HW1

Data Preparation

Missing Values

First of all, dataset contains *NA* values. There are multiple ways to handle the missing values. In this case, *BPMeds* column filled with 0, *glucose*, *totChol*, *BMI* & *heartRate* filled with average of the column and *education* filled with 1. Rest of the columns consists *NA* are dropped out.

Feature Selection

Dataset consist of 15 columns except the prediction column, which is *TenYearCHD*. So, I tried to choose the best features by using a **sklearn.feature_selection** module. Since some of the features does not affect the outcome, such as *Education*, analyzed the best features by using *chi2* as a scoring function inside of the **SelectKBest** module. As we can see from the Table 1, top 12 features selected into the pipeline.

Features	Scores
sysBP	723.87
glucose	396.88
age	320.49
totChol	239.49
cigsPerDay	221.65
diaBP	150.72
prevalentHy	91.25
diabetes	38.82
BPMeds	30.47
male	19.04
prevalentStr	16.00
BMI	14.19

Table 1: Features with Scores

Outlier Elimination

When we draw the box plots of the top 12 features, except the 1-0 binary columns, there exits some outliers, as can be seen below. The outliers are eliminated by manually.

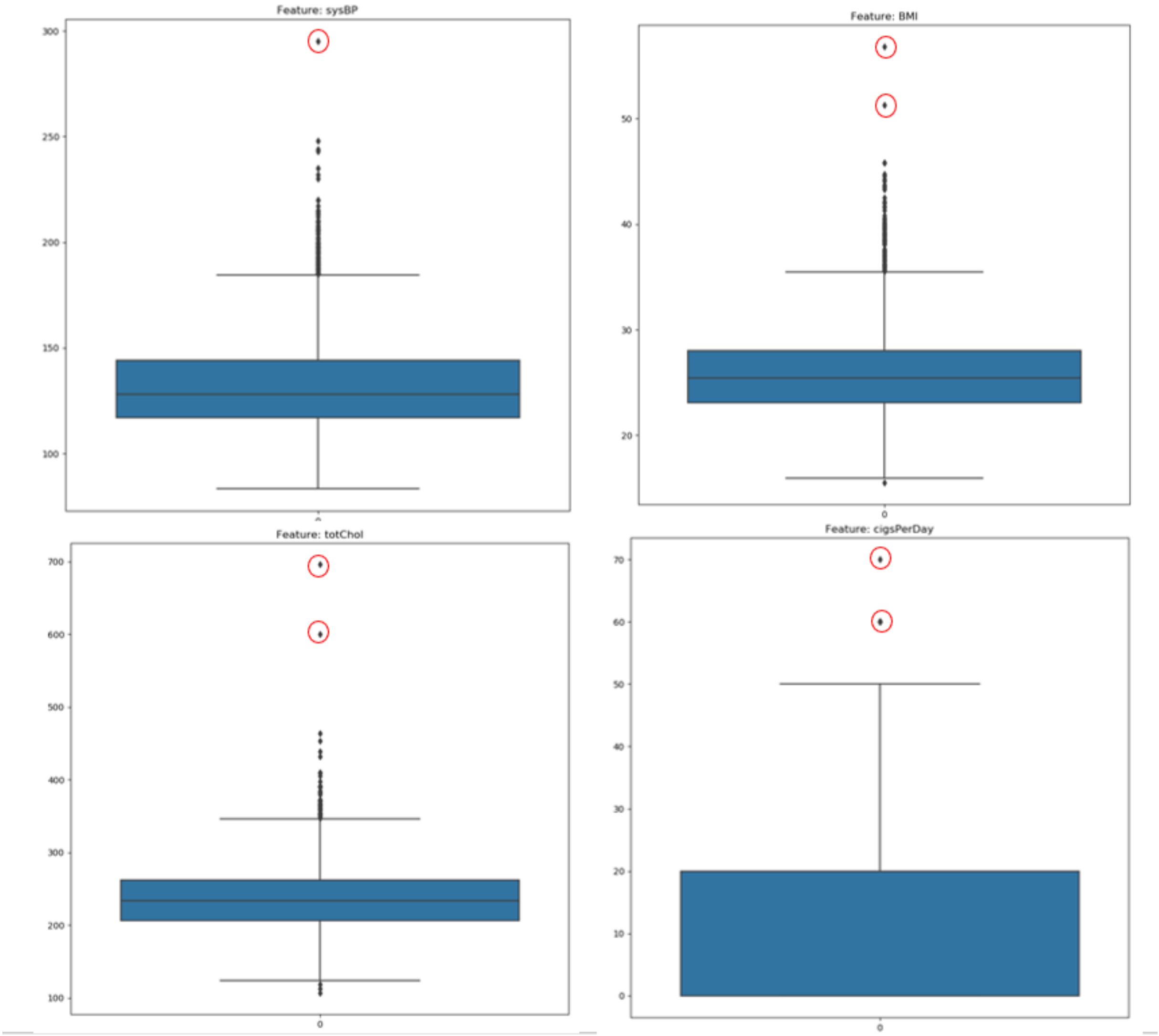


Figure 1: Example Outliers

Model Selection

For this dataset, I tried multiple models such as: ***Logistic Regression, Multi-layer Perceptron Classifier, Random Forest Classifier, Bagging Classifier, Gaussian Naive Bayes, Support Vector Classifier*** and ***k-Nearest Neighbors***. Since the performance criteria of the assignment is *ROC-AUC* score, below we can see the roc auc scores of each model, individually.

Models	Roc Auc Scores
svm	0.629
knn	0.583
naive bayes	0.712
mlp-nn	0.673
random forest	0.71
logistic	0.728
bagging	0.508

Table 2: Models with Roc Auc scores

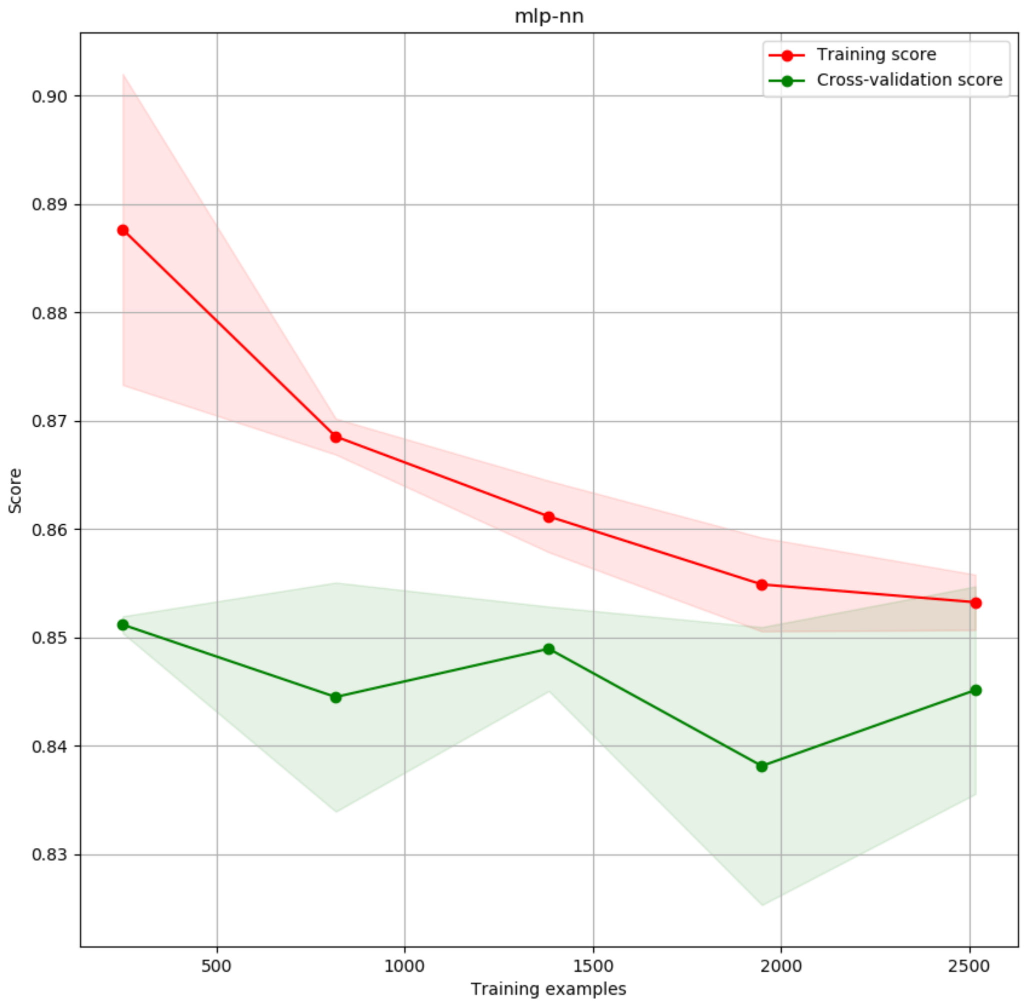
Logistic Regression, Gaussian Naive Bayes and Random Forest scored highest individually. Since our aim is to beat these models with developing an ensemble model, our benchmark will be Logistic Regression with roc auc score 0.728.

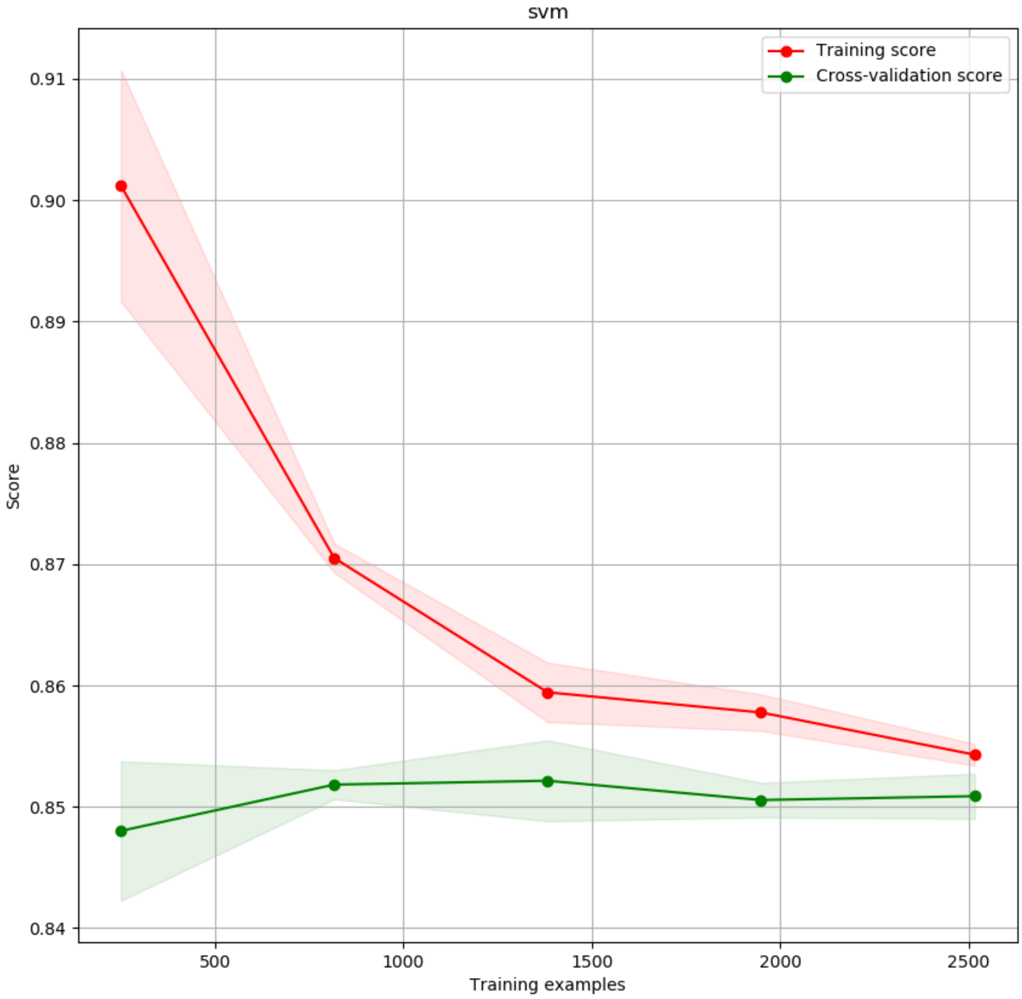
Ensemble

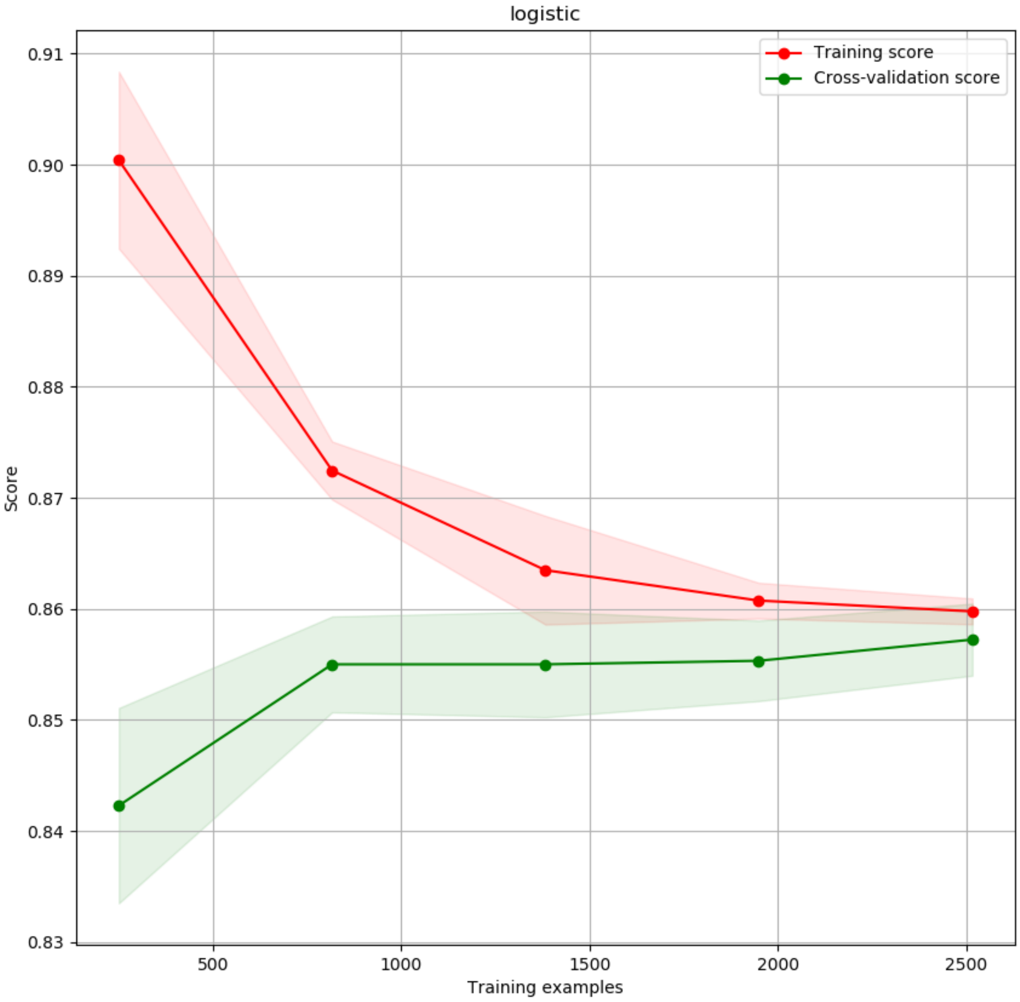
First of all, since there are lots of options, generated all the combinations of models with length of 2 to 7.

After generating all possible combinations, took the average of all combinations in order to calculate roc auc score with respect to test_y. After this operation, ***Support Vector Machin***e, ***Multi-layer Perceptron Classifier*** and ***Logistic Regression*** yielded the max roc auc score which is 0.733. For a sanity check, I also tried Voting Classifer with soft voting argument, which basically does the same thing.

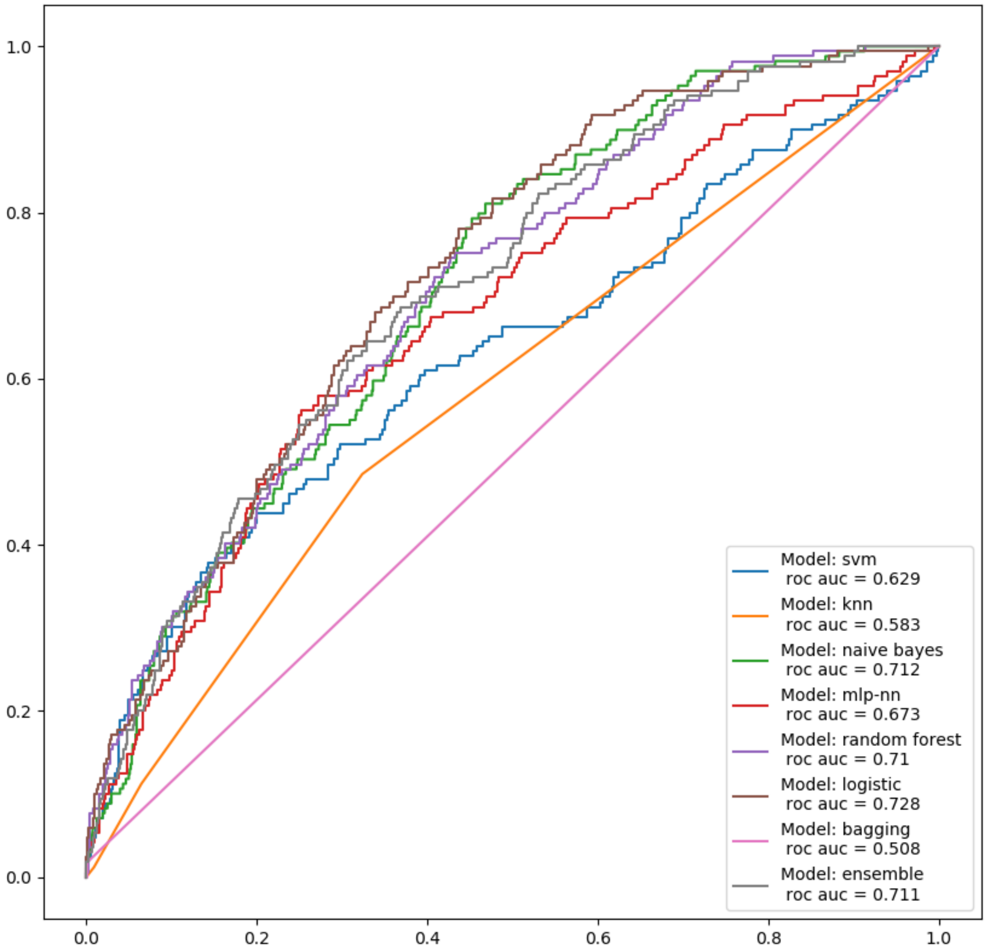
Following graphs show the learning curves of the selected models. All three graphs show a good trend which training and cross-validation curve are close together, this yields a better generalization on the predictions. (Other learning curve graphs can be found in the notebook.)




svm
Training score
Cross-validation score
0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
Score
500
1000
1500
2000
2500
Training examples


logistic
Training score
Cross-validation score
0.91
0.90
0.89
0.88
0.87
0.86
0.85
0.84
0.83
Score
500
1000
1500
2000
2500
Training examples

In conclusion, ensemble of three models yields a score of 0.733 which beats all the other models. As can be seen below, gray line which represents the ensemble slightly over the highest roc auc score.



References

- https://scikit-learn.org/stable/
- https://www.datacamp.com/community/news/feature-selection-using-selectkbest-0dv0fo0qqe48
- https://www.kaggle.com/c/santander-customer-transaction-prediction/discussion/87211
- https://mlwave.com/kaggle-ensembling-guide/
- https://medium.com/@rrfd/boosting-bagging-and-stacking-ensemble-methods-with-sklearn-and-mlens-a455c0c982de
- https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
- https://www.kaggle.com/neisha/heart-disease-prediction-using-logistic-regression
- https://towardsdatascience.com/lets-learn-about-the-roc-auc-curve-by-predicting-spam-d8007746a6f9
- https://machinelearningmastery.com/how-to-score-probability-predictions-in-python/
- https://www.kaggle.com/lauriandwu/machine-learning-heart-disease-framingham/output