Introduction

With the improvement of technology and data explosion, the application of big data analytics with data mining techniques has been greatly applied in the medical field. Big Data Means Big Changes for Medical Diagnostics. Although human body is extremely complex, the highly organized structure allows us to uncover some patterns by analyzing our bioinformation. It provides insights of the hidden information of human body and helps researchers explore and provide more advanced healthcare knowledge. In this project, we target two bioinformatic dataset, and it contains two parts, classification algorithms review and regression analysis. In part 1, we applied many classification algorithms on a Cardiovascular disease dataset with several versions of data preprocessing and tried to figure out the best classification approach based on the evaluation of these algorithms' performances. During the data pre-processing process, we tried different methods and aimed to increase the final accuracy. During the testing of each algorithm, we analyze the usage of the specific algorithm applied and generate a report of the approach. At the end of part 1, we have a comparison of all the classification algorithms we have used.

Classification and prediction

1. Dataset Analysis

The Cardiovascular disease dataset is from Kaggle and contains 70,000 instances and 13 attributes. All of the dataset values were collected at the moment of medical examination, such as age, height, weight, systolic blood pressure, diastolic blood pressure, cholesterol level, glucose level, smoking habit, alcohol intake, physical activity, and the target label, presence or absence of cardiovascular disease. Some of the categories were factual information, and some were the results of medical examination, and the rest were information given by the patient. Figure 1 illustrates the description of the dataset.

1. Dataset description (Original):

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
count	70000.00	70000.00	70000.00	70000.00	70000.00	70000.00	70000.00	70000.00	70000.00	70000.00	70000.00	70000.0	70000.0
mean	49972.42	19468.87	1.35	164.36	74.21	128.82	96.63	1.37	1.23	0.09	0.05	0.8	0.5
std	28851.30	2467.25	0.48	8.21	14.40	154.01	188.47	0.68	0.57	0.28	0.23	0.4	0.5
min	0.00	10798.00	1.00	55.00	10.00	-150.00	-70.00	1.00	1.00	0.00	0.00	0.0	0.0
25%	25006.75	17664.00	1.00	159.00	65.00	120.00	80.00	1.00	1.00	0.00	0.00	1.0	0.0
50%	50001.50	19703.00	1.00	165.00	72.00	120.00	80.00	1.00	1.00	0.00	0.00	1.0	0.0
75%	74889.25	21327.00	2.00	170.00	82.00	140.00	90.00	2.00	1.00	0.00	0.00	1.0	1.0
max	99999.00	23713.00	2.00	250.00	200.00	16020.00	11000.00	3.00	3.00	1.00	1.00	1.0	1.0

Figure 1. The dataset description

This dataset was pretty balance. The target label shows that there were about half people are healthy, and half have cardiovascular disease. It had no missing values; however, there were some clearly incorrect values on the height, weight, systolic blood pressure, and diastolic blood pressure. By examining the age, we could see that the patients distributed at a range between 30 years old and 65 years old, and we have also checked related medical information, so the 55 center meters height and 10 kilograms weight would be highly incorrect. There were also impossible values on blood pressures, such as negative values, extremely high values, and the values that diastolic blood pressure was greater than the systolic blood pressure.

2. Data Pre-processing

Some cleaning for the height, weight, and blood pressure values were needed in order to feed the dataset into classification models. We removed the samples which contained outliers and incorrect values according to the documentations from health professionals from the Centers for Disease Control and Prevention (CDC) and the World Health Organization (WHO) There were about 66,953 instances left with more reasonable values. Moreover, the age values were transformed into years instead of in days, and this was our first version of input dataset shown as Figure 2. We removed the attribute “id” and label “cardio” and fed it into classification models. However, the accuracy of the 11 classification algorithms after parameter tuning were in a range between 69% and 74%.

2. Dataset description (After clearning):

	id	age	gender	height	weight	ap_hi	ap_lo	cholesterol	gluc	smoke	alco	active	cardio
count	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.0	66953.00
mean	49958.67	53.33	1.35	164.44	73.88	126.33	81.18	1.36	1.22	0.09	0.05	0.8	0.49
std	28859.85	6.76	0.48	7.55	13.37	15.78	8.95	0.68	0.57	0.28	0.22	0.4	0.50
min	0.00	30.00	1.00	144.00	45.00	90.00	60.00	1.00	1.00	0.00	0.00	0.0	0.00
25%	24956.00	48.00	1.00	159.00	65.00	120.00	80.00	1.00	1.00	0.00	0.00	1.0	0.00
50%	50007.00	54.00	1.00	165.00	72.00	120.00	80.00	1.00	1.00	0.00	0.00	1.0	0.00
75%	74860.00	58.00	2.00	170.00	82.00	140.00	90.00	1.00	1.00	0.00	0.00	1.0	1.00
max	99999.00	65.00	2.00	186.00	125.00	180.00	110.00	3.00	3.00	1.00	1.00	1.0	1.00

Figure 2. The base version dataset after data pre-processing

The merely adequate results from the first temp were the motivations for us to go deeper into the dataset. The following preprocessed datasets were all built on the base version, and we aimed to construct more representative features to increase the accuracy. In our general knowledge, we know that we could not determine that a person has cardiovascular disease due to his or her weight or height. Furthermore, both CDC and WHO have announced that the higher BMI a person has, the higher risk he or she has cardiovascular disease, so does for the hypertension. Obesity and hypertension are closer to the cause of presenting cardiovascular disease. Instead of just looking at only attribute height or attribute weight, converting these two attributes into body mass index would be more relevant to reflect the possibility of having cardiovascular disease. It was the same reason for systolic blood pressure and diastolic blood pressure. Therefore, we consulted with health professional websites and created two new features called obesity and hypertension to replace height, weight, systolic blood pressure and diastolic blood pressure. Figure 3 illustrates the dataset after feature engineering.

3. Dataset description (After feature engirneering):

	age	gender	cholesterol	gluc	smoke	alco	active	obesity	hypertension	class
count	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.0	66953.00	66953.00	66953.00
mean	0.46	1.35	1.36	1.22	0.09	0.05	0.8	0.62	0.35	0.49
std	0.50	0.48	0.68	0.57	0.28	0.22	0.4	0.49	0.48	0.50
min	0.00	1.00	1.00	1.00	0.00	0.00	0.0	0.00	0.00	0.00
25%	0.00	1.00	1.00	1.00	0.00	0.00	1.0	0.00	0.00	0.00
50%	0.00	1.00	1.00	1.00	0.00	0.00	1.0	1.00	0.00	0.00
75%	1.00	2.00	1.00	1.00	0.00	0.00	1.0	1.00	1.00	1.00
max	1.00	2.00	3.00	3.00	1.00	1.00	1.0	1.00	1.00	1.00

Figure 3. The revised version dataset after feature engineering

In this version, we simply assign “1” to anomalous and “0” to normal both on obesity and hypertension according to the documentation from CDC and WHO. The age feature enlarged the distance measurement which would have larger impact to KNN based algorithms, so we also divided the samples into two groups according to the high risk age line, 55 years old. We also prepared another version which was that the values of age were combined into four bins with equal width approach, and the values of feature obesity and feature hypertension were divided

into several levels according to its risk level published by CDC. During the testing, we also tried applying dimensionality reduction method, such as Principle Component Analysis, with a 90% threshold on each version of dataset to see its performances.

3. Dataset description (After feature engirneering):

	age	gender	cholesterol	gluc	smoke	alco	active	obesity	hypertension	class
count	66953.00	66953.00	66953.00	66953.00	66953.00	66953.00	66953.0	66953.00	66953.00	66953.00
mean	2.28	1.35	1.36	1.22	0.09	0.05	0.8	0.98	0.93	0.49
std	0.71	0.48	0.68	0.57	0.28	0.22	0.4	0.99	1.34	0.50
min	0.00	1.00	1.00	1.00	0.00	0.00	0.0	0.00	0.00	0.00
25%	2.00	1.00	1.00	1.00	0.00	0.00	1.0	0.00	0.00	0.00
50%	2.00	1.00	1.00	1.00	0.00	0.00	1.0	1.00	0.00	0.00
75%	3.00	2.00	1.00	1.00	0.00	0.00	1.0	2.00	2.00	1.00
max	3.00	2.00	3.00	3.00	1.00	1.00	1.0	4.00	4.00	1.00

Figure 4. Another revised version dataset after feature engineering

3. Classification algorithms

In part 1, we applied some commonly used classification algorithms and some well-known, competitive classification methods on all 6 kinds of preprocessed data with 80% of training and 20% of testing. In each testing, we generated algorithm specific confusion matrix with associated precision, recall, F-1 score and accuracy. The implementation also output Receiver Operating Characteristic curve and its Area Under the Curve value. To obtain a reliable estimate of performance, we did 10-fold cross validation, and had an overall review at the end.

3.1 Naïve Bayes

Naive Bayes algorithm is a probabilistic classifier based on applying Bayes' theorem with independence assumptions between the features. This was one of the first algorithms we thought of due to some of its properties. It is simple, computationally efficient, requires relatively little data for training, and do not have lot of parameters. Moreover, it is naturally robust to missing and noise data of fast speed and well performance in many scenarios.

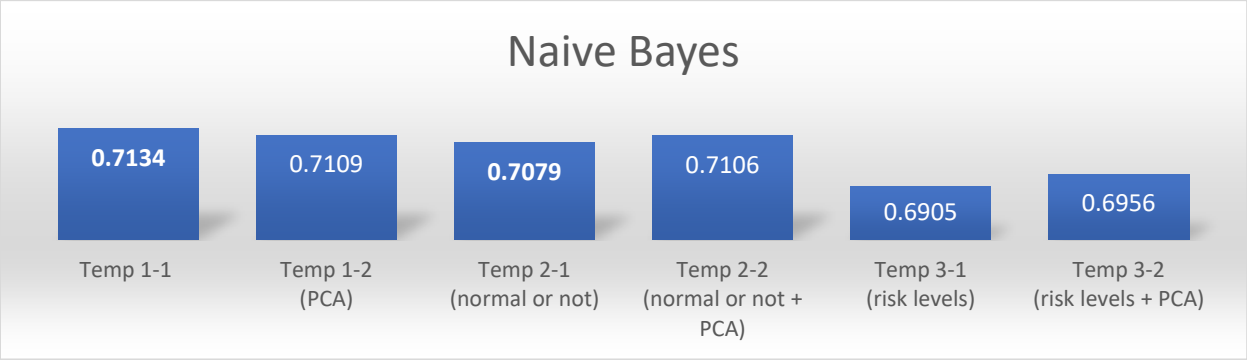

Figure 5-1. 10-fold cross validation accuracies

Based on the conducted experiments, the 10-fold cross validation accuracy results for Naïve Bayes on 6 preprocessed datasets are presented in Figure 5-1. As we can see, they were similar, and the highest accuracy was on the base dataset, the one just done data cleaning. One thing to note is that Naïve Bayes took less than 1 second on each preprocessed dataset, which was extremely fast. Figure 5-2 shows other scores and the ROC curve.

1. Naive Bayes:

Confusion Matrix:

	Actual (+)	Actual (-)
predicted (+)	4053	1243
predicted (-)	2632	5463

Classification report:

	precision	recall	f1-score	support
Negative (-)	0.67	0.81	0.74	6706
Positive (+)	0.77	0.61	0.68	6685
accuracy			0.71	13391
macro avg	0.72	0.71	0.71	13391
weighted avg	0.72	0.71	0.71	13391

10-fold cross validation accuracy: 0.7134

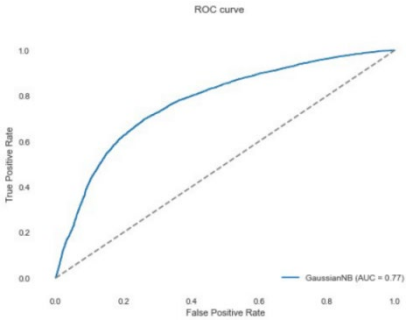

Figure 5-2. Classification report and ROC curve from temp1-1

3.2 Decision trees

Decision trees are a reliable and effective decision making technique that provide high classification accuracy with a simple representation of gathered knowledge. In this model, we got better results of accuracy, but the highest one was still on the base dataset with a max depth of 5 trees and minimum 4 of samples required to be at a leaf node. Figure 6-1 and 6-2 illustrates the brief report for Decision Tree classification.

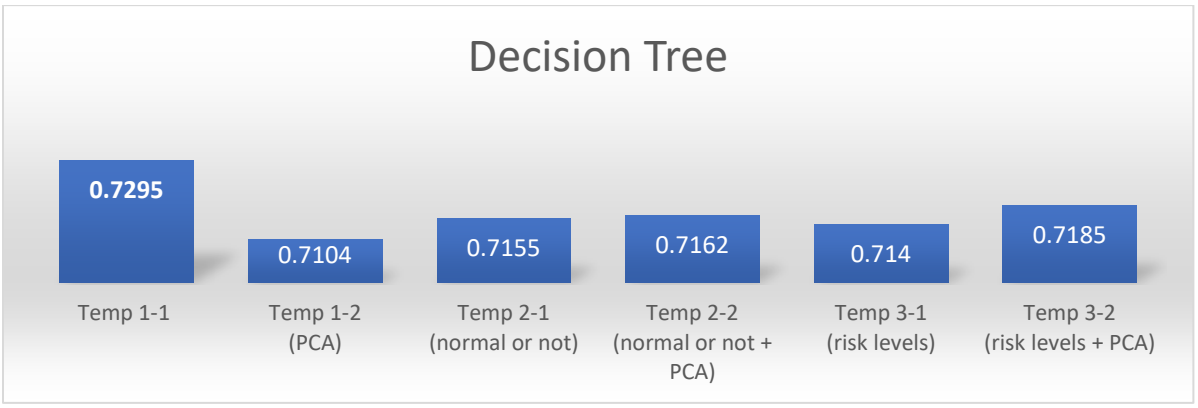

Figure 6-1. 10-fold cross validation accuracies

2. Decision Trees:

Confusion Matrix:

	Actual (+)	Actual (-)
predicted (+)	4149	1119
predicted (-)	2536	5587

Classification report:

	precision	recall	f1-score	support
Negative (-)	0.69	0.83	0.75	6706
Positive (+)	0.79	0.62	0.69	6685
accuracy			0.73	13391
macro avg	0.74	0.73	0.72	13391
weighted avg	0.74	0.73	0.72	13391

10-fold cross validation accuracy: 0.7295

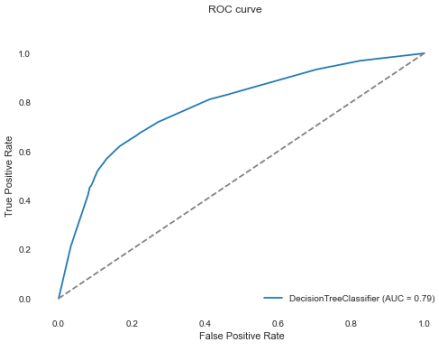

Figure 6-2. Classification report and ROC curve from temp 1-1

3.3 K-Nearest Neighbors

In order to have a better performance, the appropriate k value is needed. If k is too small, sensitive to noise points. If k is too large, neighborhood may include points from other classes. After parameter tuning, we found that when k =17, we got the highest accuracy on temp 2-1 dataset, which is the version simply assigning “1” to anomalous and “0” to normal both on

obesity and hypertension. We successfully prevent distance measures from being dominated by one of the attributes like the problems on Temp 1 datasets.

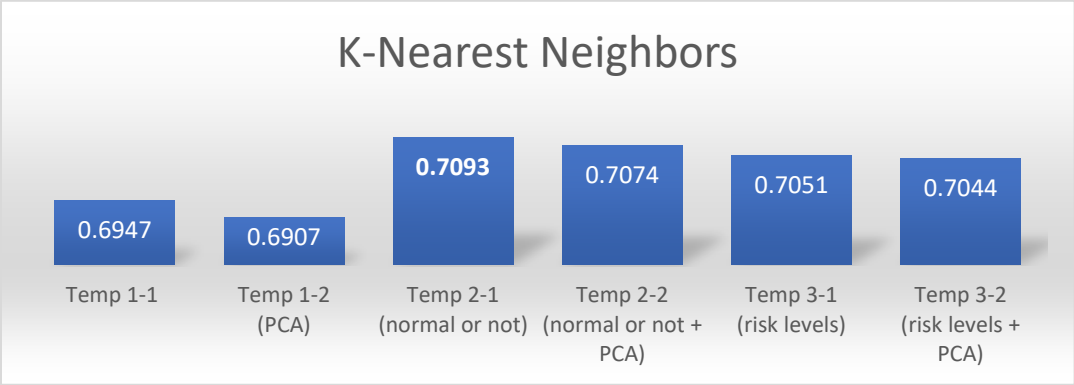

Figure 7-1. 10-fold cross validation accuracies

```
3. K-Nearest Neighbors:

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4127 1390 ]
          (-) [ 2558 5316 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.68      0.79      0.73      6706
Positive (+)       0.75      0.62      0.68      6685

    accuracy                           0.71     13391
   macro avg       0.71      0.71      0.70     13391
weighted avg       0.71      0.71      0.70     13391

10-fold cross validation accuracy: 0.7093
```

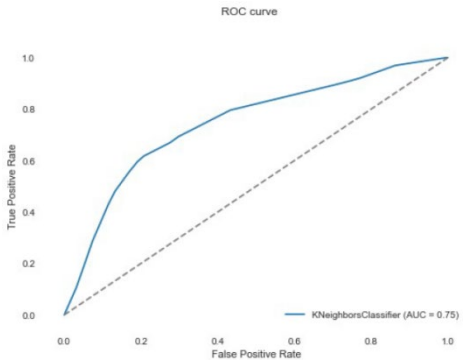

Figure 7-2. Classification report and ROC curve from temp 2-1

3.4 K-Nearest Neighbors (bagging)

Although the preprocessed data did increase the performance of K-Nearest Neighbor algorithm, the results of accuracy were relatively lower. We aimed to use the ensemble method to construct a set of KNN classifiers from the training data, and it worked on most datasets. Bagging improves the generalization error by reducing the variance of the base classifier. The highest one was on the Temp 2-2 dataset with k=17.

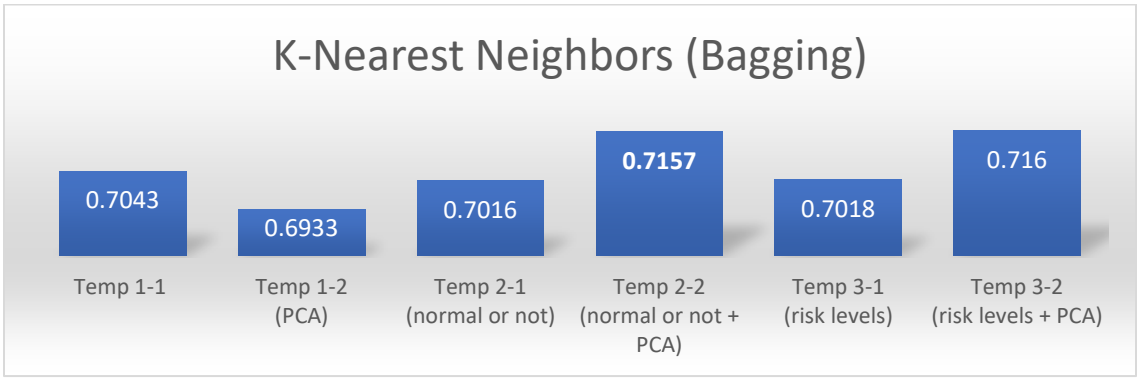

Figure 8-1. 10-fold cross validation accuracies

```
4. K-Nearest Neighbors (Bagging):

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4247 1465 ]
          (-) [ 2438 5241 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.68      0.78      0.73      6706
Positive (+)       0.74      0.64      0.69      6685

    accuracy                           0.71     13391
   macro avg       0.71      0.71      0.71     13391
weighted avg       0.71      0.71      0.71     13391

10-fold cross validation accuracy: 0.7157
```

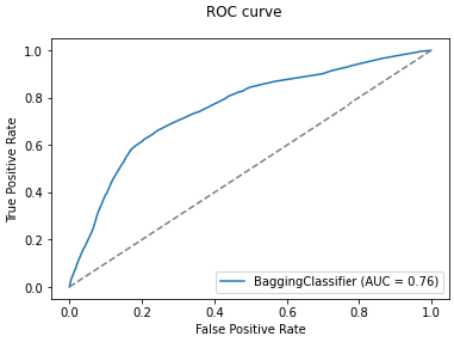

Figure 8-2. Classification report and ROC curve from temp 2-2

3.5 Logistic Regression

A lot of logistic regressions have been performed to identify risk factors for various diseases or for mortality from a particular ailment. It is a statistical technique used when we wish to estimate the probability of a dichotomous outcome. In our case, we wanted to predict the presence or absence of cardiovascular disease. The probability of the outcome is the dependent variable and the various factors that influence it are the risk factors. The highest score was still on the base version dataset with 72.68%.

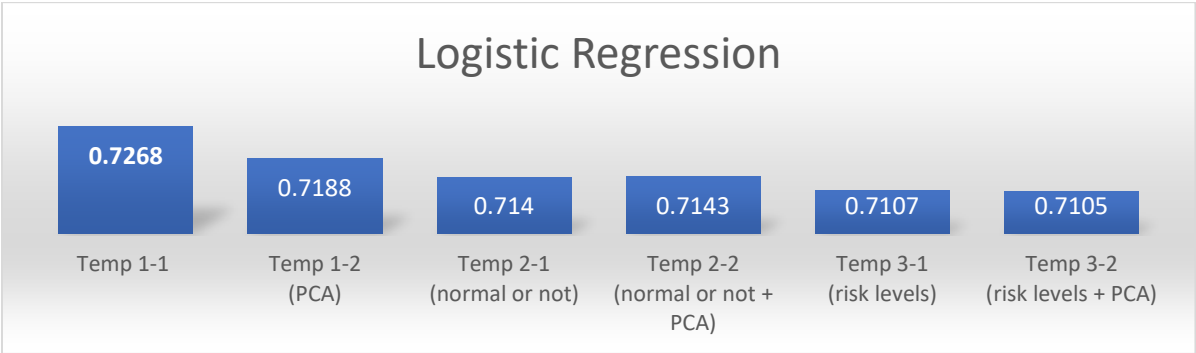

Figure 9-1. 10-fold cross validation accuracies

```
5. Logistic Regression:

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4081 1320 ]
          (-) [ 2604 5386 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.67      0.80      0.73      6706
Positive (+)       0.76      0.61      0.68      6685

    accuracy                           0.71     13391
   macro avg       0.71      0.71      0.70     13391
weighted avg       0.71      0.71      0.70     13391

10-fold cross validation accuracy: 0.7143
```

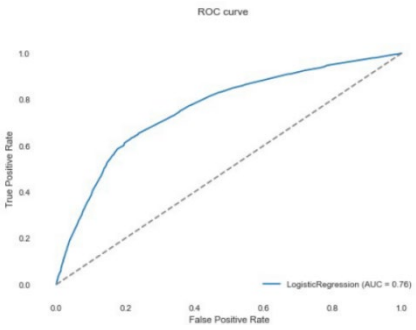

Figure 9-2. Classification report and ROC curve from temp 1-1

3.6 Random Forest

Random forest is one of the most used algorithms because of its simplicity and diversity. It is also an ensemble method that construct a set of Decision tree classifiers. Just like Decision trees, the highest performance was on the base data set with a max depth of 12.

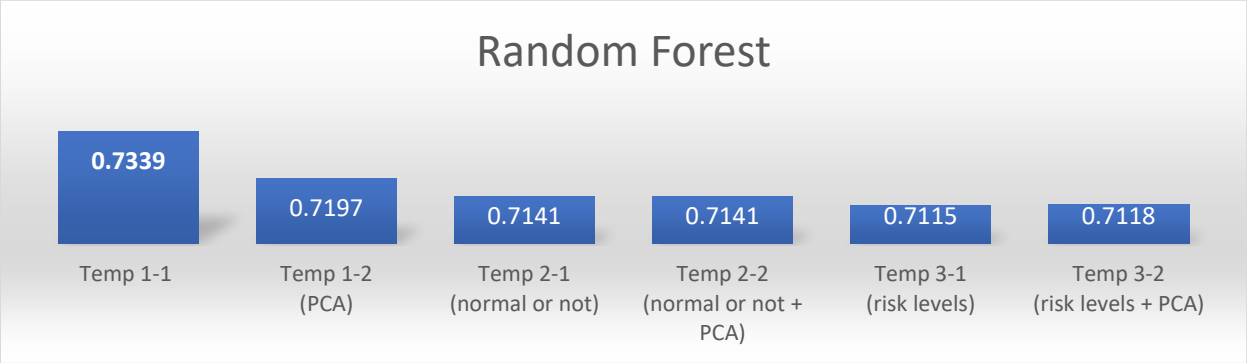

Figure 10-1. 10-fold cross validation accuracies

```
6. Random Forest:

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4448 1394 ]
          (-) [ 2237 5312 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.70      0.79      0.75      6706
Positive (+)       0.76      0.67      0.71      6685

    accuracy                           0.73     13391
   macro avg       0.73      0.73      0.73     13391
weighted avg       0.73      0.73      0.73     13391

10-fold cross validation accuracy: 0.7339
```

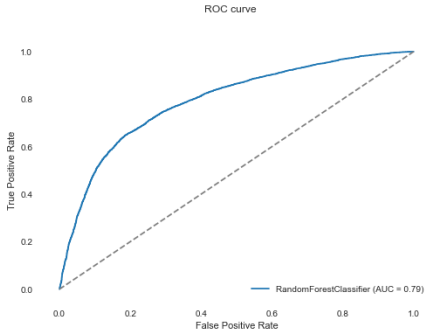

Figure 10-2. Classification report and ROC curve from temp 1-1

3.7 Adaboost

Adaboost is a simple weak classification algorithm promotion process. This process can improve the classification ability of the data through continuous training.

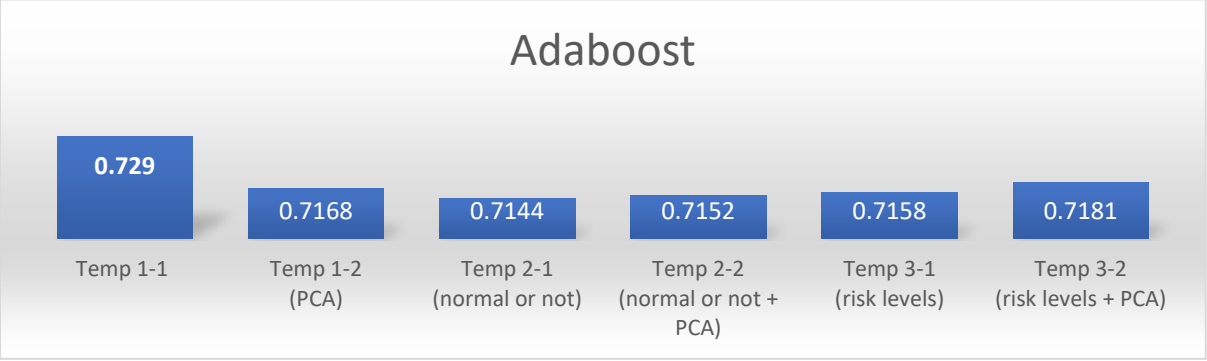

Figure 11-1. 10-fold cross validation accuracies

```
7. Adaboost:

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4262 1262 ]
          (-) [ 2423 5444 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.69      0.81      0.75      6706
Positive (+)       0.77      0.64      0.70      6685

    accuracy                           0.72     13391
   macro avg       0.73      0.72      0.72     13391
weighted avg       0.73      0.72      0.72     13391

10-fold cross validation accuracy: 0.729
```

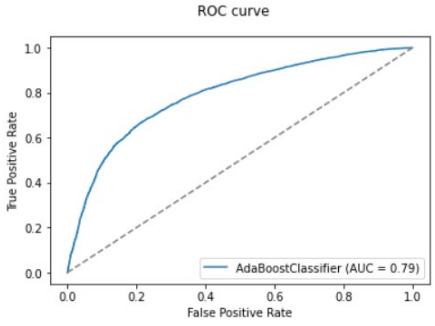

Figure 11-2. Classification report and ROC curve from temp 1-1

3.8 Gradient Boosting Decision Tree

Gradient boosted decision trees algorithm uses boosting method to combine series decision trees to achieve a strong learner from many sequentially connected weak learners. It is highly efficient on classification and more accurate compared to random forests.

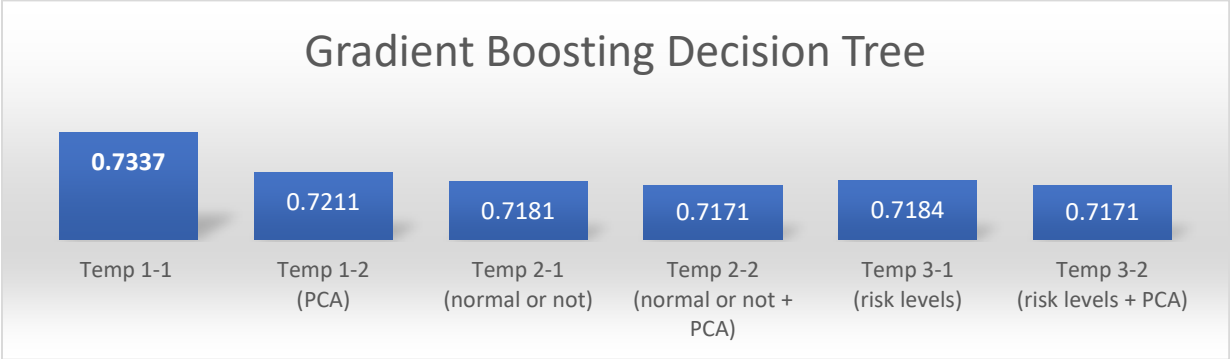

Figure 12-1. 10-fold cross validation accuracies

```
8. Gradient Boosting Decision Tree:

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4531 1445 ]
          (-) [ 2154 5261 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.71      0.78      0.75      6706
Positive (+)       0.76      0.68      0.72      6685

    accuracy                           0.73     13391
   macro avg       0.73      0.73      0.73     13391
weighted avg       0.73      0.73      0.73     13391

10-fold cross validation accuracy: 0.7337
```

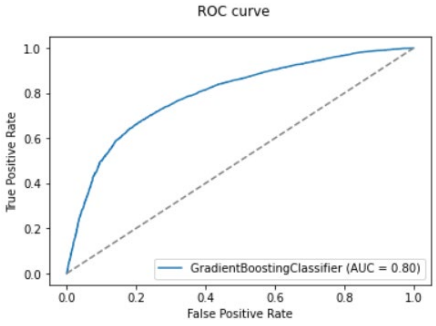

Figure 12-2. Classification report and ROC curve from temp 1-1

3.9 Support Vector Machine (Linear)

Linear SVM is a very useful data mining algorithm for solving classification problems from large data sets with high dimensional data efficiently. However, in our testing it spent most of the time compared to other algorithms with similar results of accuracy.

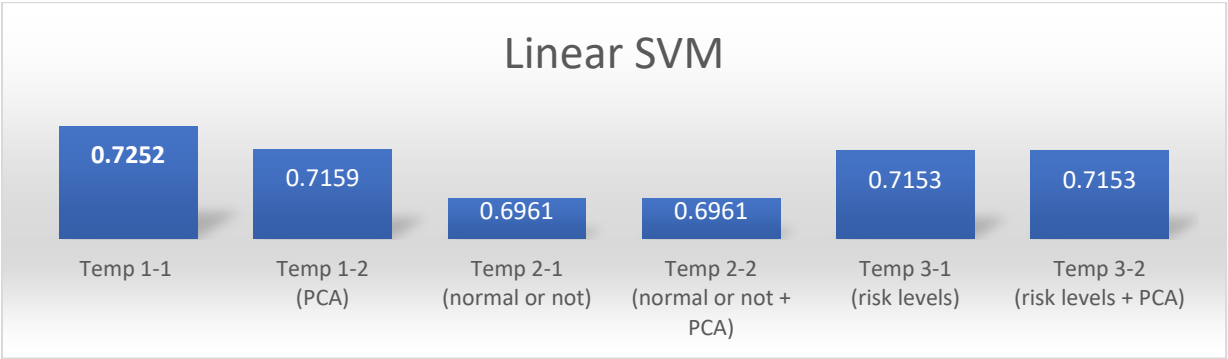

Figure 13-1. 10-fold cross validation accuracies

```
9. Linear SVM:

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4168 1219 ]
          (-) [ 2517 5487 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.69      0.82      0.75      6706
Positive (+)       0.77      0.62      0.69      6685

    accuracy                           0.72     13391
   macro avg       0.73      0.72      0.72     13391
weighted avg       0.73      0.72      0.72     13391

10-fold cross validation accuracy: 0.7252
```

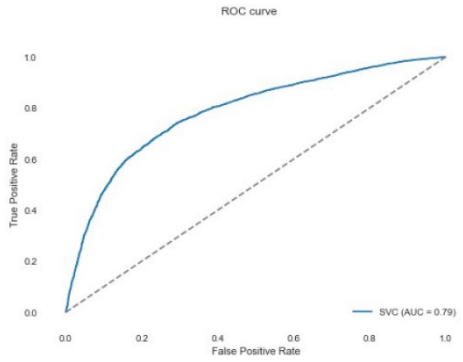

Figure 13-2. Classification report and ROC curve from temp 1-1

3.10 Support Vector Machine (RBF Kernel)

Kernel function maps each sample point to an infinite dimensional feature space, making linearly inseparable data linearly separable. Because support vector machines employing the kernel trick do not scale well to large numbers of training samples or large numbers of features in the input space, and the dataset we used contains relatively large numbers of instances, so we choose RBF kernel.

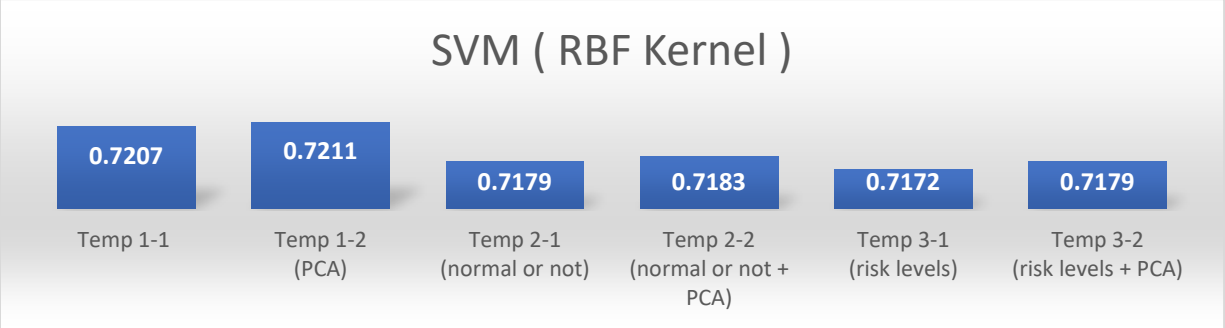

Figure 14-1. 10-fold cross validation accuracies

```
10. SVM (Gaussion Kernel (rbf):

Confusion Matrix:
                Actual
               (+)   (-)
predicted (+) [ 4369 1459 ]
          (-) [ 2316 5247 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.69      0.78      0.74      6706
Positive (+)       0.75      0.65      0.70      6685

    accuracy                           0.72     13391
   macro avg       0.72      0.72      0.72     13391
weighted avg       0.72      0.72      0.72     13391

10-fold cross validation accuracy: 0.7211
```

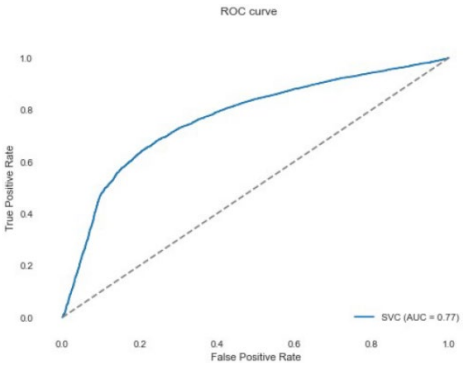

Figure 14-2. Classification report and ROC curve from temp 1-2

3.11 Extreme Gradient Boosting

The XGBOOST algorithm is based on GBDT and Random Forest. This algorithm is widely used in various fields due to its high accuracy, parallelizable processing, and portability. This is also one of the most commonly used methods for Kaggle contestants. This algorithm has higher accuracy, can adapt to the situation of high feature dimension, can effectively prevent overfitting. However, adjusting the parameters is more complicated and time-consuming.

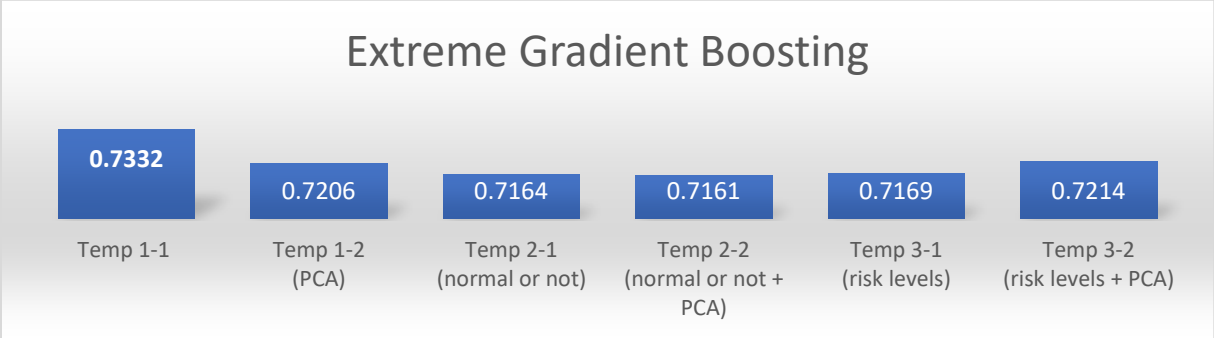

Figure 15-1. 10-fold cross validation accuracies

```
11. Extreme Gradient Boosting:

Confusion Matrix:
                 Actual
                (+)   (-)
predicted (+) [ 4506 1453 ]
          (-) [ 2179 5253 ]

Classification report:
              precision    recall  f1-score   support

Negative (-)       0.71      0.78      0.74      6706
Positive (+)       0.76      0.67      0.71      6685

    accuracy                           0.73     13391
   macro avg       0.73      0.73      0.73     13391
weighted avg       0.73      0.73      0.73     13391

10-fold cross validation accuracy: 0.7332
```

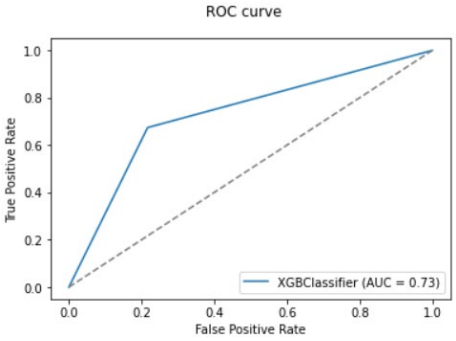

Figure 15-2. Classification report and ROC curve from temp 1-2

4. Comparison and Evaluation

4.1 Accuracy

10-fold cross validation score	Temp 1-1	Temp 1-2 (PCA)	Temp 2-1 (normal or not)	Temp 2-2 (normal or not + PCA)	Temp 3-1 (risk levels)	Temp 3-2 (risk levels + PCA)
Naive Bayes	**0.7134**	0.7109	0.7079	0.7106	0.6905	0.6956
Decision Tree	**0.7295**	0.7104	0.7155	0.7162	0.714	0.7185
K-Nearest Neighbors	0.6947	0.6907	**0.7093**	0.7074	0.7051	0.7044
K-Nearest Neighbors (Bagging)	0.7043	0.6933	0.7016	**0.7157**	0.7018	0.716
Logistic Regression	**0.7268**	0.7188	0.714	0.7143	0.7107	0.7105
Random Forest	**0.7339**	0.7197	0.7141	0.7141	0.7115	0.7118
Adaboost	**0.729**	0.7168	0.7144	0.7152	0.7158	0.7181
Gradient Boosting Decision Tree	**0.7337**	0.7211	0.7181	0.7171	0.7184	0.7171
Linear SVM	**0.7252**	0.7159	0.6961	0.6961	0.7153	0.7153
SVM (RBF Kernel)	**0.7207**	0.7211	0.7179	0.7183	0.7172	0.7179
Extreme Gradient Boosting	**0.7332**	0.7206	0.7164	0.7161	0.7169	0.7214

Figure 16. 10-fold cross validation score for 11 classification methods

The first evaluation for all the classification algorithms we have used is the accuracy of 10-fold cross validation. It is clearly that they all have similar accuracy between 69% and 74%. Our approach to do feature engineering to increase the accuracy was not successful, and there were only two KNN based algorithms got influenced due to the scaling to decrease distance

measurement. Although Random forest classifier is most accurate, the difference of highest three accuracy scores is less than 0.0007. In other words, we should also consider other factors.

4.2 ROC and AUC

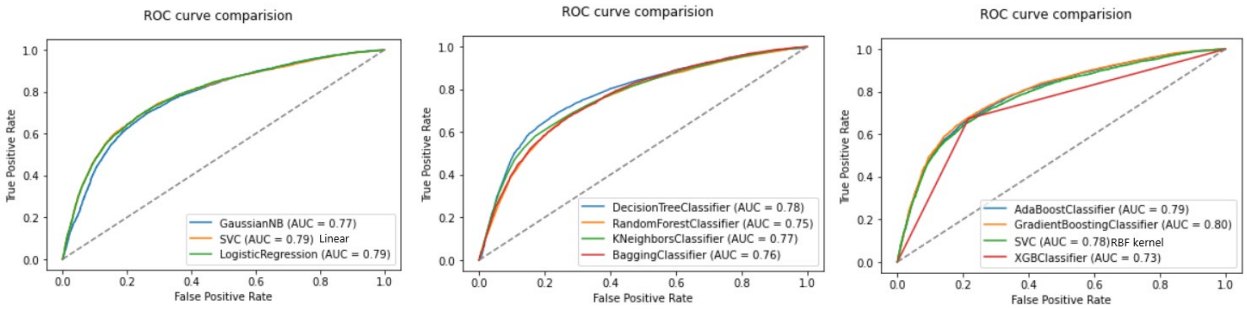

Figure 17. ROC and AUC comparison on Temp 1-1

Despite we did compare all 11 classification approaches for 6 datasets, here we show the ROC curve comparison example from the base dataset. The highest AUC belongs to Gradient Boosting Decision Tree and got 0.8, indicating that its performance was the best. However, some of them still have very close performances. Next, we are going to compare their speed.

4.3 Speed

Execution time (seconds)	Temp 1-1	Temp 1-2 (PCA)	Temp 2-1 (normal or not)	Temp 2-2 (normal or not + PCA)	Temp 3-1 (risk levels)	Temp 3-2 (risk levels + PCA)
Naive Bayes	0.64	0.98	0.58	0.56	0.48	0.57
Decision Tree	1.59	3.01	0.78	1.16	0.84	1.6
K-Nearest Neighbors	20.27	4.83	63.03	8.07	34.84	5.99
K-Nearest Neighbors (Bagging)	42.02	7.75	151.63	16.62	115.47	11.55
Logistic Regression	120.45	3.75	6.55	2.77	8.04	3.52
Random Forest	64.17	158.66	26.8	34.21	28.86	51.67
Adaboost	33.55	47.41	31.7	35.71	31.58	41.88
Gradient Boosting Decision Tree	46.7	92.69	35.95	48.96	35.46	65.3
Linear SVM	15530.31	12177.52	612.97	1604.32	804.76	1688.88
SVM (RBF Kernel)	1210.56	1075.22	1123.28	1287.5	2699.75	2735.15
Extreme Gradient Boosting	239.22	314.88	239.4	62.26	83.91	67.43

Figure 18. Execution time comparison (in seconds)

One thing to note is that in Figure 18, the execution time calculated the time that a specific program ran on the CPU, so it was not the real time passed in reality. In other words, it took longer than the table illustrates, but it is still very helpful. As we can see, most of our time in testing were spent on waiting linear SVM and RBF kernel SVM. If we now consider the top four performance models (Random Forest, Gradient Boosting Decision Tree, Extreme Gradient Boosting, and Decision Tree), Decision Tree is extremely fast. Overall, Gradient Boosting Decision Tree algorithm is also fast with higher accuracy and AUC.