives a test score and RMSE of 75.4% and 0.50 respectively. Overall, the test time of this model appears to be faster than the others and this model gives the lowest test accuracy.

4.5 Logistic Regression

Logistic Regression is a sigmoid-like or S-shaped function usually written as

$$\sigma(z) = \frac{1}{1+e^{-z}} \in (0,1). \tag{4.2}$$

When the output of the linear model with a sigmoid function is squashed, the procedures become

$$z = w^T x + b, \quad \text{and} \quad y = \sigma(z), \quad \text{with} \quad l_{\text{SE}}(y,t) = \frac{1}{2}(y-t)^2 \quad \text{and} \quad y \in [0,1][8]. \tag{4.3}$$

In the algorithm, I used the $l2$ penalty from the code

```
logistic_regression= LogisticRegression(penalty='l2')
```

which applies a square and gives a higher accuracy compared to l1. The effect on the R2 score and root mean square error (RMSE) is checked, with the performance of the model measured on the test set. The model gives a test score and RMSE of 83.6% and 0.40 respectively.

4.6 Ensemble: Stacking

Ensemble of model means combining a set of models or a set of predictions whose individual output is combined to make a final output in order to make a better prediction [6]. This is one of the ways to fine-tune the system, and this approach will often generate a better prediction when compared to the best individual models. Most known Ensemble methods are bagging, boosting, and stacking. This section explores stacking (short for stacked generalization) of models. I train different classifiers (Random Forest, Decision Tree, and Extreme Gradient Boosting) on different algorithms in order to increase the chance of making very different types of errors while improving the ensemble's accuracy. Briefly, an Extreme Gradient Boosting (XGB) is an optimized implementation of Gradient Boosting in popular Python library [6]. In fact, stacking performs a classification task on a new instance. As a sample code, the following algorithm creates and trains a stacking classifier in Scikit-Learn.

```
def get_stacking():
  # define the base models
  level0 = list()
  level0.append(('random_forest', RandomForestClassifier()))
  level0.append(('cart', DecisionTreeClassifier()))
  level0.append(('xgradboost', XGBClassifier()))
  # define meta learner model
  level1 = LogisticRegression()
  # define the stacking ensemble
  model = StackingClassifier(estimators=level0, final_estimator=level1, cv=5)
  return model

stacked_model = get_stacking()
stacked_model = stacked_model.fit(normalized_X_train_m01, y_train)
y_pred_stack = stacked_model.predict(normalized_X_test_m01)
test_stack = stacked_model.score(normalized_X_test_m01, y_test)
acc_stack = metrics.accuracy_score(y_true=y_test, y_pred=y_pred_stack)
error_stack = rmse(y_test, y_pred_stack)
```

The effect on R2 score and root mean square error (RMSE) is checked, with the performance of the model measured on the test set. The model gives a test score and RMSE of 86.9% and 0.36 respectively.

4.7 Multilayer Perceptron Classifier (Artificial Neural Network)

Multilayer perceptron (MLP) is a feed-forward pass artificial neural network and could be referred to as networks consisting of multiple layers of perceptrons (with threshold activation) that train using Back-propagation. MLP is a simple type of neural net architecture designed to process multi-variable inputs and consists of fully connected layers [6, 8]. The output units are a function of the input units so that $y = f(x) = \phi(Wx + b)$, with W denoting the weights matrix, and b the biases.

MLPClassifier supports multi-class classification when *Softmax* is applied at the output later. The MLP code is given as

```
MLPC = MLPClassifier().fit(normalized_X_train_m01,y_train).
```

The effect on R2 score and root mean square error (RMSE) is checked, with the performance of the model measured on the test set. The model gives a test score and RMSE of 86.9% and 0.36 respectively.

5 Results and findings

Algorithm selection for a chance of heart attack is performed and reported. This section further explains the results of different models and make comparison based on their accuracy, R2 score, error and confusion matrix on the test data to access their predictive power.

5.1 Comparison between model accuracy and error

Model performance could be measured using accuracy and error and especially when the dataset is balanced. But performance measurement from accuracy may not be a reliable performance metric when the dataset is imbalanced. Table 2 and Figure 3 compare and show the prediction ability of the models. SVM model (magenta color in Figure 3) outperformed other models by having the highest accuracy and the lowest RMSE. Stacking and MLP (cyan and orange color in Figure 3) performed second best by giving similar results (see Table 2 and Figure 3). The least performing model is Decision Tree based on their accuracy (75.4%) and error (0.5) from Table 2.

Table 2: ***Table of models, accuracies, scores and errors***

Model	**Accuracy (%)**	**R2 Score (%)**	**RMSE**
Random Forest	83.606557	83.606557	0.404888
Support Vector Machine	88.524590	88.524590	0.338754
K-Nearest Neighbour	83.606557	83.606557	0.404888
Decision Tree	75.409836	75.409836	0.495885
Logistic Regression	83.606557	83.606557	0.404888
Stacking Models	86.885246	86.885246	0.362143
Multi-Layer Perceptron	86.885246	86.885246	0.362143

5.2 Comparison between models and methods using confusion matrix

In addition to the comparison made based on the accuracy and error, the predicted results are summarised in table layout to allow visualization of the performance measure of the proposed ML models for this binary classification problem. One of the best ways that we can evaluate the performance of a classifier is to check the confusion matrix. The idea behind a confusion matrix is to count the number of instances that a class A was wrongly classified as class B, i.e., the number of instances a classifier confused class A with class B.

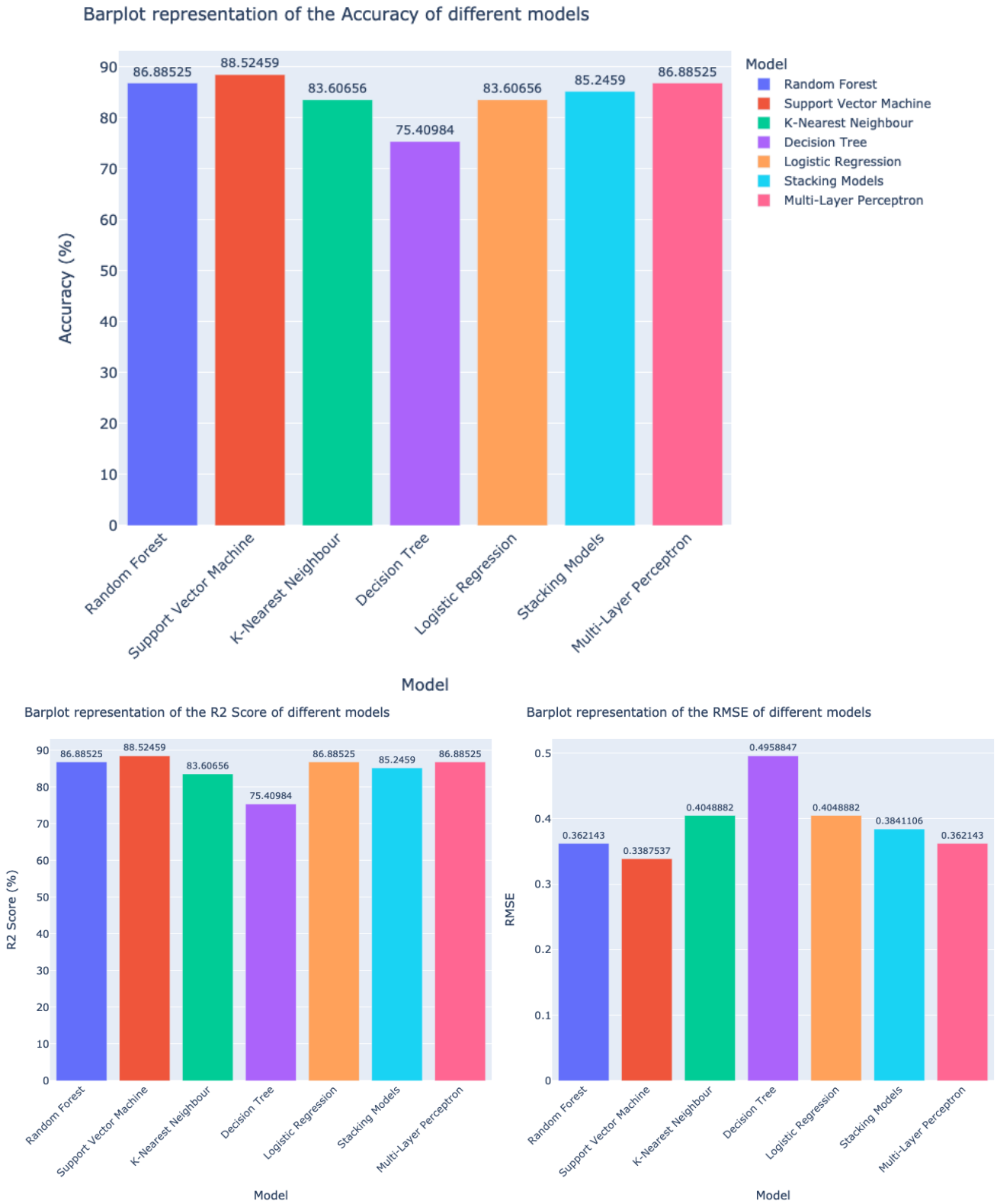


Figure 3: ***Barplot representation of the accuracy (row 1), R2 Score (row 2) and RMSE (row 3) for Random Forest, SVM, KNN, Decision Tree, Logistic Regression, Stacking, and MLPC. The lower the RMSE error and the higher the accuracy and R2 Score, the better the performance of the model. For example, SVM with polynomial kernel have the highest accuracy and R2 score and a lower RMSE value. This model performed best of all proposed models.***

A confusion matrix is used to make prediction on the test set, so that each row represents actual class while each column represents a predicted class. Figure 4 below illustrates how our prediction is compared to the actual target. A perfect classifier should have only true positive and true negative so that its confusion matrix would have nonzero values only on its main diagonal. A sensible model will have the principal diagonal element values (TN and TP from Figure 4) very high with the off-diagonal (FP and FN from Figure 4) very low.

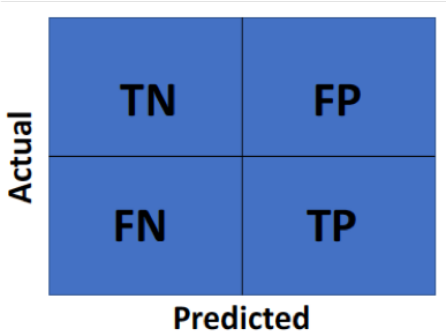
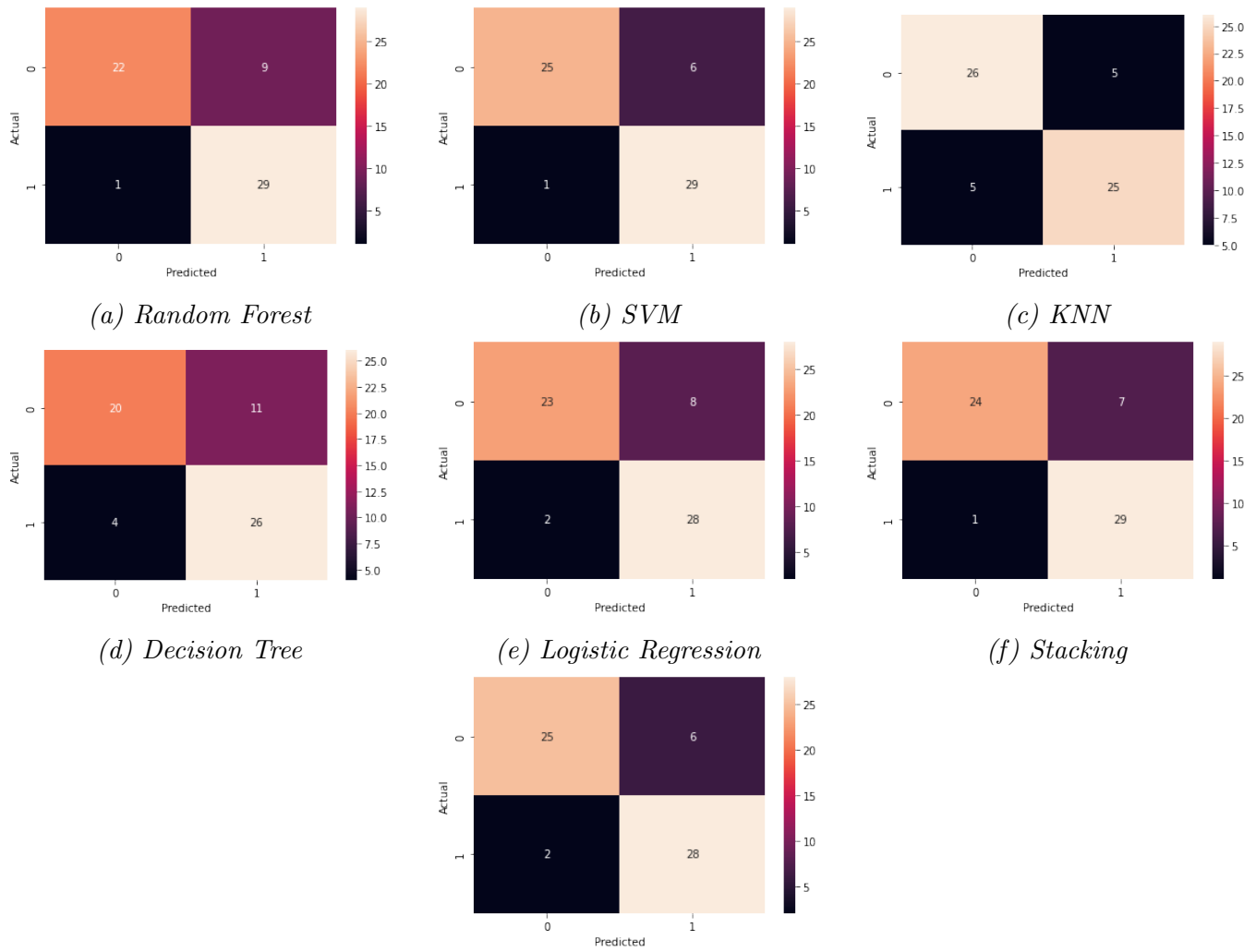


Figure 4: ***An illustration of a confusion matrix showing examples of true negative (top left), false positive (top right), false negative (lower left) and true positive (lower right).***

Figures 5 below shows the confusion matrix for each of the models and illustrates how the models correctly/wrongly classified having to not having a heart attack. In this figure, the first row of the matrix considers less chance of heart attack (the negative class 0), while the second row considers more chance of heart attack (the positive class 1). Figure 5b for SVM performs best at correctly making prediction with one false negative (the model predicted less chance of a heart attack once but it is false) and six false positive (the model predicted more chance of heart attack six times but it is false). On the first row, 25 of them are correctly classified as less chance of heart attack (TN: true negatives), while the remaining 6 are wrongly classified as having more chance heart attack (FP: false positives). On the second row, 1 is wrongly classified as less chance of heart attack (FN: false negatives), while the remaining 29 are correctly classified as having more chance of heart attack (TP: true positives). Overall, SVM has 7 misclassified instances, Stacking (Figure 5f) and MLP classifier (Figure 5g) have the same number of misclassified instances (8) and Decision Tree (Figure 5d) has the highest (15). Obviously, this supports our claim on SVM performing best and Decision Tree performing much worst than the rest. Further information regarding the precision and recall can be viewed from the classification report obtained from the code in my GitHub repository [5].



(a) Random Forest *(b) SVM* *(c) KNN*

(d) Decision Tree *(e) Logistic Regression* *(f) Stacking*

(g) MLPC

Figure 5: ***Confusion matrix representation of the classification accuracy of each model. For example, SVM with polynomial kernel have the highest accuracy and R2 score and a lower RMSE value. This model performed best of all proposed models.***

6 Conclusions and future direction

The main goal of the study is to evaluate different machine learning models (Random Forest, Support Vector Machine (SVM), K-Nearest Neighbour, Decision Tree, Logistic Regression, Ensemble by stacking, Multilayer Perceptron Neural Network) and compare based on their accuracy, error and confusion matrix. We assessed the model that performed best at predicting a chance of heart attack. The comparison showed that SVM performed best with an accuracy of 88.5%, error of 0.34 and a misclassification of 7 instances. Whereas Decision Tree performed much worst than the rest of the models with an accuracy of 75.4%, error of 0.50 and a misclassification of 15 instances.

The results and conclusions made have important implication and limitations. Due to time constraint, I could not tune the hyperparameters to the desired level, and I believe that the accuracy and the performance of some of the models could be improved by further tuning. We could also improve the predictability if we train more neural network models and deep learning, especially in a multiclass classification algorithm. Possible work in this direction is advisable and encouraged. In addition, this study has no need for reducing the dimension of the data using the Principal Component Analysis (PCA), which is used for features reduction when a data has a lot of features.

The implication of the results in this study means that in the medical world, for early detection and

management of heart attack, SVM model with a higher accuracy will be a great model to select. In other words, there is need to further tune hyperparameters. Similarly, in terms of dataset, some features are either not important or have been explained by other features, causing multicollinearity data. Features like these should be removed or not considered. Instead, features like race, socioeconomic status, access to medical care and so on, should be included in the dataset to make the data more inclusive. Despite these limitations, the study was able to explore different models, their algorithms and prediction level. This study complements different studies available on the comparison of different machine learning models for predicting a chance of heart attack.

Acknowledgement

I gratefully acknowledge Vector Institute for the opportunity given to learn. I sincerely appreciate the support from the instructor and tutors in attaining this level in Machine Learning.

References

[1] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. An empirical comparison of machine learning models for time series forecasting. *Econometric reviews*, 29(5-6):594–621, 2010.

[2] Dan Assaf, Ya'ara Gutman, Yair Neuman, Gad Segal, Sharon Amit, Shiraz Gefen-Halevi, Noya Shilo, Avi Epstein, Ronit Mor-Cohen, Asaf Biber, et al. Utilization of machine-learning models to accurately predict the risk for critical covid-19. *Internal and emergency medicine*, 15(8):1435–1443, 2020.

[3] Naresh Bhat. *Heart Attack Data on kaggle*, (accessed March 2, 2022). https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility.

[4] Kangyang Chen, Hexia Chen, Chuanlong Zhou, Yichao Huang, Xiangyang Qi, Ruqin Shen, Fengrui Liu, Min Zuo, Xinyi Zou, Jinfeng Wang, et al. Comparative analysis of surface water quality prediction performance and identification of key water parameters using different machine learning models based on big data. *Water research*, 171:115454, 2020.

[5] Jummy F. David. *A comparison of different machine learning models for predicting heart attack*, (created March 6, 2022). https://github.com/funkedavid82/Capstone-Project-on-Machine-Learning-and-Neural-Network.

[6] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.

[7] Hayder Hasan, Helmi ZM Shafri, and Mohammed Habshi. A comparison between support vector machine (svm) and convolutional neural network (cnn) models for hyperspectral image classification. In *IOP Conference Series: Earth and Environmental Science*, volume 357, page 012035. IOP Publishing, 2019.

[8] Vector Institute. Notes on machine learning - black & indigenous program. 2022.

[9] Alexander James, Yaser S Abu-Mostafa, and Xiao Qiao. Nowcasting recessions using the svm machine learning algorithm. *arXiv preprint arXiv:1903.03202*, 2019.

[10] Davoud Davoudi Moghaddam, Omid Rahmati, Mahdi Panahi, John Tiefenbacher, Hamid Darabi, Ali Haghizadeh, Ali Torabi Haghighi, Omid Asadi Nalivan, and Dieu Tien Bui. The effect of sample size on different machine learning models for groundwater potential mapping in mountain bedrock aquifers. *Catena*, 187:104421, 2020.

[11] Mitra Montazeri, Mohadeseh Montazeri, Mahdieh Montazeri, and Amin Beigzadeh. Machine learning models in breast cancer survival prediction. *Technology and Health Care*, 24(1):31–42, 2016.

[12] Omid Rahmati, Nasser Tahmasebipour, Ali Haghizadeh, Hamid Reza Pourghasemi, and Bakhtiar Feizizadeh. Evaluation of different machine learning models for predicting and mapping the susceptibility of gully erosion. *Geomorphology*, 298:118–137, 2017.

[13] scikit learn. *sklearn.preprocessing.MinMaxScaler¶*, (accessed March 6, 2022). https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.

[14] UCL. *Heart Attack Data with complete attributes*, (accessed March 6, 2022). https://archive.ics.uci.edu/ml/datasets/Heart+Disease.

[15] WHO. *Cardiovascular diseases (CVDs)*, (Assessed March 8, 2022). https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds).

A Appendix

The complete code used in the report and the submitted report will soon be available in my GitHub repositiory [5].