Recognizing Gender of a Human by applying Machine Learning on their Voices

Karan Jain (RedId - 820217249) and Chirag Palesha (RedId - 820915544)

***Abstract*—The project is intended to recognize the gender (male or female) of a person by using their voices. The concept behind it is converting the sample voices into a dataset with multiple features and feeding the information to a machine learning algorithm. Once done that, we test the further voices to know if a machine can distinguish between a male and a female voice. We have applied different machine learning algorithms and models like Logistic Regression, K-means, Random Forest and Support Vector Machine(SVM). We could achieve the maximum accuracy of _____ using ____ algorithm.**

I. Introduction

RECOGNIZING gender of a person, if its male or female, by listening to a persons' voice, from a human ear, is an easy task, considering the listener have good hearing capabilities. But making a computer machine recognizing the gender of a person by using their voice sample is not an easy thing to do. However, if achieved, a system like that would be very helpful to make the communication between machine and human personalized and more interactive.

We decided to take machine learning approach to solve this problem. We have used a dataset of 3,168 records with 20 features and one label (male and female), for the same. This dataset has been formed by applying acoustic analysis on sample voices of 3,168 males and females in .wav formatted.

We have feed this dataset to different machine learning algorithms such as Logistic Regression, K-means, Random Forrest and SVM to make the machine learn about different aspect of human voice and learn if it is male and female so that it can distinguish between male and female voices for further samples. We use python language for the same.

In this report, we have presented the accuracy and learning rate of different machine learning algorithms and models.

II. Data Set

A. *Raw Data*

The sample of 3,168 voices are downloaded from different sources in the .wav format. Since we need the machine to learn about these voices, we must use these .wav files such that it provides us some more information in the form of features. This information can be extracted from .wav files by applying **specan function** from the **WarbleR** R package. Provided the start and end times, specan function measures 20 acoustic parameters on acoustic signals, which later used for acoustic analysis.

B. *Dataset brief*

This CSV contains 3168 rows and 21 columns, consisting of 20 features which we get after pre-processing the .wav files using Specan function from the Warble R package and the 21st column is the Label column which classify the row as male and female.

C. *Features of Dataset*

Following are the features we get after processing the .wav files:

- **meanfreq**: mean frequency (in kHz)
- **sd**: standard deviation of frequency
- **median**: median frequency (in kHz)
- **Q25**: first quantile (in kHz)
- **Q75**: third quantile (in kHz)
- **IQR**: interquantile range (in kHz)
- **skew**: skewness (see note in specprop description)
- **kurt**: kurtosis (see note in specprop description)
- **sp.ent**: spectral entropy
- **sfm**: spectral flatness
- **mode**: mode frequency
- **centroid**: frequency centroid (see specprop)
- **meanfun**: average of fundamental frequency measured across acoustic signal
- **minfun**: minimum fundamental frequency measured across acoustic signal
- **maxfun**: maximum fundamental frequency measured across acoustic signal
- **meandom**: average of dominant frequency measured across acoustic signal
- **mindom**: minimum of dominant frequency measured across acoustic signal
- **maxdom**: maximum of dominant frequency measured across acoustic signal
- **dfrange**: range of dominant frequency measured across acoustic signal
- **modindx**: modulation index.

III. EXPERIMENTS

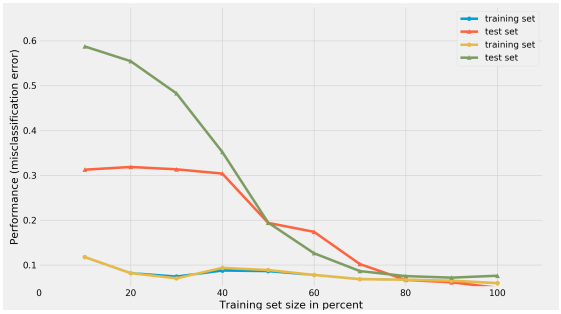
A. Logistic Regression

One of the widely-known machine learning classification algorithm, Logistic Regression predicts the probability of a features dependent labels. The label is a binary variable that contains data coded as 1 (male in this case) or 0 (female in this case). In other words, the logistic regression model predicts P(Y=1) as a function of X.

Input variable: Features from the CSV file
Predict variable (desired target): y—is the voice of a male? (binary: "1", means "Yes", "0" means "No")

In the graphs below, color blue represents validation training data, color orange represents validation test data, color yellow represents test training data and color green represents test data.

Case 1: When C = 10

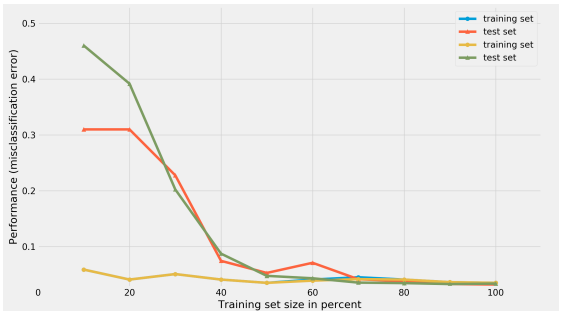


Learning Curve

Validation Data:
Accuracy Score: 0.951515151515
Precision Score: 0.869791666667
Recall Score: 0.982352941176
Confusion Matrix: [[765 50] [6 334]]

Testing Data:
Accuracy Score: 0.924740484429
Precision Score: 0.942465753425
Recall Score: 0.93860845839
Confusion Matrix: [[381 42] [45 688]]

Case 2: When C = 100

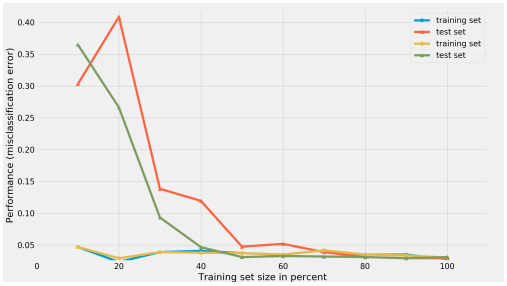


Learning Curve

Validation Data:
Accuracy Score: 0.967965367965
Precision Score: 0.919667590028
Recall Score: 0.976470588235
Confusion Matrix: [[786 29] [8 332]]

Testing Data:
Accuracy Score: 0.966262975779
Precision Score: 0.970189701897
Recall Score: 0.976807639836
Confusion Matrix: [[401 22] [17 716]]

Case 3: When C = 1000



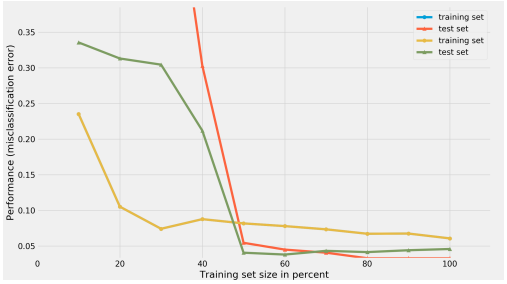
Validation Data:
Accuracy Score: 0.970562770563
Precision Score: 0.932203389831
Recall Score: 0.970588235294
Confusion Matrix: [[791 24] [10 330]]

Testing Data:
Accuracy Score: 0.969723183391
Precision Score: 0.974184782609
Recall Score: 0.978171896317
Confusion Matrix: [[404 19] [16 717]]

Resolving Overfitting Problem:

If we decrease the number of features to only the important features, while feeding the machine, we observed better accuracy comparatively.

With C =1000



Learning Curve

Validation Data:
Accuracy Score: 0.9670995671
Precision Score: 0.917127071823
Recall Score: 0.976470588235
Confusion Matrix: [[785 30] [8 332]]

Testing Data:
Accuracy Score: 0.97058823592
Precision Score: 0.971659919028
Recall Score: 0.982264665757
Confusion Matrix: [[302 21] [13 720]]

Observations:

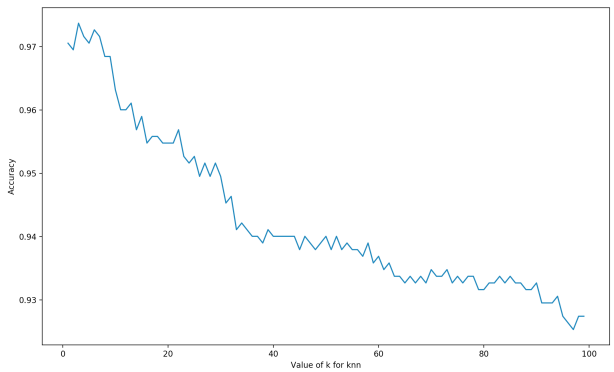
We found the following observation after these experiments:

1. More the regularization (C) better is the accuracy.
2. For our dataset, we observed that when C = 1000, we get most accuracy
3. We observed overfitting, where different features/predictors outputs different accuracy.

B. KNN

To carry on our study with different Machine Learning algorithms and models, we choose to go with K-nearest neighbors algorithm (k-NN). Since our goal is to recognize regularities in our data, so that a machine can efficiently make decisions, we can use k-NN for classification.

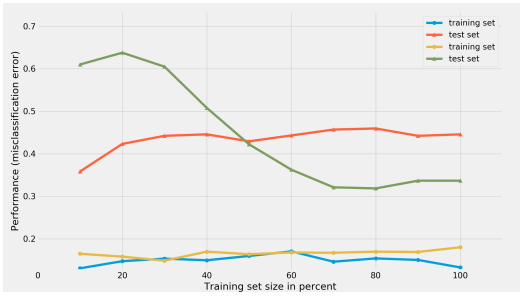
Input variable: Features from the CSV file
Predict variable (desired target): y—is the voice of a male? (binary: "1", means "Yes", "0" means "No")



With k:3 we get most high accuracy of 0.973711882229

When we try to find the accuracy with all the features, it leads to over-fitting.

In the graphs below, color blue represents validation training data, color orange represents validation test data, color yellow represents test training data and color green represents test data.



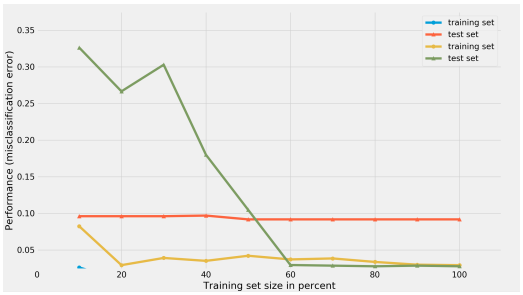
Learning Curve

Validation Data:
Accuracy Score: 0.594805194805
Precision Score: 0.395765472313
Recall Score: 0.714705882353
Confusion Matrix: [[444 371] [97 243]]

Test Data:
Accuracy Score: 0.663494809689
Precision Score: 0.75219941349
Recall Score: 0.699863574352
Confusion Matrix: [[254 169] [220 513]]

Resolving Overfitting Problem:

If we decrease the number of features to only the important features, while feeding the machine, we observed better accuracy comparatively.



Learning Curve

Validation Data:
Accuracy Score: 0.97316017316
Precision Score: 0.969604863222
Recall Score: 0.938235294118
Confusion Matrix: [[805 10] [21 319]]

Test Data:
Accuracy Score: 0.9723183391
Precision Score: 0.974289580514
Recall Score: 0.982264665757
Confusion Matrix: [[404 19] [13 720]]

Observations:

We found the following observation after these experiments:

1. More the regularization better is the accuracy.
2. We observed overfitting, where different features/predictors outputs different accuracy.
3. The more we increase the value of 'k' the lesser the accuracy we get.
4. According to our dataset we get the best accuracy when k=3

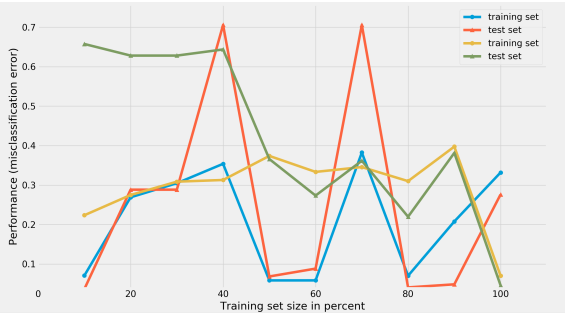
C. Random Forest

To take our experiments further for studying machine learning models and methods we try applying Random Forest machine learning method on our existing dataset. In Random Forest, while constructing a tree, we split the node, which is the best one among a random subset of the features. This leads to the increase in the biasness of the forest but it is more than compensated because of the decrease in variances. Therefore, it is overall a better model. We have studied this algorithm in six different scenarios. First three, by keeping the maximum depth constant and changing the number of trees and other three by keeping the number of trees constant and changing the maximum depth.

Input variable: Features from the CSV file
Predict variable (desired target): y—is the voice of a male? (binary: "1", means "Yes", "0" means "No")

In the graphs below, color blue represents validation training data, color orange represents validation test data, color yellow represents test training data and color green represents test data.

Case 1: When Number of Trees = 1 and Max Depth = 1

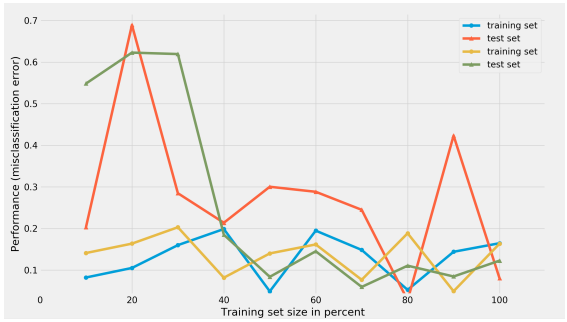


Learning Curve

Validation Data:
Accuracy Score: 0.735930735931
Precision Score: 0.696629213483
Recall Score: 0.182352941176
Confusion Matrix: [[788 27] [278 62]]

Test Data:
Accuracy Score: 0.714532871972
Precision Score: 0.692087702574
Recall Score: 0.990450204638
Confusion Matrix: [[100 323] [7 726]]

Case 2: When Number of Trees = 1 and Max Depth = 3

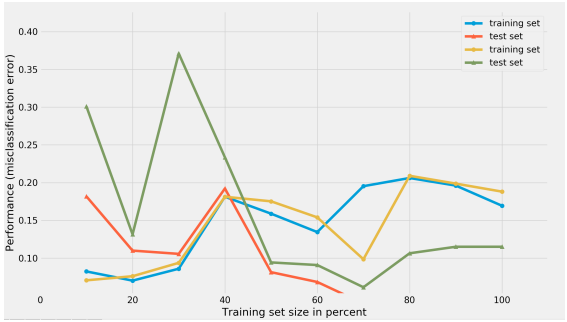


Learning Curve

Validation Data:
Accuracy Score: 0.891774891775
Precision Score: 0.768079800499
Recall Score: 0.905882352941
Confusion Matrix: [[722 93] [32 308]]

Test Data:
Accuracy Score: 0.886678200692
Precision Score: 0.867970660147
Recall Score: 0.968622100955
Confusion Matrix: [[315 108] [23 710]]

Case 3: When Number of Trees = 100 and Max Depth = 1

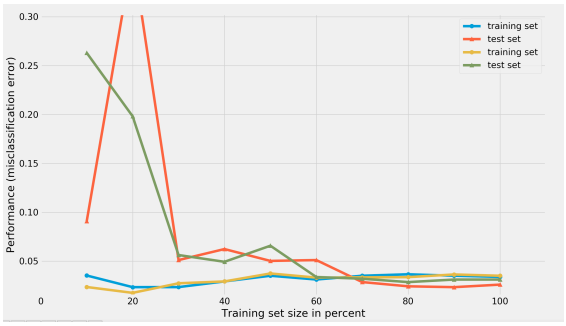


Learning Curve

Validation Data:
Accuracy Score: 0.972294372294
Precision Score: 0.966666666667
Recall Score: 0.938235294118
Confusion Matrix: [[804 11] [21 319]]

Test Data:
Accuracy Score: 0.897923875433
Precision Score: 0.864768683274
Recall Score: 0.994542974079
Confusion Matrix: [[309 114] [4 729]]

Case 4: When Number of Trees = 100 and Max Depth = 3

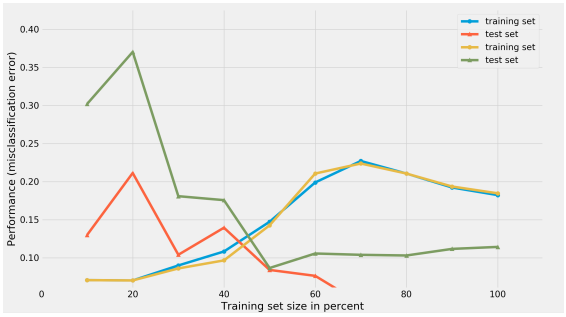


Learning Curve

Validation Data:
Accuracy Score: 0.981818181818
Precision Score: 0.984802431611
Recall Score: 0.952941176471
Confusion Matrix: [[810 5] [16 324]]

Test Data:
Accuracy Score: 0.969723183391
Precision Score: 0.965333333333
Recall Score: 0.987721691678
Confusion Matrix: [[397 26] [9 724]]

Case 5: When Number of Trees = 1000 and Max Depth = 1

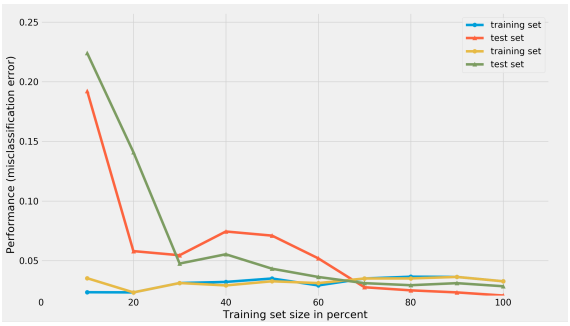


Learning Curve

Validation Data:
Accuracy Score: 0.971428571429
Precision Score: 0.969418960245
Recall Score: 0.932352941176
Confusion Matrix: [[805 10] [23 317]]

Test Data:
Accuracy Score: 0.886678200692
Precision Score: 0.851635514019
Recall Score: 0.994542974079
Confusion Matrix: [[296 127] [4 729]]

Case 6: When Number of Trees = 1000 and Max Depth = 3



Learning Curve

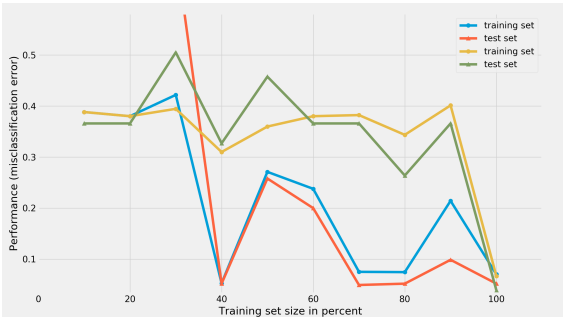
Validation Data:
Accuracy Score: 0.978354978355
Precision Score: 0.984615384615
Recall Score: 0.941176470588
Confusion Matrix: [[810 5] [20 320]]

Test Data:
Accuracy Score: 0.971453287197
Precision Score: 0.969168900804
Recall Score: 0.986357435198
Confusion Matrix: [[400 23] [10 723]]

Resolving Overfitting Problem:

If we decrease the number of features to only the important features, while feeding the machine, we observed better accuracy comparatively.

With Number of Trees = 1000 and Max Depth = 3



Learning Curve

Validation Data:
Accuracy Score: 0.978354978355
Precision Score: 0.984615384615
Recall Score: 0.941176470588
Confusion Matrix: [[810 5] [20 320]]

Test Data:
Accuracy Score: 0.971453287197
Precision Score: 0.969168900804
Recall Score: 0.986357435198
Confusion Matrix: [[400 23] [10 723]]

Observations:

We found the following observation after these experiments:
1. More the regularization better is the accuracy.
2. We observed overfitting, where different features/predictors outputs different accuracy.
3. We observe that when number of trees are less we get higher rate of error.
4. Maximum depth of tree affects accuracy, the more the depth the better the accuracy.
5. The best accuracy we get is by having 100 number of trees and 3 max. depth

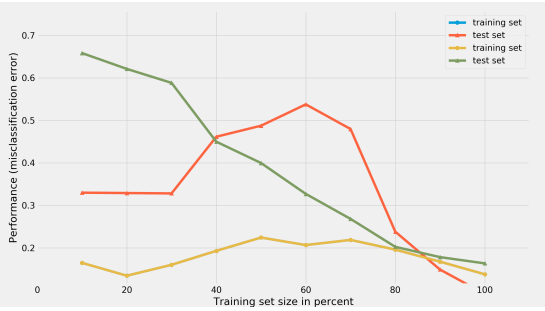
D. Support Vector Machine (SVM)

To conclude our study on machine learning algorithms and models we choose SVM as our last element of experiment. Since the dataset, we have is labeled against the features. We have taken a supervised approach and studied six different scenarios. First three, by keeping the Gamma constant and changing the kernel type and other three by keeping the kernel type constant and changing the gamma.

Input variable: Features from the CSV file
Predict variable (desired target): y—is the voice of a male? (binary: "1", means "Yes", "0" means "No")

In the graphs below, color blue represents validation training data, color orange represents validation test data, color yellow represents test training data and color green represents test data.

Case 1: When C = 1 and Kernel = Linear

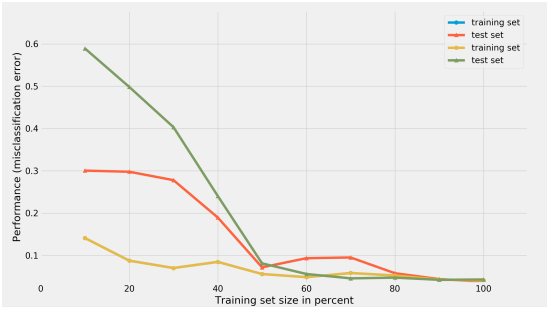


Learning Curve

Validation Data:
Accuracy Score: 0.908225108225
Precision Score: 0.782608695652
Recall Score: 0.952941176471
Confusion Matrix: [[725 90] [16 324]]

Test Data:
Accuracy Score: 0.836505190311
Precision Score: 0.869565217391
Recall Score: 0.87312414734
Confusion Matrix: [[327 96] [93 640]]

Case 2: When C = 5 and Kernel = Linear

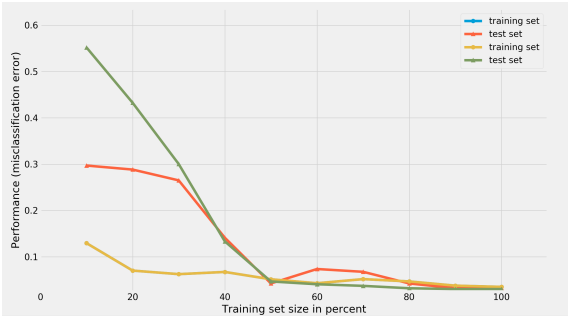


Learning Curve

Validation Data:
Accuracy Score: 0.961904761905
Precision Score: 0.895721925134
Recall Score: 0.985294117647
Confusion Matrix: [[776 39] [5 335]]

Test Data:
Accuracy Score: 0.956747404844
Precision Score: 0.962110960758
Recall Score: 0.969986357435
Confusion Matrix: [[395 28] [22 711]]

Case 3: When C = 10 and Kernel = Linear

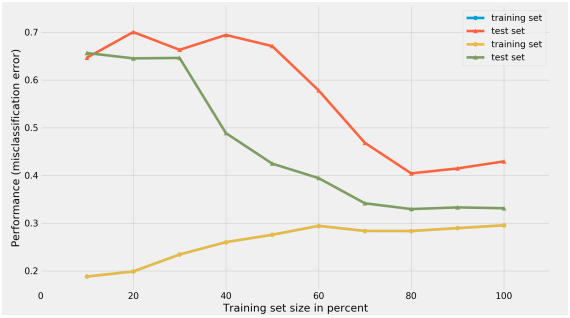


Learning Curve

Validation Data:
Accuracy Score: 0.967965367965
Precision Score: 0.915068493151
Recall Score: 0.982352941176
Confusion Matrix: [[784 31] [6 334]]

Test Data:
Accuracy Score: 0.969723183391
Precision Score: 0.975476839237
Recall Score: 0.976807639836
Confusion Matrix: [[405 18] [17 716]]

Case 4: When C = 1 and Kernel = rbf

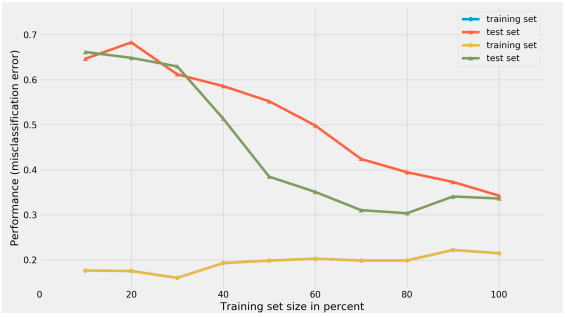


Learning Curve

Validation Data:
Accuracy Score: 0.570562770563
Precision Score: 0.393442622951
Recall Score: 0.847058823529
Confusion Matrix: [[371 444] [52 288]]

Test Data:
Accuracy Score: 0.668685121107
Precision Score: 0.734584450402
Recall Score: 0.74761255116
Confusion Matrix: [[225 198] [185 548]]

Case 5: When C = 5 and Kernel = rbf

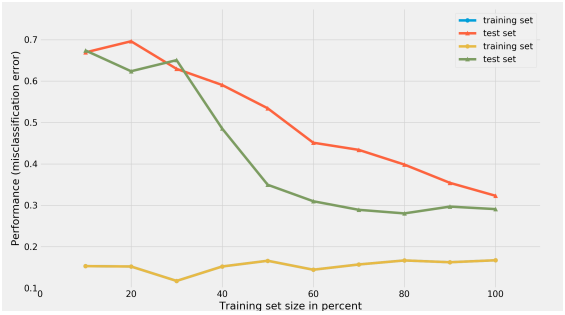


Learning Curve

Validation Data:
Accuracy Score: 0.657142857143
Precision Score: 0.453333333333
Recall Score: 0.8
Confusion Matrix: [[487 328] [68 272]]

Test Data:
Accuracy Score: 0.663494809689
Precision Score: 0.763803680982
Recall Score: 0.679399727149
Confusion Matrix: [[269 154] [235 498]]

Case 6: When C = 10 and Kernel = rbf



Learning Curve

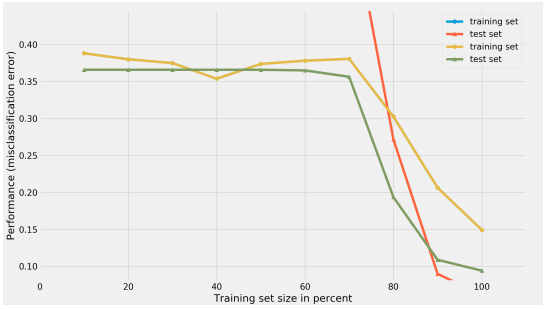
Validation Data:
Accuracy Score: 0.677056277056
Precision Score: 0.471204188482
Recall Score: 0.794117647059
Confusion Matrix: [[512 303] [70 270]]

Test Data:
Accuracy Score: 0.709342560554
Precision Score:0.79234167894
Recall Score: 0.733969986357
Confusion Matrix: [[282 141] [195 538]]

Resolving Overfitting Problem:

If we decrease the number of features to only the important features, while feeding the machine, we observed better accuracy comparatively.

With C =10 and Kernel = Linear



Learning Curve

Validation Data:
Accuracy Score: 0.967965367965
Precision Score: 0.915068493151
Recall Score: 0.982352941176
Confusion Matrix: [[784 31] [6 334]]

Test Data:
Accuracy Score: 0.969723183391
Precision Score: 0.975476839237
Recall Score: 0.976807639836
Confusion Matrix: [[405 18] [17 716]]

Observations:

We found the following observation after these experiments:
1. More the regularization better is the accuracy.
2. We observed overfitting, where different features/predictors outputs different accuracy.
3. We observe that in comparison to linear kernel, rbf kernel has less accuracy.
4. For our dataset we get best accuracy with linear kernel and C=10.

IV. Conclusion

Best accuracy of Logistic Regression – 0.970588235294, after resolving overfitting issue and with C=1000.

Best accuracy of K-NN – 0.97316017316, after resolving overfitting issue and with k = 3.

Best accuracy of Random Forest – 0.978354978355, with number of trees=1000 and maximum depth = 3.

Best accuracy of SVM – 0.967965367965, after resolving overfitting issue and with C =10 and Kernel = Linear.

After conducting different experiments and referring to those we concluded that the model which gave us better result is Random Forest.

We also observed that, in all the models most of the failure cases occur when mode is ~0.13, minfun is ~0.21, maxdom is ~6.7, median is ~0.9 and meanfun is ~1.8.

Also, we have noticed that resolving an overfitting problems plays significant role in most of the models.

V. References

The Harvard-Haskins Database of Regularly-Timed Speech

Telecommunications & Signal Processing Laboratory (TSP) Speech Database at McGill University, Home

VoxForge Speech Corpus, Home

Festvox CMU_ARCTIC Speech Database at Carnegie Mellon University

Scikit-learn documentation to implements different algorithms using python.

VI. Team Member Contribution

Both, Karan Jain and Chirag Palesha have equally contributed when it comes to project idea brainstorming, creating a blueprint and editing the report.

Chirag Palesha: Took the lead and started examining the dataset that we downloaded. He later started conducting experiments like that of Logistic Regression and KNN, and noted his insights in the final project report. He later also cross verified the results concluded by Karan Jain, for Random Forest and Support Vector Machine methods.

Karan Jain: He have looked for better suitable platform for conducting the experiments, which is Python. He later started conducting experiments like that of Random Forest and Support Vector Machine, and noted his insights in the final project report. He later also cross verified the results concluded by Chirag Palesha, for Logistic Regression and K-NN methods.