Data Augmentation and Comparison of Model Performances

Hashim Saeed
Department of Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, PA 15213
{hashims}@andrew.cmu.edu

1 Executive Summary

In this project, we explored the problem of data augmentation and its effect on model performances. The idea behind our approach is to somehow create a novel technique which would leverage the data between two dissimilar datasets and help us to increase the classification performance of a model on each individual dataset. The approach was tried and tested on two datasets related to the problem of Heart Disease Risk Detection. Extensive data augmentation evaluation, feature selection and comparative model performance analysis led to the conclusion that our novel approach increases the performance of Random Forest Classifier significantly compared to the baseline in terms of the Area Under the Curve (AUC) metric as well as the Classification Accuracy.

2 Introduction

Heart Diseases have become prevalent issue over the years due to rapid increase in the number of patients. This problem is especially serious in the United States where 610,000 die of heart diseases every year i.e. 1 in every 4 deaths (1). When compared to cancer, another major cause of death in the US, we see that heart diseases result in a greater loss of lives for both men and women as shown in 1.

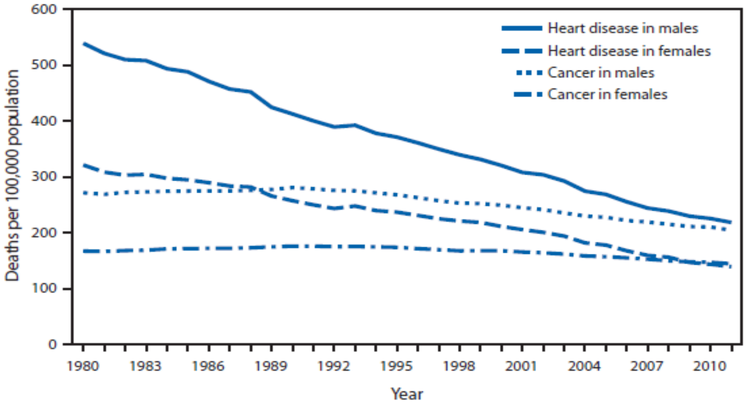


Figure 1: Death-rates by Heart-Diseases and Cancer (2)

Even the youth have become subjected to heart diseases over the years primarily due to increased cholesterol diets and lack of physical activity especially among the teenagers. Many people also experience sudden heart pains even when they have no corresponding physical symptoms beforehand. However, it has been determined that there are several physiological indicators that can help determine heart diseases earlier. Another study has shown that approximately 47% of people suffer heart attack outside the hospitals when they ignore the early warning signs (3) .We wish to use these features to create a novel framework that can help us to detect the risk of heart disease.

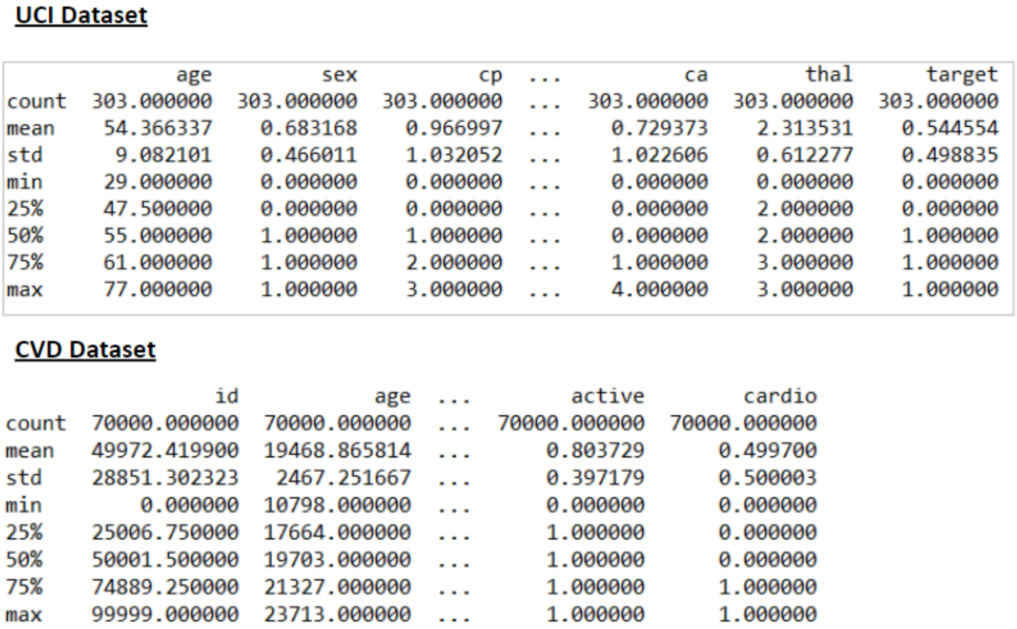
3 Literature Review

Before we started data augmentation process and looked at we can supervised learning to increased the feature space to get better results, we went through pre-existing literature which might have used similar approaches to aid in their tasks of either supervised or unsupervised learning.

The first paper we looked at investigated the concept of lifelong learning (4). Lifelong learning regards the situations where learner faces a whole stream of continuous tasks. The problem, if defined in a mathematical context, is figuring out the $n - th$ task by using $n - 1$ learning tasks to maximise generalization accuracy (4). Thus comes the concept of **Transfer Learning** when the model trained on dataset can be used to predict information for another dataset. In this case, using a classification model on continuous data at each time step t, the author proposes to add the result of the model as another feature in the dataset and feed it as raw input to the next model at time step t+1. In this way, n^{th} learning task will have input data having the original number of features in the data in addition to the $n - 1$ learned predictions of the model. As the paper carries on to show, this allows an increase in the accuracy compared to having the model predicting label at n^{th} step on the original data itself.

The second research we investigated with a similar demonstration of transfer learning in an unsupervised learning application (5). The paper itself focused on Unsupervised Learning and Transfer Learning Challenge on how pre-labelling data using transfer learning increases the performance eventually. The algorithm itself works by first using unsupervised learning on dataset to generate training labels. This is followed by a supervised learning algorithm which generates an additional variable in the new dataset and then the algorithm is repeated again. This shows a rise in performance especially using deep neural networks.

However our problem is different where we wish to utilize 2 independent datasets to somehow increase the performance of a classification model on each independently. Having said this, we first preprocessed the data and applied different augmentation techniques which eventually led to our devised framework.



UCI Dataset

```
              age         sex          cp  ...          ca        thal      target
count  303.000000  303.000000  303.000000  ...  303.000000  303.000000  303.000000
mean    54.366337    0.683168    0.966997  ...    0.729373    2.313531    0.544554
std      9.082101    0.466011    1.032052  ...    1.022606    0.612277    0.498835
min     29.000000    0.000000    0.000000  ...    0.000000    0.000000    0.000000
25%     47.500000    0.000000    0.000000  ...    0.000000    2.000000    0.000000
50%     55.000000    1.000000    1.000000  ...    0.000000    2.000000    1.000000
75%     61.000000    1.000000    2.000000  ...    1.000000    3.000000    1.000000
max     77.000000    1.000000    3.000000  ...    4.000000    3.000000    1.000000
```

CVD Dataset

```
                 id           age  ...        active        cardio
count  70000.000000  70000.000000  ...  70000.000000  70000.000000
mean   49972.419900  19468.865814  ...      0.803729      0.499700
std    28851.302323   2467.251667  ...      0.397179      0.500003
min        0.000000  10798.000000  ...      0.000000      0.000000
25%    25006.750000  17664.000000  ...      1.000000      0.000000
50%    50001.500000  19703.000000  ...      1.000000      0.000000
75%    74889.250000  21327.000000  ...      1.000000      1.000000
max    99999.000000  23713.000000  ...      1.000000      1.000000
```

Figure 2: Summary Statistics of both Data-sets

4 Data-sets

Before we state our methodology as well as our work process, lets look at the 2 data-sets we used for all our experiments and model results. Through research we found two potential data sets that we used to train our machine learning models. Both these datasets are related to patients who do or don't suffer from heart diseases and are very dissimilar. The current data sets we have found are as follows:

1. **Heart Disease UCI Dataset**: This data set is hosted on the Data Science and Competition website *Kaggle*. This data consists of 284 patient entries with 14 features corresponding to each entry. Since this data only contains majority of positive samples, it would be necessary to supplement this data set with negative examples and perform noise addition to create a strong model. This data set can be found at the following site: `https://www.kaggle.com/ronitf/heart-disease-uci`

2. **CardioVascular Disease Dataset**: This data set is also hosted on *Kaggle* contains 3 types of input features; Objective, Examination and Subjective. It has 70,000 patient records and 11 features corresponding to each. Again, data set supplementation may be neccessary as well as matching this with the other datasets for our model to train efficiently. This data can be found at the following site: `https://www.kaggle.com/sulianova/cardiovascular-disease-dataset`

The brief summary of both datasets can be seen in the Figure 2 above. The **target** and **cardio** variable for both datasets refers to whether the patient has heart disease or not.

5 Hardware/Software Tools and Resources

To develop and train the different models we used the **Python** programming language with the **SciKit** Library. The models which we implemented and compared include Naive Bayes, Logistic Regression, SVM, KNN, Decision Trees and Random Forest.

The various feature selection algorithms such as LASSO and correlation can also be implemented using the aforementioned libraries. **Numpy** library can help with additional data processing that we require.

6 Preliminary Data Analysis

After reviewing the summary of both datasets, first step was to extracting features which are common between the two datasets. The selected features so further analysis as well as model training are:

1. **Age**
2. **Gender**
3. **Cholesterol Level in Blood**
4. **Highest Blood Pressure**

After selecting these features, the first step was observing the correlation between these features and the target variables to observe the any linear dependency among them. Figure 3 shows the correlation matrix for both datasets individually.

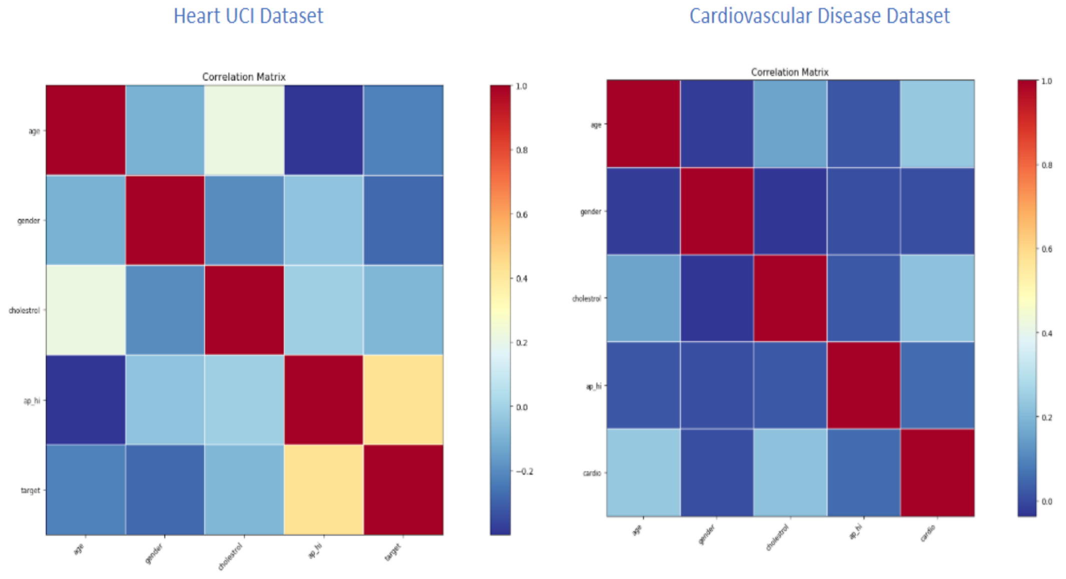


Figure 3: Correlation Matrix

We can see by viewing the correlation matrix that three of the four variables age, gender and cholesterol level have approximately the same correlation value with respect to the target variable. High Blood Pressure has slightly different correlation values for the two datasets.

7 Approach-1: Principal Component Analysis

The first data augmentation approach that we took was using Principal Component Analysis to obtain principal components of the datasets which then can be used as additional features appended to the data on which we could perform classification. PCA was performed on CardioVascular Dataset and each Principal Component was added sequentially to the dataset to see the performance increase in the Area under Curve values for different parameteric and non-parameteric models. After iterative analysis, **two** principal components resulted in best results for various which can be seen in Figure 5.

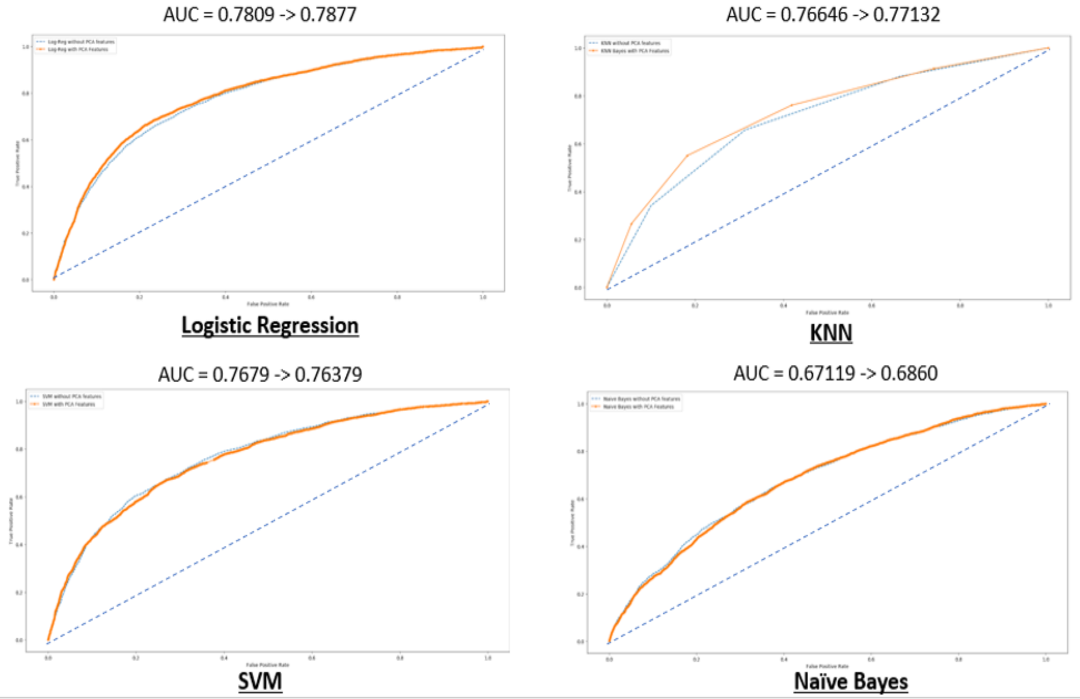


Figure 4: Comparison of PCA on AUC for different classifiers

By looking at the model performances after two principal components were added as additional features, we observe that **greatest increase** in AUC was shown by Naive Bayes while the best performance is recorded by Logisitic Regression. However, if we consider the change itself, the value is 0.009 which is a very small difference. This shows that PCA doesn't contribute much to increasing the predictive ability of each of the models and so we must look somewhere else.

8 Novel Approach: Iterative Feature Addition

Now lets look at our novel approach for data augmentation. We wanted to use the idea of transfer learning by leveraging information in two datasets to train a model which could help us in increasing its performance on each individual dataset. The algorithm is as follows and also shown diagrammatically in Figure **??**:

1. Trained one specific model on the training dataset.
2. Use the trained model to predict the target variable for the testing dataset.
3. The resultant probability is merged with the testing dataset and another model is trained on the new dataset.
4. The new model is used to predict the target variable for the original dataset and the predicted probability is added to the original dataset as an additional feature.
5. Finally, the model is trained and cross-validated on the training dataset itself to see how it performs compared to the model prediction on the original data.

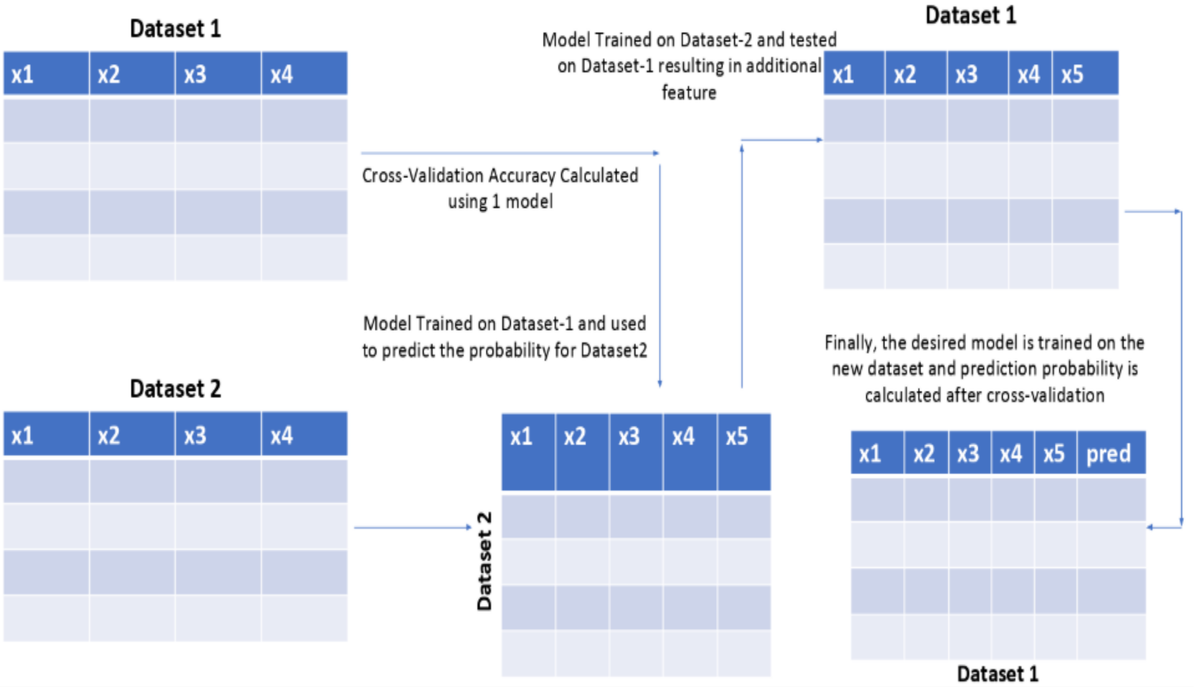

Dataset 1
x1
x2
x3
x4
Model Trained on Dataset-2 and tested on Dataset-1 resulting in additional feature
Dataset 1
x1
x2
x3
x4
x5
Cross-Validation Accuracy Calculated using 1 model
Model Trained on Dataset-1 and used to predict the probability for Dataset2
Finally, the desired model is trained on the new dataset and prediction probability is calculated after cross-validation
Dataset 2
x1
x2
x3
x4
Dataset 2
x1
x2
x3
x4
x5
x1
x2
x3
x4
x5
pred
Dataset 1

Figure 5: Data Augmentation Algorithm

9 Preliminary Results

We used the above-mentioned algorithm once to increase the feature size by 1 and trained various models to compare Area under the Curve values and observe whether this method results in better performance. First we observed the difference for Logistic Regression and K-Nearest Neighbours which showed the best results individually.

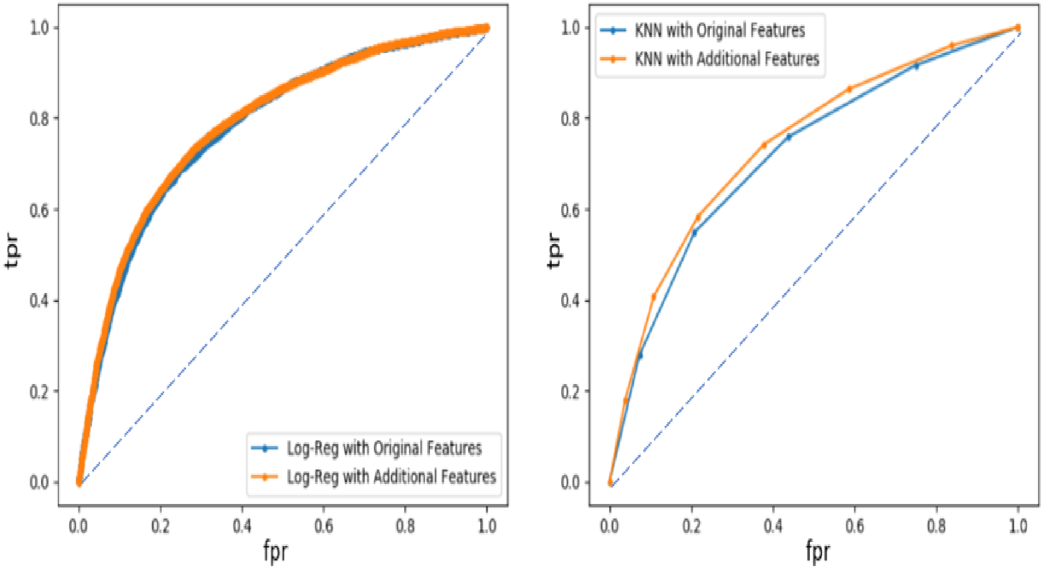
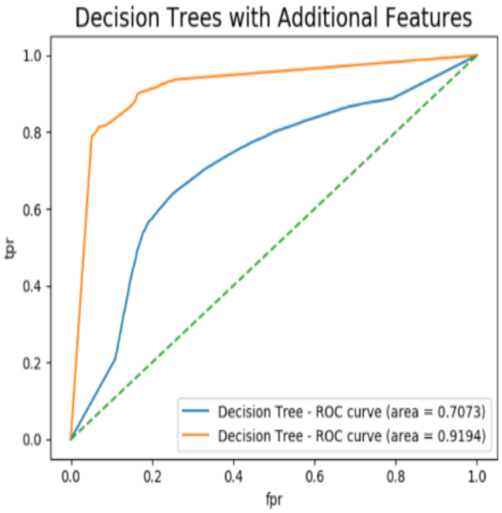


Figure 6: Logistic Regression and KNN Performance after Data Augmentation

As we can observe by looking at the Receiver Operating Curves (ROC) for both classifiers in Figure 6, we again note that this framework didn't result in a substantial increase. However, when we implemented this framework and used Decision Trees as the classifier, we observe the following results.



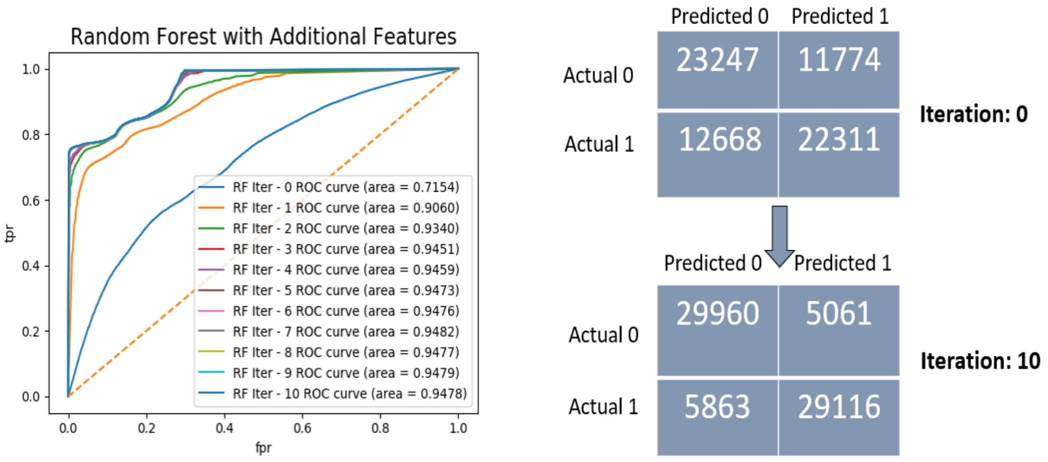
- AUC Score after **10-fold** cross-validation
- Blue Curve represents the initial cross-validated AUC.
- **Orange Curve** represents the AUC of the cross-validated model at the end of the augmentation.
- For all the steps in the process, Decision Tree Classifier was used.
- Parameters: **Leaves = 20, Depth = 20**

Figure 7: Decision Trees Performance with Data Augmentation

Decision Trees show a substantial increase in Area under the Curve value from 0.707 to 0.919 after 10-fold Cross-Validation using our data augmentation algorithm. This concluded that our algorithm was leading to a better model performance and we needed to investigate this process further.

10 Results: Random Forests

After concluding the increase in performance of Decision Trees through the proposed framework, we used our algorithm **iteratively** to increase the number of features at each step and tested the performance of Random Forests at each step to see how they perform.



	Predicted 0	Predicted 1
Actual 0	23247	11774
Actual 1	12668	22311

	Predicted 0	Predicted 1
Actual 0	29960	5061
Actual 1	5863	29116

Figure 8: Random Forest Performance with each Iteration

Figure 8 shows the ROC Curves with their corresponding AUC values as we sequentially increase the number of features using proposed framework. Just a single iteration increases the value of AUC to 0.906 compared to original AUC value of 0.7154. The depth of the tree was selected to be 20 as well number of leaves for optimal results. We can also see the confusion matrix in Figure 8 which conclusively shows an increase in the accuracy in 10^{th} iteration compared to original accuracy. To clearly see how AUC changes as iterations increase, we observe Figure 9 which shows that the although the initial jump is very high, after 3 iterations the algorithm converges to a final value and then the iterative process doesn't result in considerable change.

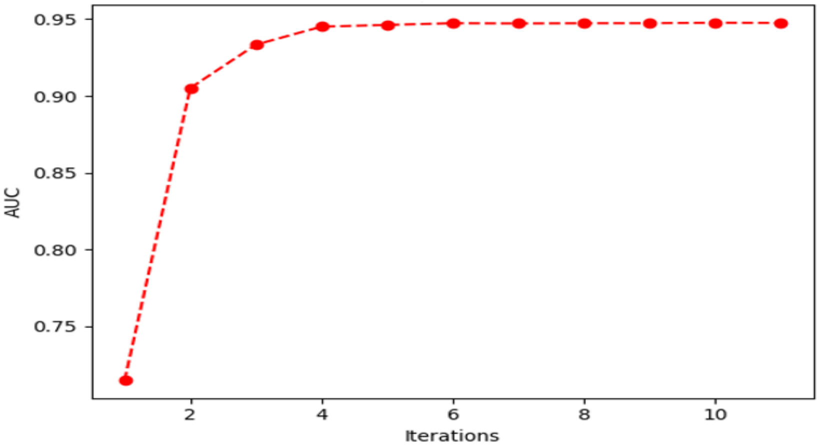


Figure 9: AUC with respect to iteration

Now we observe the correlation between the new variables to see how it changes as we increase the iterations of the framework. Figure 10 shows the correlation matrix as we increase iterations. This clearly concludes that although we are getting a higher performance, the new features become increasing correlated with each other as we increase iterations. This observation can helps us to avoid repeated use of the algorithm and limit number of iterations

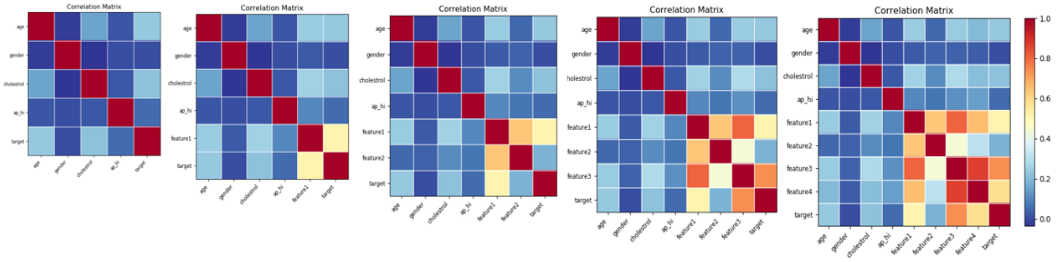


Figure 10: Correlation Matrix as Iterations Increase

11 Reversed Approach

Now we observe how the algorithm performs when we reverse the process i.e. use the algorithm to predict the target variable of the smaller dataset i.e. Heart Disease UCI Dataset. We wish to see how AUC changes as we increase iterations and whether our algorithm can work both ways.

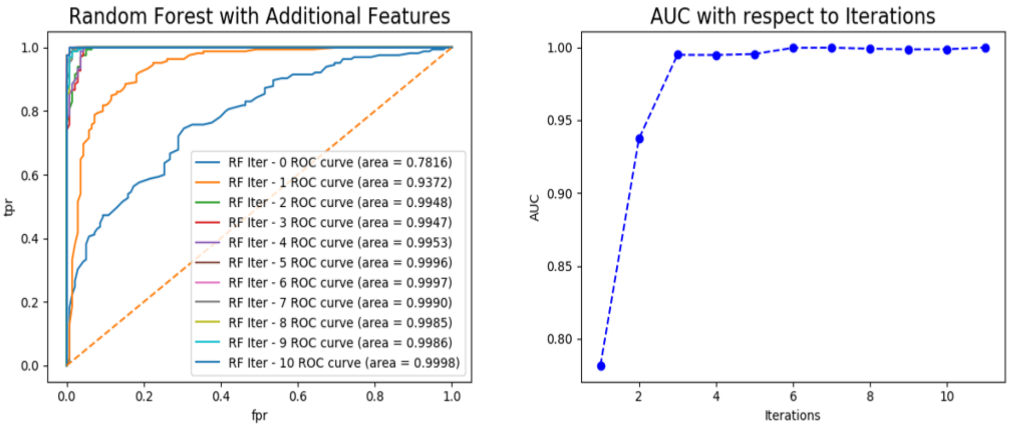
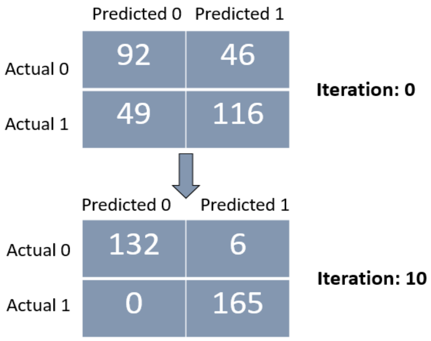


Figure 11: a) ROC Curves per Iteration b) AUC vs Iterations

Our results prove conclusive as we can see in Figure 11 that AUC increases from 0.78 to approximately 1 as iterations increase. The proposed framework can work both ways and is shown to increase the performance of classification model significantly compared to original feature space. The classification accuracy can be seen in Figure 12 which supports our results.



	Predicted 0	Predicted 1
Actual 0	92	46
Actual 1	49	116

Iteration: 0

	Predicted 0	Predicted 1
Actual 0	132	6
Actual 1	0	165

Iteration: 10

Figure 12: a) ROC Curves per Iteration b) AUC vs Iterations

12 Conclusions and Future Work

We have proposed a novel data augmentation algorithm which can help to increase the performance of a classifier such as Random Forest significantly compared to if the classifier was used on the original dataset itself. This framework can be used in situations when we have 2 datasets with some common features and thus help us to leverage the data in two datasets to train the models to perform better than before.

This has great applications especially in practical cases such as Heart Rate Risk Detection problem we discussed in the paper. In most real-life situations, we have a limited dataset at a current time but a large extensive data of the past such as patients that came for past 5 years compared to patients in this month. In this way, leveraging the data in the large dataset and applying out proposed framework, we can get a very high AUC and accuracy on the smaller dataset. There are many other existent problems where we have loads of past data having some different features and so is unused. Our approach leverages that data and helps us to use it to aid in other tasks and increase our predictive performances.

One way this framework could be improved further is to test on very different situations and observe the minimum common elements needed for this framework to succeed. Another aspect which can be further improved is keeping some data from one dataset independent in our framework process and used the trained model to see how it would perform on unseen data. Random Forest can also be further looked into to find the ideal parameters which can boost the performance even more.

References

[1] C. for Disease Control, Prevention, *et al.*, "Underlying cause of death 1999–2013 on cdc wonder online database, released 2015. data are from the multiple cause of death files, 1999–2013, as compiled from data provided by the 57 vital statistics jurisdictions through the vital statistics cooperative program," 2015.

[2] C. for Disease Control, Prevention, *et al.*, "Quickstats: age-adjusted death rates* for heart disease and cancer,† by sex—united states, 1980–2011," *Morbidity and Mortality Weekly Report (MMWR)*, 2014.

[3] C. for Disease Control, P. (CDC, *et al.*, "State-specific mortality from sudden cardiac death–united states, 1999.," *MMWR. Morbidity and mortality weekly report*, vol. 51, no. 6, p. 123, 2002.

[4] S. Thrun, "Is learning the n-th thing any easier than learning the first?," in *Advances in neural information processing systems*, pp. 640–646, 1996.

[5] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36, 2012.