Machine Learning Final Class Project

CAP6610

University of Florida, Department of Computer Science and Engineering

Surya Sudharshan, UFID:5019-3163

Ruyue Yuan, UFID:9718-6171

Kendall Parker, UFID:9365-5436

Mehrdad Alvandipour, UFID: Unknown

Part-1

Our Approach

Feature Extraction from the Data:

After organizing and pre-processing the dataset, the set of musical features representing timbral, rhythmic, and dynamic and pitch information of music was extracted from the audio using the **Librosa** library. The **Librosa** library is an integrated set of functions written in Python computing environment, which offers intuitive syntax for extracting a wide range of musical features as well as performing statistical analysis.

There exists a myriad number of options to select features on audio signals. Here we show just a few features and there respective meanings, that are candidates for our potential feature set.

Feature	*Description*	*Summarization method*
Spectral Rolloff	*The frequency below which 85% of total energy of the spectrum is contained*	Mean, SD, Slope, PF, PA, PE
Spectral Brightness	*Measures the amount of energy above 1500Hz*	Mean, SD, Slope, PF, PA, PE
Spectral Roughness	*Is estimated by computing the peaks of the spectrum and taking the average of all dissonances between all possible pairs of peaks*	Mean, SD, Slope, PF, PA, PE
MFCC (13 coefficients)	*See table 2 for description. In this work first order differences, or delta MFCC-s are used*	Mean, SD, Slope, PF, PA, PE
Spectral Flux	*The distance between the spectra of successive frames*	Mean, SD, Slope, PF, PA, PE
Regularity	*Measures the degree of variation of the successive peaks of the spectrum*	Mean, SD, Slope, PF, PA, PE
Centroid	*Estimates the geometric center of the spectral distribution*	Mean, SD, Slope, PF, PA, PE
RMS	*Root Mean Square energy of the signal*	Mean, SD, Slope, PF, PA, PE
Low energy	*Mean of the low energy value over all frames*	*Mean*
Pulse clarity	*The strength of the main beat. Estimates rhythmic clarity of the track. Calculated by finding the maximum correlation value from the autocorrelation function of the onset detection curve.* (Latrillot,Eerola,Toiviainen & Fornari, 2008)	*Mean*
Chromagram (12 pitch classes)	*Shows the distribution of spectral energy along the pitch classes.*	Mean, SD, Slope, PF, PA, PE
Zero-crossing rate	*Counts the amount of sign changes of the signal*	Mean, SD, Slope, PF, PA, PE

Table showing some features and their descriptions

After doing extensive research on the internet about what would constitute as good features and how many good features will be necessary to build a good model, we arrived at these features as the best suitors for the data after considering the feature selection algorithms implemented by **Valeri Tsatsishvili** in her thesis on Automatic Subgenre Classification of Heavy Metal Music and **Parul Pandey's** blog on features for classification of music based on genres.

Correlation based feature selection (CFS)	Wrapper with J48 (W-J48)	Wrapper with KNN (W-KNN)
Spectral Rolloff	*Spectral Rolloff*	*Spectral Rolloff*
Brightness	*Chromagram*	*Brightness*
Centroid	*Mfcc*	*Mfcc*
Mfcc	*RMS*	*Chromagram*
Zerocrossing rate	*Flux*	
Chromagram	*Zerocrossing rate*	
	Regularity	

The three feature selection algorithms implemented by Valeri Tsatsishvili for music classification, combined with the suggestions put forth by Parul Pandey, we chose a common feature set from these 3 feature selection algorithms combined with a few other parameters to make for a **27 dimensional feature space**

- *Zero Crossing Rate:*

  The zero crossing rate is the rate of sign-changes along a signal, i.e., the rate at which the signal changes from positive to negative or back. This feature has been used heavily in both speech recognition and music information retrieval. It usually has higher values for highly percussive sounds like those in metal and rock.
- *Spectral Centroid*:

  It indicates where the ”centre of mass” for a sound is located and is calculated as the weighted mean of the frequencies present in the sound. Consider two songs, one from a blues genre and the other belonging to metal. Now as compared to the blues genre song which is the same throughout its length, the metal song has more frequencies towards the end. So spectral centroid for blues song will lie somewhere near the middle of its spectrum while that for a metal song would be towards its end.
- *Spectral Rolloff*:

  It is a measure of the shape of the signal. It represents the frequency below which a specified percentage of the total spectral energy, e.g. 85%, lies.
- *Mel-Frequency Cepstral Coefficients*:

  The Mel frequency cepstral coefficients (MFCCs) of a signal are a small set of features (usually about 10–20) which concisely describe the overall shape of a spectral

envelope. It models the characteristics of the human voice. We are considering all 20 MFCCs for our models. Therefore MFCC feature that we will be using is a 20 dimensional feature.

- *Chroma Frequencies*:

  Chroma features are an interesting and powerful representation for music audio in which the entire spectrum is projected onto 12 bins representing the 12 distinct semitones (or chroma) of the musical octave.

- *Root Mean Squared Error:*

  It computes the root mean square value for each frame from the audio samples or the spectrogram. We choose to extract the root mean square error from the audio samples because its faster as it doesn't require a STFT calculation

Feature Pre Processing:

The above features have a broad spectrum range of values for all the songs present in our training, validation and test set. For our convenience and the models' convenience, we have scaled the data using the **MinMaxScaler** provided by **sklearn.**

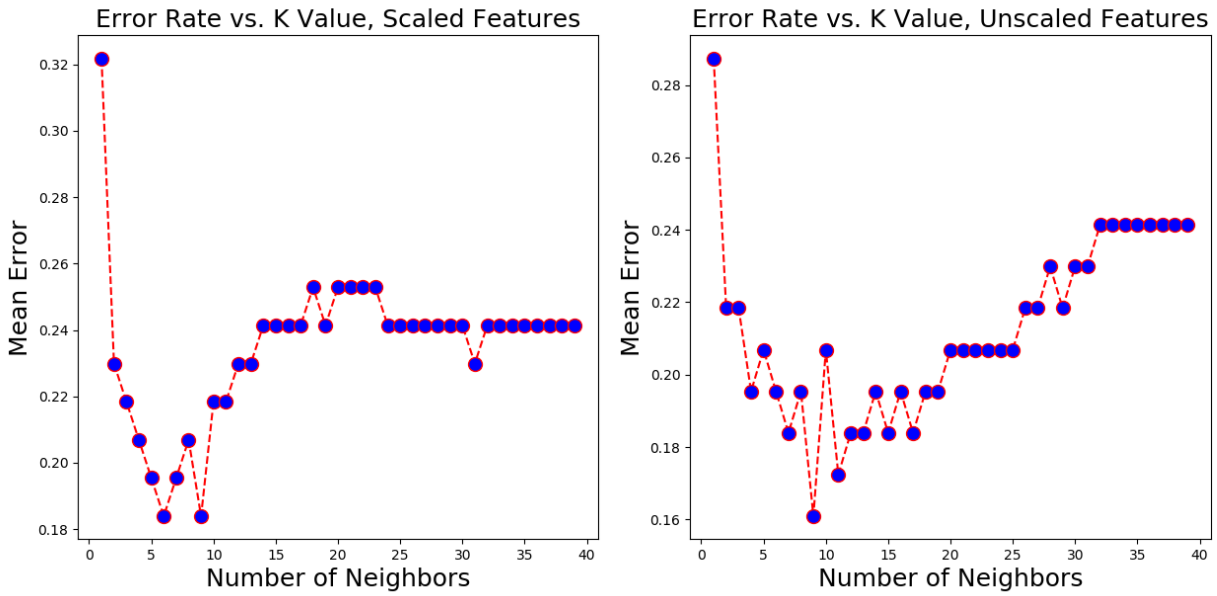
A preliminary assessment was first performed to see if standardization improved accuracy of results and if it impacted the optimal performance of the models. Standardization was performed on all datasets with the following equation. The **MinMaxScaler** from sklearn was used because it preserves the shape of the original distribution without meaningfully changing the information embedded in the original data

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

The scaling has been chosen as an option to 4 out of the 5 models that we have implemented. We have experimented with scaling the features and without scaling the features and we present a detailed explanation of our experiments.

K-Nearest Neighbors

Due to its intuitive nature and simple implementation, the K-Nearest Neighbors (KNN) Algorithm for classification was also explored for this project. KNN uses instance-based learning, and without an explicit training phase, the calculation time is much faster than its classification counterparts such as random forest and logistic regression. There is one hyperparameter to tune, and that is the number of nearest neighbors. Additionally, the impact of standardizing the feature vectors and balancing the training data can be explored to compare accuracy results. Homogeneity of features is necessary because of the distance metric needed to compute neighbors. Features must have the same scale since absolute differences in features weight the same.

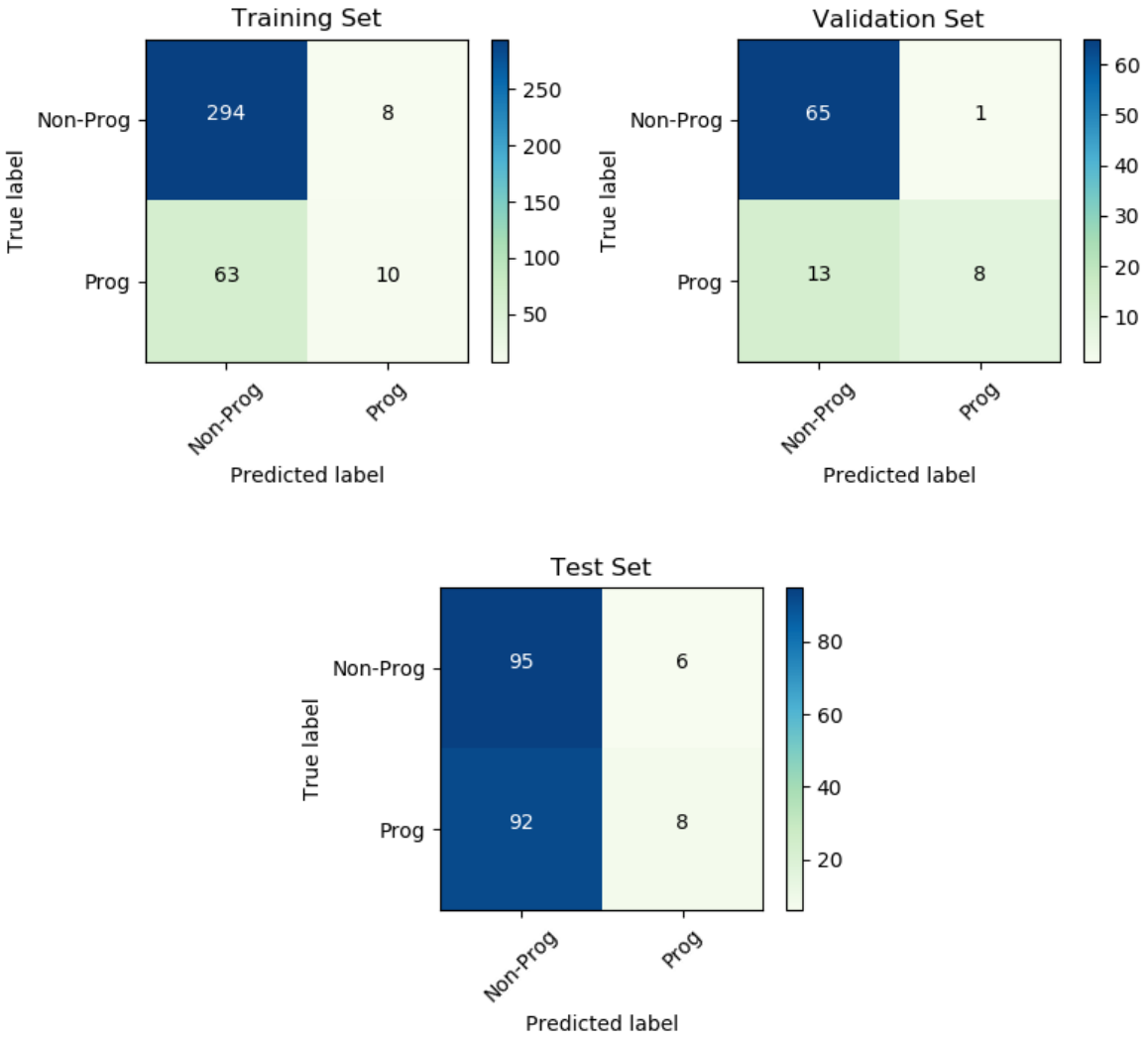


Assessing Optimal Number of Neighbors for Unbalanced Data Based on Validation Set

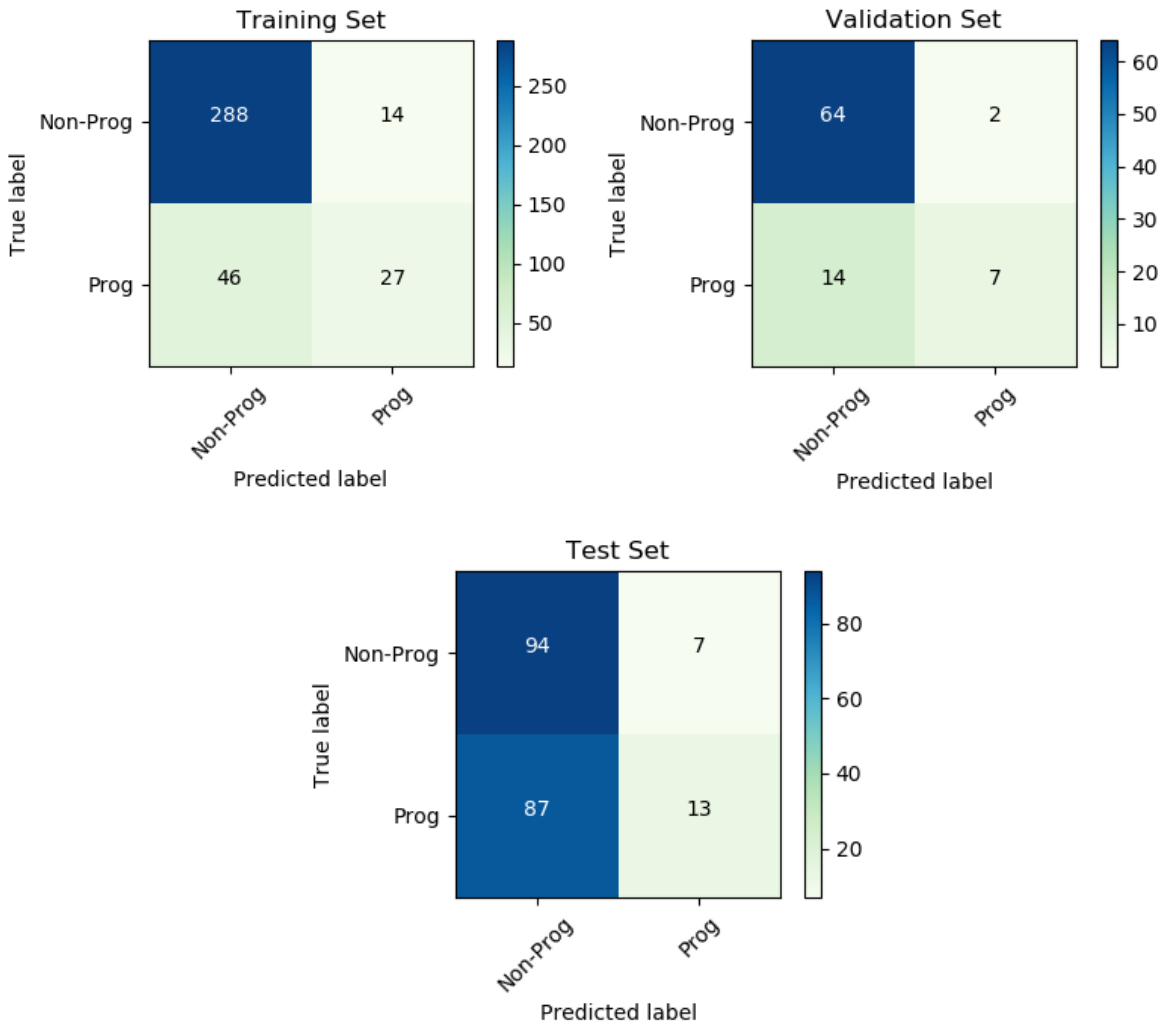
Looking at plots of error rate versus number of neighbors for the validation set in Figure #, we see that the error is the lowest at $K = \{6,9\}$ for the scaled features and $K = \{9\}$ for the unscaled features. To compare the accuracy of scaled and unscaled features, $K = 9$ was chosen for both since low K-values are sensitive to outliers. It is important to note the optimal number of neighbors was chosen based on the fit to the validation set and not according to the fit on the training set. The latter resulted in an optimal K-value of 1, which lead to an overfit of 100% accuracy to the training set.

Set	Accuracy without Scaling, K = 9	Accuracy with Scaling, K = 9
Unbalanced Training	81.1%	84%
Validation	83.9%	81.6%
Testing	51.2%	53.2%

Accuracy results are shown in Table #. Very similar accuracies are achieved in both scenarios. This makes the importance of scaling the data questionable for this application. But overall higher accuracies are achieved with scaling. The confusion matrices in Figure # and Figure # summarize the performance of classifiers created on the unbalanced training, validation, and test sets.



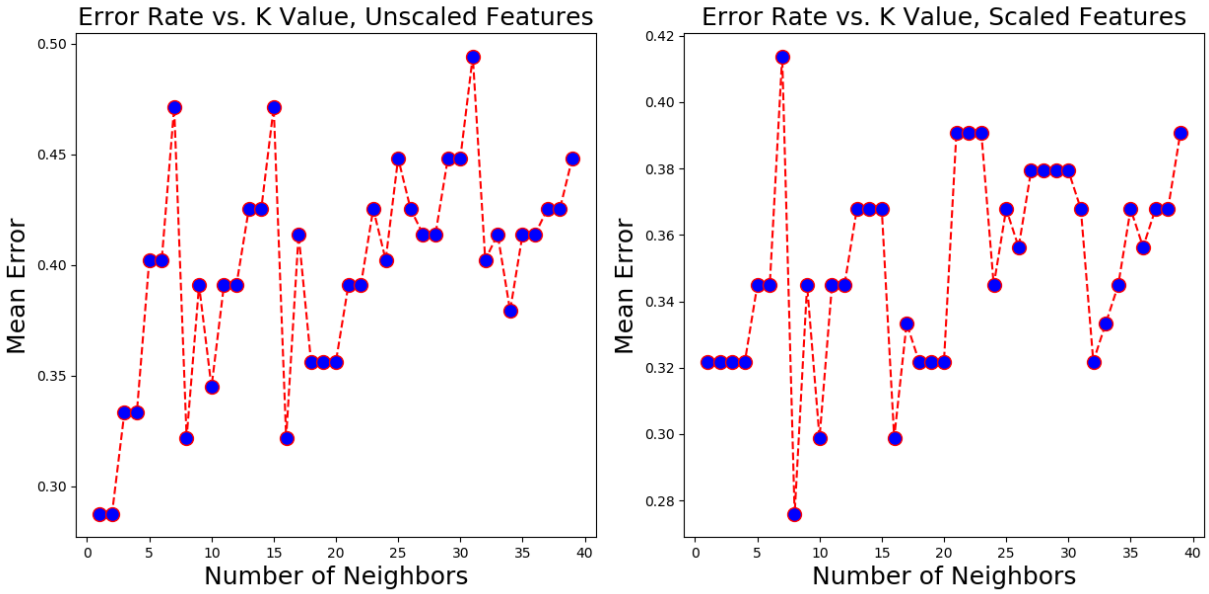
Confusion Matrices of Unbalanced Training Set with Unscaled Features, K = 9



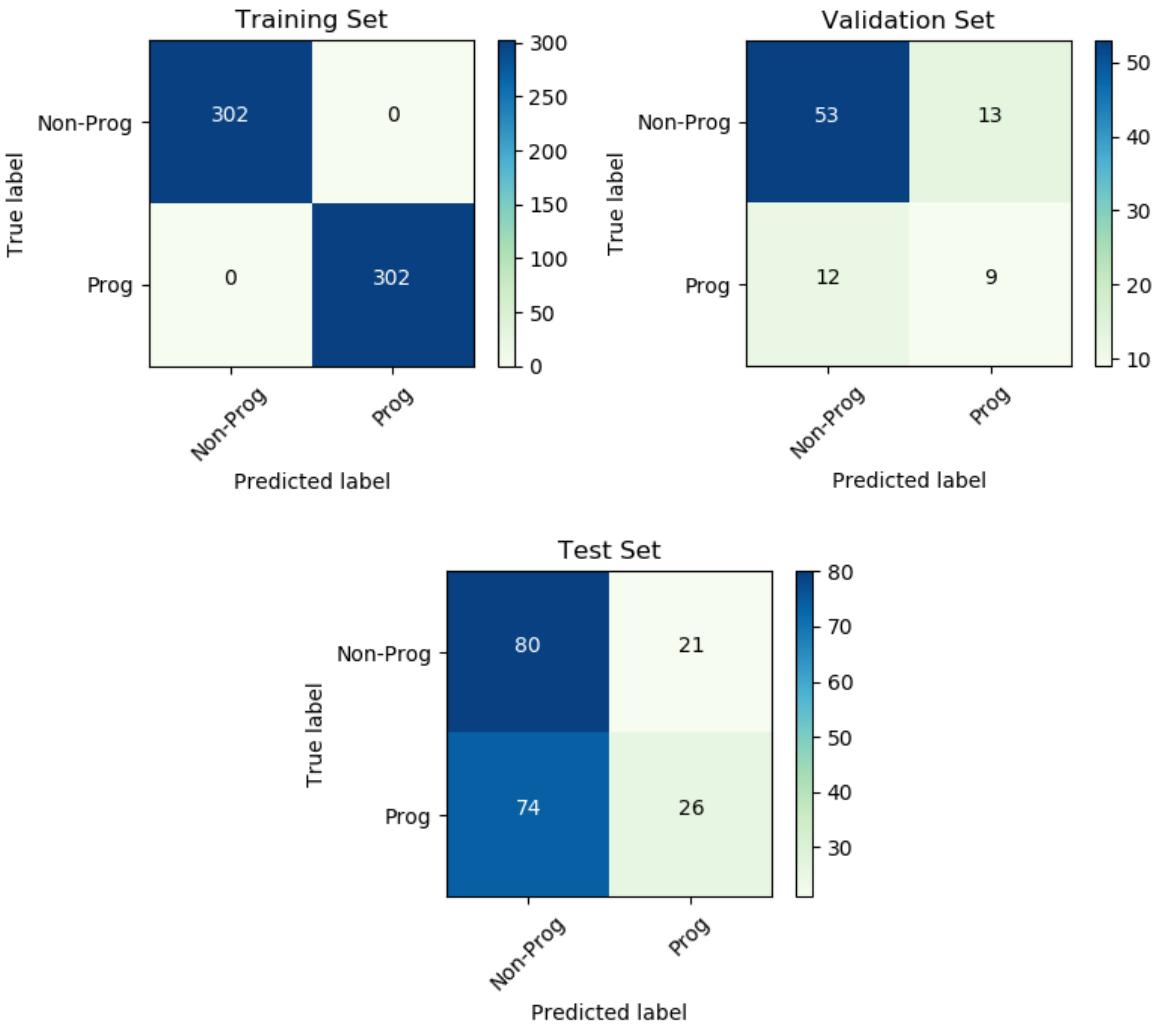
Confusion Matrices of Unbalanced Training Set with Scaled Features, K = 9

A key downfall of KNN is apparent in the above confusion matrices: sensitivity to imbalanced data. The training and validation sets both have more non-prog data than prog data. This results in better classification of the non-prog songs. Thus, the training set was padded in an early observation to assess the implications of the unbalanced training data. Results showed that error had no clear trend versus number of nearest neighbors but, 2 was the optimal K-value for unscaled data and 8 was the optimal K-value for scaled data. This can be seen in Figure #. The accuracy here is summarized in Table #. Clearly, there is an overfit to training data when it is not scaled due to the low K-value. Depending on acceptable accuracy ranges for the validation set, it is possible a higher K-value could be chosen to prevent outlier sensitivity.

Set	**Accuracy without Scaling, K = 2**	**Accuracy with Scaling, K = 8**
Balanced Training	100%	78.6%
Validation	71.2%	72.4%
Testing	52.7%	62.1%



Assessing Optimal Number of Neighbors for Balanced Training Data Based on Validation Set



Confusion Matrices of Balanced Training Set with Unscaled Features

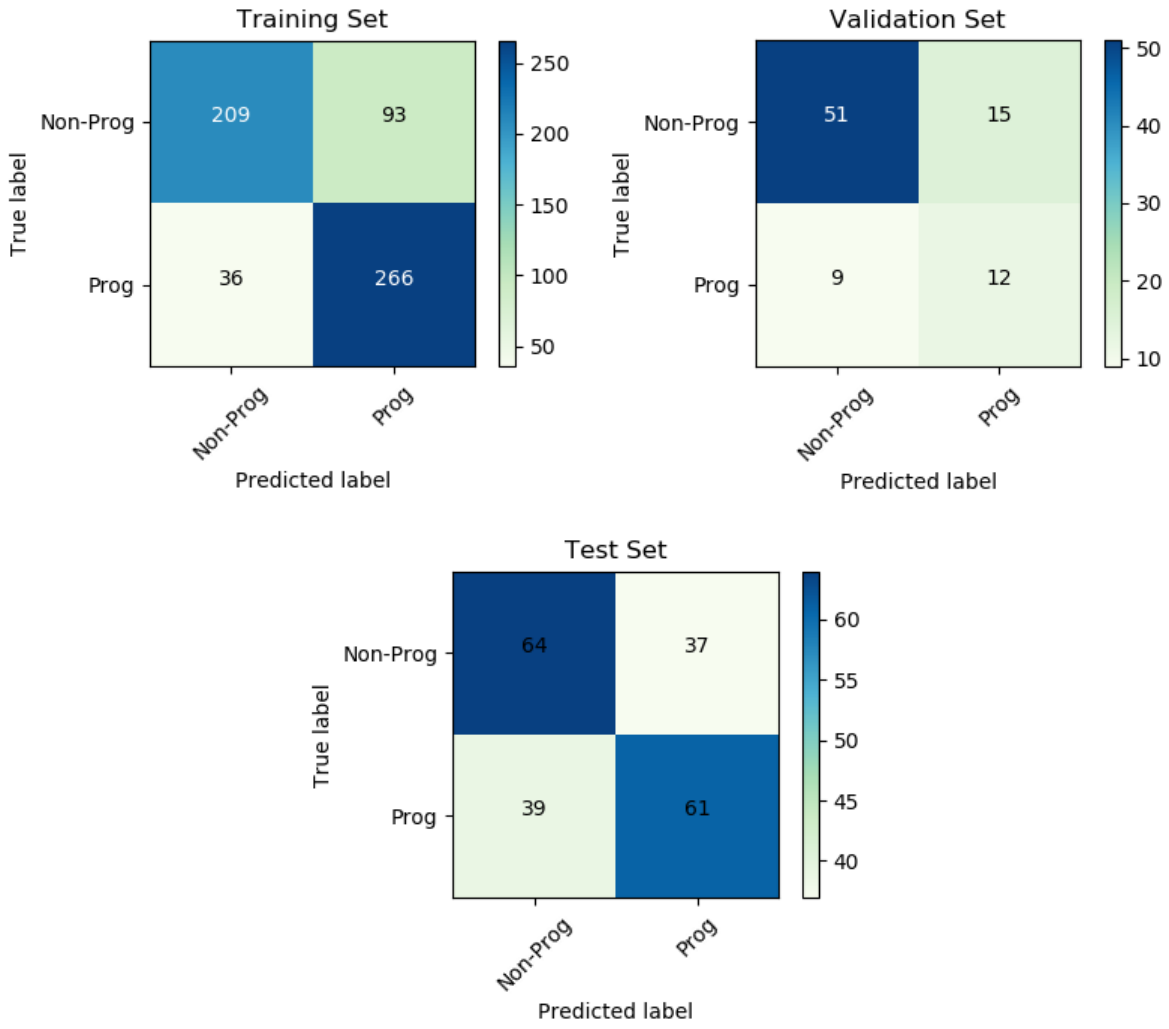


Figure #. Confusion Matrices of Balanced Training Set with Scaled Features

To summarize, artificially balancing the data caused an overfit on the balanced data and barely improved accuracy on the test set. In addition, selecting the number of nearest neighbors based on the validation set did not generally improve test set performance, especially if outliers are present in the test set. In the future for similar applications, a threshold of validation performance should be specified beforehand to prevent overfitting. Of the 4 variations of KNN compared, balanced training data paired with scaled features (for all datasets) provided the best test set performance with **62.1% classification accuracy.**

Since via experiments and above presented proofs, balancing the data set caused no considerable changes in the accuracy or the confusion matrices, we will not be considering the case of data balance vs data imbalance for future models. We just proceed with the data sets that we have.

Working with decision threshold and AUROC:

We observed that there were a lot of false negative in our confusion matrices, so in order to optimize the problem of false negatives, we experimented with the AUC ROC curve and started to experimenting with different thresholds.

The next 3 models that we present have decision threshold and AUROC tuning in them. Generally the decision threshold for classifying an input to belong to the positive class is if and only if the predicted class probability is >=0.5. We try to tweak the decision threshold in order to obtain the best performance. After experimenting a lot, the decision threshold of 0.35 to 0.45 seems to work best by avoiding False Negatives.

Random Forest Classifier

Random forest classifier is an ensemble algorithm. Ensembled algorithms are those which combine more than one algorithms of the same or different kind to perform classification. Random forest classifier creates a set of decision trees from randomly selected subset of training set. It then aggregate the votes from different decision trees to decide the final class of the test object. We chose random forest cause technically it should work well because a single decision tree maybe prone to noise, but aggregate of many decision trees reduce the effect of noise giving more accurate results.

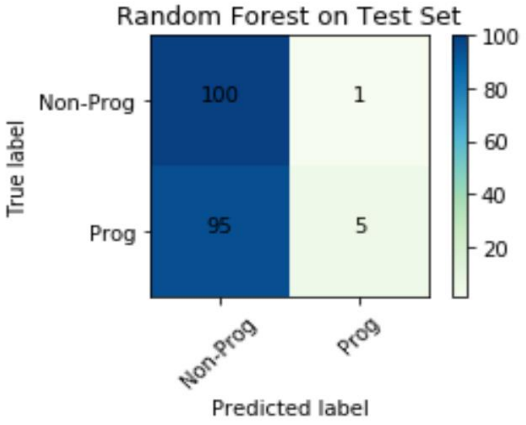
Random forest classifier has a few parameters that we can play around with and optimize them as per our data. They are maximum depth, number of estimators, minimum split etc. The question of balancing the data has also been explored in this model and see how it effects the performance of the model.

Below we are listing the experiments,

1.First with not combining the validation set with the training set for training after hyper parameter tuning.

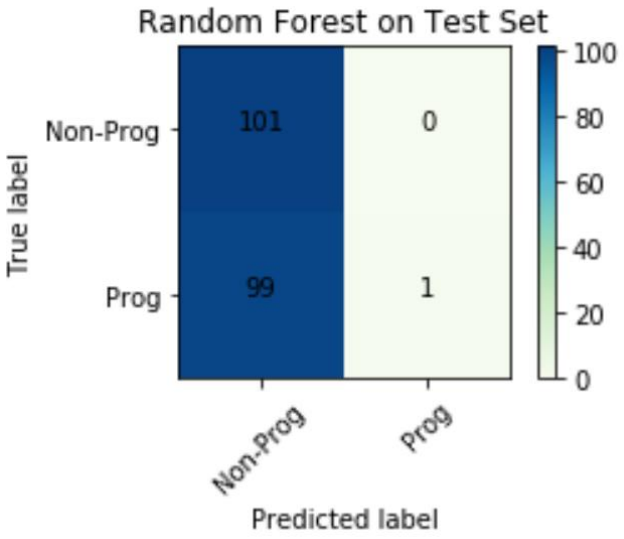
This is the best model of the Random Forest, accuracy metrics and confusion matrices

```
Training set accuracy:  0.9306666666666666
Validation set accuracy : 0.8045977011494253
Validation set confusion matrix:
[[65  1]
 [16  5]]
Test set accuracy : 0.527363184079602
[[100   1]
 [ 94   6]]
```



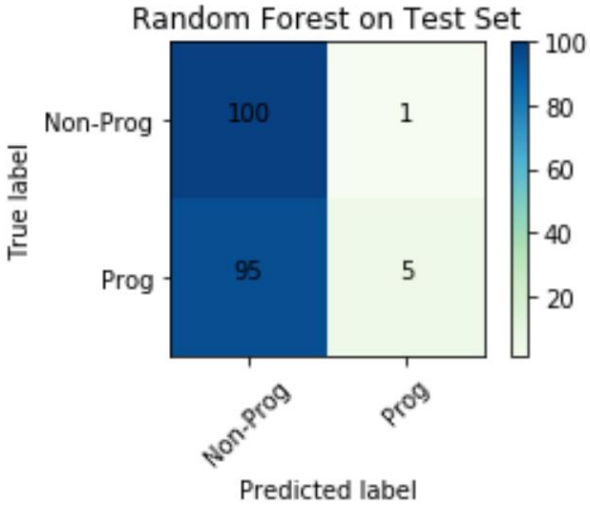
This is the best model with variation of a hyper parameter of **max depth** reduced to 3 from 4

```
Training set accuracy:  0.8373333333333334
Validation set accuracy : 0.7586206896551724
Validation set confusion matrix:
[[65  1]
 [20  1]]
Test set accuracy : 0.5074626865671642
[[101   0]
 [ 99   1]]
```

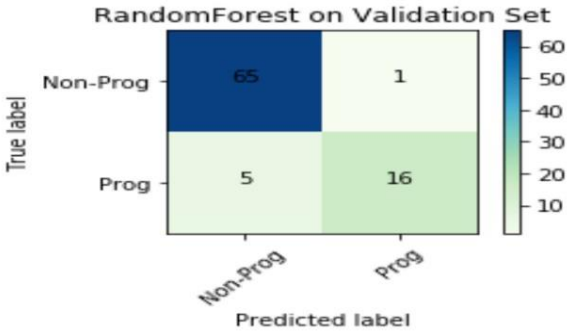


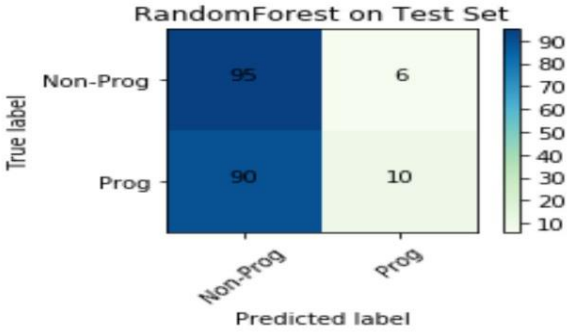
This is the best model with variation of a hyper parameter of **number of estimators** set to 50 from 100

```
Training set accuracy:  0.92
Validation set accuracy : 0.7931034482758621
Validation set confusion matrix:
[[64  2]
 [16  5]]
Test set accuracy : 0.5223880597014925
[[100   1]
 [ 95   5]]
```



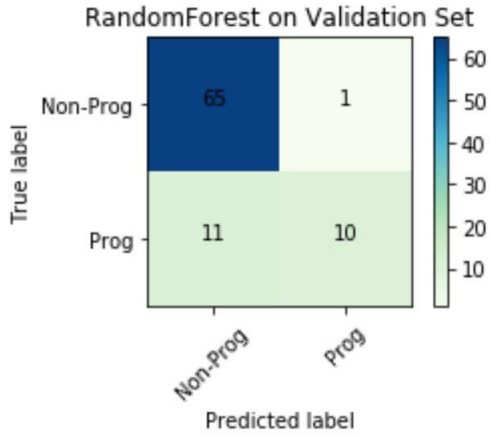
2. Below is the same best model with the validation set now combined with the training set and retraining the model on this new combined set.

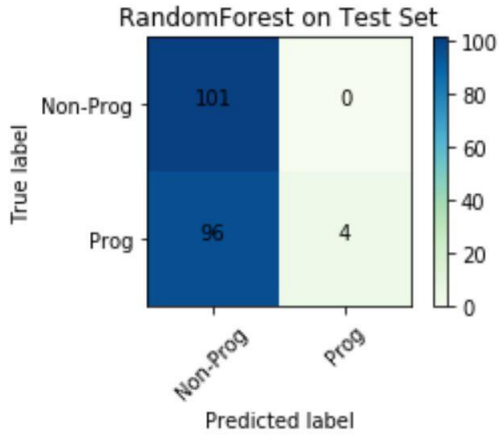





```
Training set accuracy:  0.9069264069264069
Validation set accuracy : 0.9310344827586207
Validation set confusion matrix:
[[65  1]
 [ 5 16]]
Test set accuracy : 0.5223880597014925
[[95  6]
 [90 10]]
```

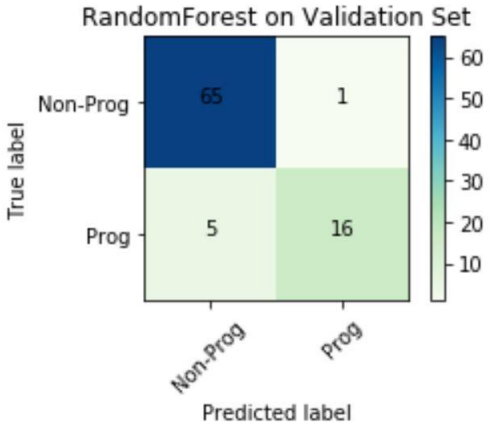
This is the variation of the above best model with hyper parameter of **max depth** set to 3 from 4

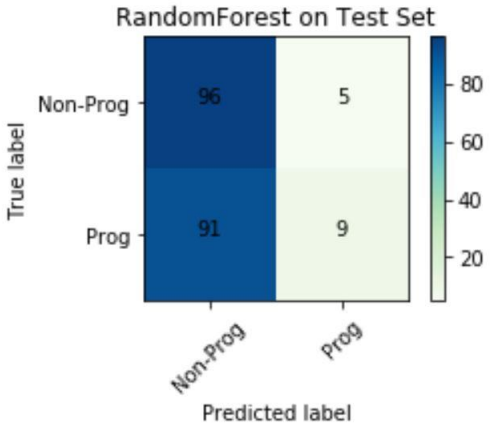





```
Training set accuracy:  0.8506493506493507
Validation set accuracy : 0.8620689655172413
Validation set confusion matrix:
[[65  1]
 [11 10]]
Test set accuracy : 0.5223880597014925
[[101   0]
 [ 96   4]]
```

The variation of the model with hyper parameter of **number of estimators** set to 50 from 100






```
Training set accuracy:  0.9004329004329005
Validation set accuracy : 0.9310344827586207
Validation set confusion matrix:
[[65  1]
 [ 5 16]]
Test set accuracy : 0.5223880597014925
[[96  5]
 [91  9]]
```

As we can see combining the validation set with the training set and then retraining the model did not bring about much changes in the accuracy or the confusion matrices. Random forests are less intuitive than K-Nearest Neighbors. Because of the large number of decision trees, it is hard to have an intuitive grasp of the relationship existing in the input data. As we can see that the **maximum accuracy** achieved via **Random Forest Classifiers was 53%**

Gradient Boosting Classifier:

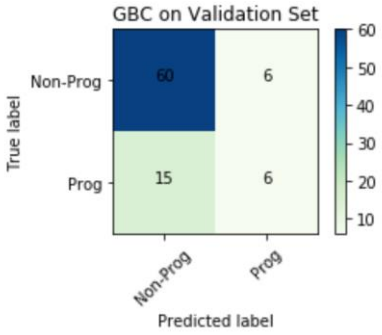
Boosting is a method of converting weak learners into strong learners. In boosting, each new tree is a fit on a modified version of the original data set. The gradient boosting algorithm (gbm) can be most easily explained by first introducing the AdaBoost Algorithm. The AdaBoost Algorithm begins by training a decision tree in which each observation is assigned an equal weight. After evaluating the first tree, we increase the weights of those observations that are difficult to classify and lower the weights for those that are easy to classify. The second tree is therefore grown on this weighted data. Here, the idea is to improve upon the predictions of the first tree. Our new model is therefore *Tree 1 + Tree 2*. We then compute the classification error from this new 2-tree ensemble model and grow a third tree to predict the revised residuals. We repeat this process for a specified number of iterations. Subsequent trees help us to classify observations that are not well classified by the previous trees

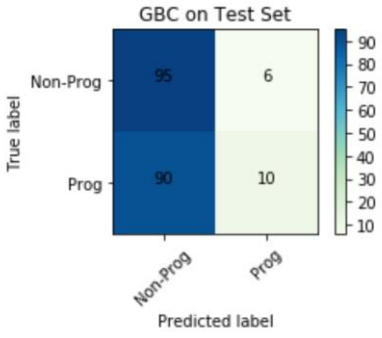
Gradient Boosting trains many models in a gradual, additive and sequential manner. The major difference between AdaBoost and Gradient Boosting Algorithm is how the two algorithms identify the shortcomings of weak learners (eg. decision trees). While the AdaBoost model identifies the shortcomings by using high weight data points, gradient boosting performs the same by using gradients in the loss function (*y=ax+b+e , e needs a special mention as it is the error term)*.

We performed various experiments on the GB Classifier, we play around with the hyper parameters of number of estimators and maximum depth and present the results below.

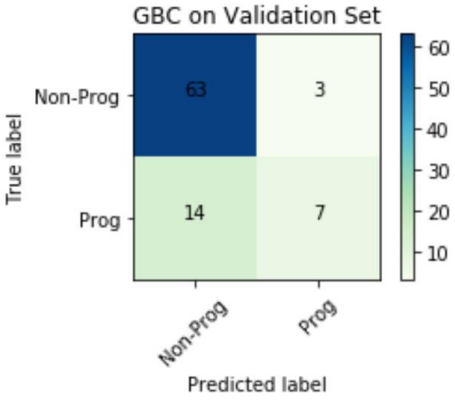
1. **The model is just trained on the training set only, below are the results.**

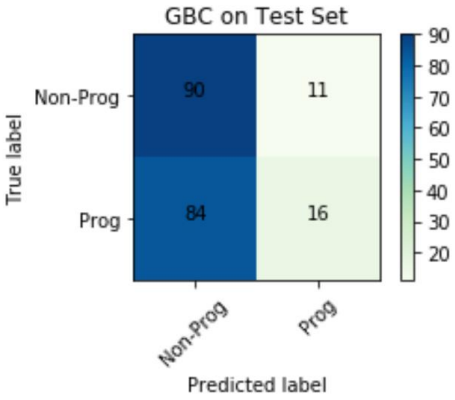
```
train accuracy:  0.976
 Val accuracy : 0.7586206896551724
[[60  6]
 [15  6]]
Test set accuracy : 0.5223880597014925
[[95  6]
 [90 10]]
```





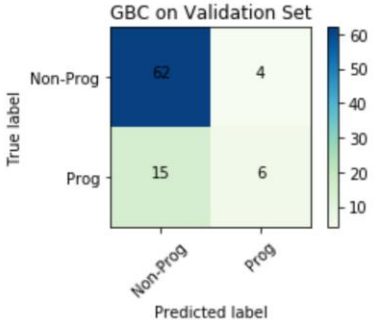
The model is now tuned with hyper parameter **number of estimators set to 100 from 40**

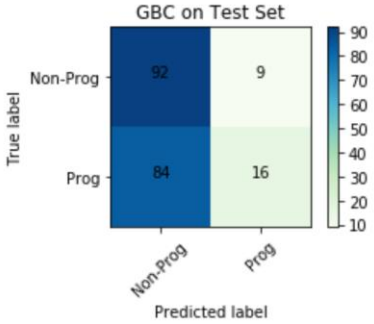





```
train accuracy:  1.0
 Val accuracy : 0.8045977011494253
[[63  3]
 [14  7]]
Test set accuracy : 0.527363184079602
[[90 11]
 [84 16]]
```

The model is now tuned with hyper parameter **max depth set to 4 from 3**



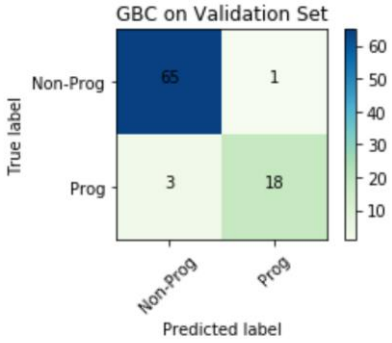


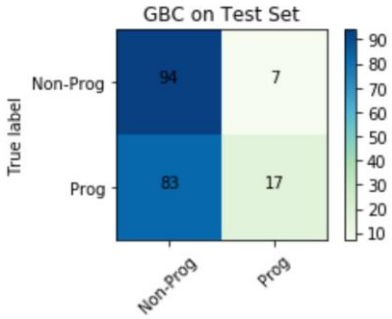

```
train accuracy:  1.0
 Val accuracy : 0.7816091954022989
[[62  4]
 [15  6]]
Test set accuracy : 0.5373134328358209
[[92  9]
 [84 16]]
```

We see that the model is over fit on the training set and hence cannot be trusted on other sets.

2.The model is now trained on a combined set of training and validation sets after hyper parameter tuning, once the model is trained on a training set before.

The results are shown below

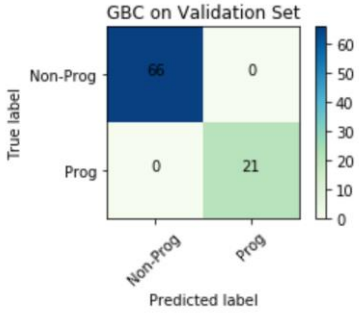


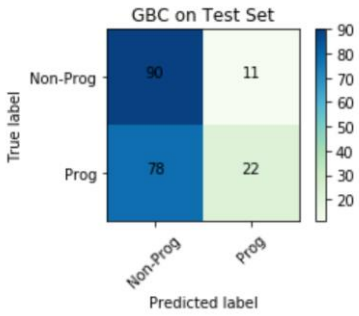



```
train accuracy:  0.9632034632034632
 Val accuracy : 0.9540229885057471
[[65  1]
 [ 3 18]]
Test set accuracy : 0.5522388059701493
[[94  7]
 [83 17]]
```

We see that the model actually increases in accuracy by around 3%

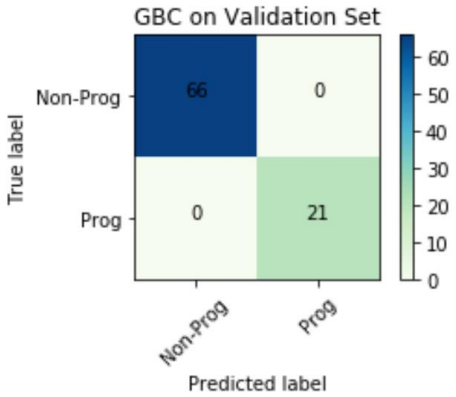
The model is now tuned with hyper parameter **number of estimators set to 100 from 40**

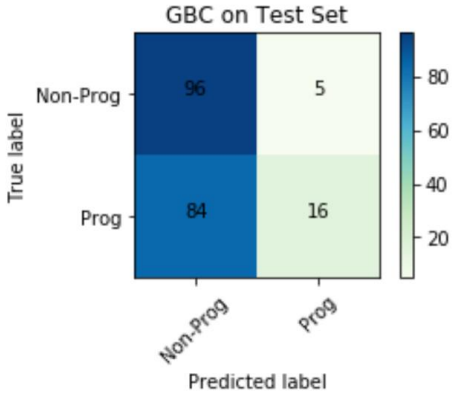





```
train accuracy:  1.0
 Val accuracy : 1.0
[[66  0]
 [ 0 21]]
Test set accuracy : 0.5572139303482587
[[90 11]
 [78 22]]
```

The model is now tuned with hyper parameter **max depth set to 4 from 3**






```
train accuracy:  1.0
 Val accuracy : 1.0
[[66  0]
 [ 0 21]]
Test set accuracy : 0.5572139303482587
[[96  5]
 [84 16]]
```

We can again see that the model has overfit on the training and validation sets and hence can not be trusted as an efficient model when it comes to fairness in test sets.

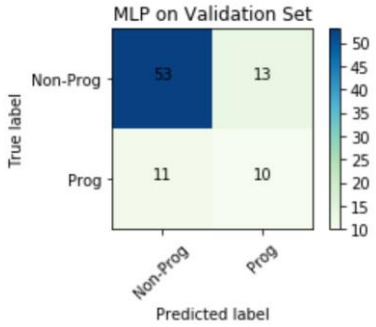
Multilayer Perceptron:

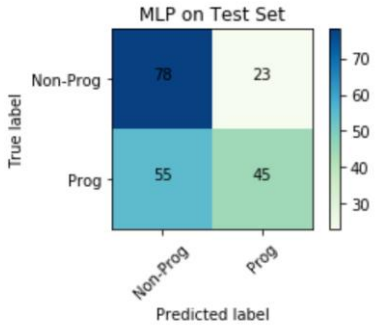
A multilayer perceptron (MLP) is a class of feedforward artificial neural network. A MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function.

We chose an MLP to tackle the problem of classifying prog vs non prog because Multilayer perceptrons are intuitive models in the sense that they are often applied to supervised learning problems, they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs. Training involves adjusting the parameters, or the weights and biases, of the model in order to minimize error. Backpropagation is used to make those weigh and bias adjustments relative to the error, and the error itself can be measured in a variety of ways, including by root mean squared error.

We performed various experiments with the MLP as well. We first play around with the hyper parameters of hidden layer sizes.

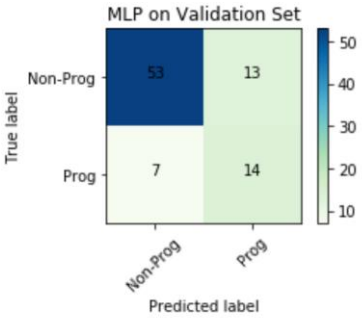
1.The model is trained on just the training set, without merging with the validation set and is experimented upon. Below are the results.

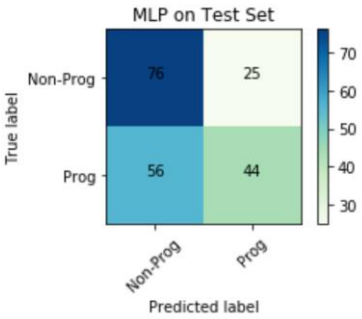





```
train accuracy:  0.8933333333333333
 Val accuracy : 0.7241379310344828
[[53 13]
 [11 10]]
Test set accuracy : 0.6119402985074627
[[78 23]
 [55 45]]
```

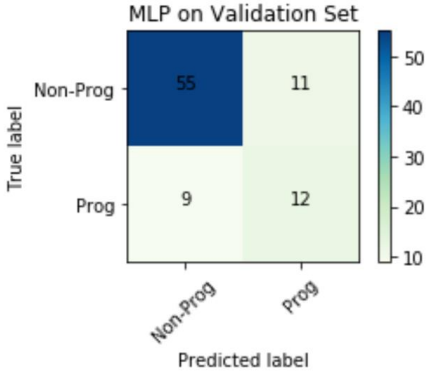
The model with hyper parameter **hidden layer having 3 layers reduced from 4:**

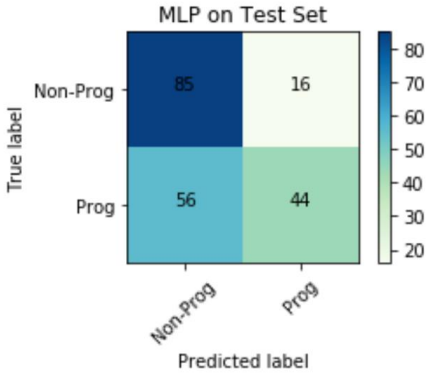





```
train accuracy:  0.8426666666666667
 Val accuracy : 0.7701149425287356
[[53 13]
 [ 7 14]]
Test set accuracy : 0.5970149253731343
[[76 25]
 [56 44]]
```

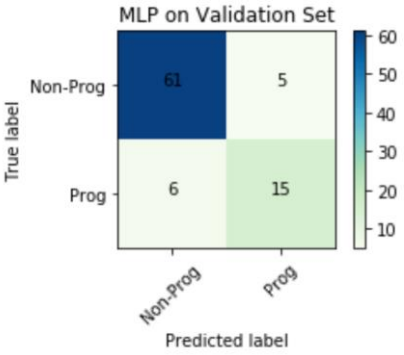
The model with hyper parameter **alpha set to $1e^{-3}$ from $1e^{-2}$:**

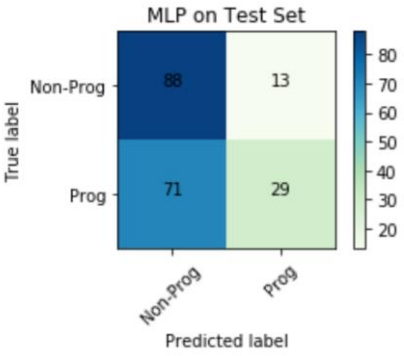





```
train accuracy:  0.904
 Val accuracy : 0.7701149425287356
[[55 11]
 [ 9 12]]
Test set accuracy : 0.6417910447761194
[[85 16]
 [56 44]]
```

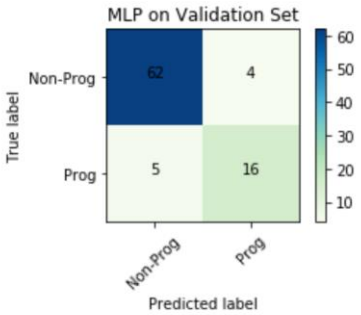
2. Now we train the MLP model on the validation set combined with the training set and then test on the test set.

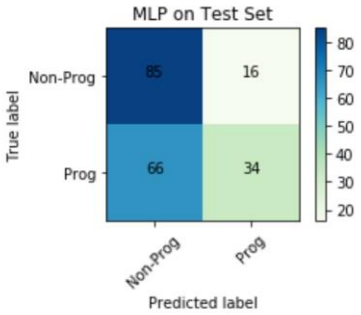





```
train accuracy:  0.829004329004329
 Val accuracy : 0.8735632183908046
[[61  5]
 [ 6 15]]
Test set accuracy : 0.582089552238806
[[88 13]
 [71 29]]
```

The model with hyper parameter **alpha set to $1e^{-3}$ from $1e^{-2}$**

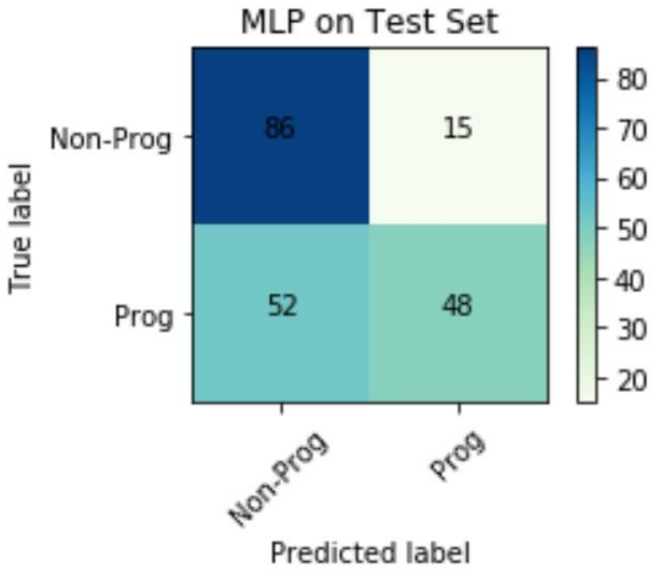





```
train accuracy:  0.8463203463203464
 Val accuracy : 0.896551724137931
[[62  4]
 [ 5 16]]
Test set accuracy : 0.5920398009950248
[[85 16]
 [66 34]]
```

Now we present our best performing model yet on the test set. It is an MLP model with 4 hidden layers and alpha set to $1e^{-2}$, a high normalization rate alpha ensures that the validation and training sets don't overfit, thereby ensuring good model performance on the test set. This model was just trained on the training set, without combining the training and validation set.

```
Test set accuracy : 0.6666666666666666
[[86 15]
 [52 48]]
```



This is the best performing model with a test set accuracy of 67%

We observe that again in the MLP classifier, combining the training set and validation set after hyper parameter tuning on the validation set yielded no considerable change in accuracy. Hence we conclude with our best performing model as an MLP model providing 67% accuracy on the test set.

The best performing model of each classifier is stored in the pickle files. The best performing model overall is the MLP classifier which is stored in the "best_model_mlp.pkl" file. Running this model on the test set with the features we have generated gave us the highest accuracy of 67%.

The Jupyter Notebook has all models except the K-Nearest Neighbhors, the K-Nearest Neighbors model is provided as a separate python file in the zip file that we have submitted.

We have also tried other models including an SVC and recurrent neural networks, but the test set accuracy was below 50% which is worse in performance when considered with us randomly guessing and still having a 50% chance to get it right. So we haven't reported it in this project report

Part -2

Discussions on the issue of Logical Supervenience, Local and Global Supervenience and the relevance of Logical Supervenience on present day AI

The notion of supervenience is a philosophy relation that describes set of properties. David Chalmers defines a set of properties *A* to supervene on another set *B* if no two things can differ with respect to *A*-properties without also differing with respect to their *B*-properties. In his Stanford Encyclopedia of Philosophy article on the topic, Chalmers stated to a degree that the mental is supervenient on the physical. We follow with necessary definitions to assist in defining our first viewpoint on this matter.

Properties a1 and a2 $\in$ *A* are considered **base-indiscernible** if they are exactly alike with respect to *B*-properties. Further, a1 and a2 can be considered **supervenient-indiscernible** to b1 and b2 $\in$ B because their supervenience hierarchies are identical. For the discussion of local and global supervenience, **world** is defined as the whole of the physical universe, including the earth and all the people and things on it. In this philosophical context, world can also refer to the ontological world of individual human beings. But the latter definition is not used in this context; I will stick to the former definition to fit well with my examples that follow. Also, **consciousness** is interpreted as mental awareness and perception.

The thesis for **local supervenience** is as follows: For any worlds w1 and w2, and any individuals x $\in$ w1 and y $\in$ w2, if x and y are base-indiscernible then they are supervenient-

indiscernible. There are strong and weak varieties of local supervenience; the difference being whether correlations between base and supervenient properties hold *in the actual worlds only* or *across possible worlds.* Below are definitions:

Weak: For any world w, and two objects within that world x1 and x2, if x1 and x2 are base-indiscernible, then they are supervenient-indiscernible. For example, if my thoughts supervene on my actions locally then the same should be true for any other person on my world. We are therefore both physically and psychologically the same on Earth since we are base-indiscernible.

Strong: For any world w1 and w2, and for any 2 objects x in w1 and y in w2, if x and y are base indiscernible then they are supervenient indiscernible. For example, consider The Flash Series shown on Netflix. The important plot detail is that there are parallel universes with doppelgangers for each series character on each Earth within each universe. The Flash on Earth-1 (E-1) has a physical duplicate on Earth-2 (E-2) to Earth-K (E-K), with K as positive integer. These doppelgangers share physical attributes with E-1 Flash. Thus, they likely have similar thoughts to E-1 Flash because their underlying personality traits, stemming from psychological underpinnings, are similar. And in the series, we see that most, if not all, doppelgangers share the same ego. Moreover, the Flashes are base-indiscernible and therefore are also supervenient-indiscernible.

To contrast, if The Flash and his band of doppel brothers psychologically weakly local supervene on the physical, then only when E-17 Flash travels to Earth 1 will both Flashes start thinking alike and making similar decisions. Otherwise, when they are on their respective Earths their psyche are not alike. This is because the correlations between base and supervenience properties are maintained in each world but are different across multiple worlds.

Global supervenience says the following: For any possible worlds, w1 and w2, if w1 and w2 are base-indiscernible then they are supervenient indiscernible. A clear example of this is E-1 and E-X on The Flash series. E-1 is where we live and exist now. E-X is a parallel Earth to E-1 with a similar history to us but this earth was created after ours so their timeline is not where we are now. There is an episode of the Flash where the characters on E-1 travel to E-X to help prevent a holocaust. If the tide of human psychological development on E-X was sinister enough to

transition to a holocaust, then I think it's safe to assume the base properties of human psychology were similar enough to E-1 that they are traversing an almost identical course of history to us.

This leads to a clear position on the topic of interest. From a local and global perspective, consciousness is indeed logically supervenient on the physical. Donald Davidson, a philosopher who played a key role in defining supervenience wonderfully summed the perspective that will be argued here: "...there cannot be two events alike in all physical respects but differing in some mental respect, or that an object cannot alter in some mental respect without altering in some physical respect". Locally, be it weak or strong, some difference in consciousness must occur for any change in the physical to be possible. I could change my mind and perceive that my hair was purple. But it is not until I actually dye my hair that my hair will change color. A strong local example of this is the comparison of E-1 Flash and E-2 Flash. E-2 Flash has an "ideal" life in terms of the dreams and desires of E-1 Flash. But, they both perceivably had similar childhoods and identical families. However, their decisions (conscious mental choices) varied throughout time and their lives turned out different. Globally, this still holds.

From the counter argument by Tae-Ryang Kim to Chalmers, apparently Chalmers position is that the distinction between global and local supervenience does not matter too much when it comes to conscious experience. It seems that way from my assessment also. If there is no tangible change in the physical, then there is no change at all. Your thoughts and consciousness cannot truly vary and make an impact unless something physically changes.

To continue the discussion about supervenience of consciousness, we need to distinguish natural supervenience and logical supervenience. Natural supervenience is based on two properties having a supervenience relation with respect to the natural laws. As the example given in David Chalmers' *The Conscious Mind* that

> "the pressure exerted by one mole of a gas systematically depends on its temperature and volume according to the law pV = KT, where K is a constant (I pretend for the purposes of illustration that all gases are ideal gases). In the actual world, whenever there is a mole of gas at a given temperature and volume, its pressure will be determined, (Chalmers, 36)"

we see that the supervenience, or dependence, is bounded by this formula of a fact of this particular natural world. This correlation between all physics measurements simply reflect their relations in nature. Moreover, this natural supervenience relation is much weaker than the logical supervenience relation. To understand the difference between natural supervenience and logical

supervenience, we are going to define logical supervenience next. David Chalmers explains the concept of logical supervenience as the following: “B-properties supervene logically on A-properties if no two logically possible situations are identical with respect to their A-properties but distinct with respect to their B-properties. (Chalmers, 35)” In short, if B-properties are logically supervenient on A-properties, there cannot be changes made with respect to B-properties without changes in A-properties. One problem of logical supervenience occurs when two identical entities share the same A-properties with one that has additional B-properties. As Chalmers later redefines logical supervenience, “B-properties are logically supervenient on A-properties if the B-properties in our world are logically determined by the A-properties in the following sense: in any possible world with the same A-facts, the same B-facts will hold. (Chalmers, 39)” According to Chalmers, as long as we emphasize that the difference of A-properties causes the difference in B-properties, we can avoid the complications of those additional B-properties. In later discussion, we will use this refined definition for logical supervenience.

Before we understand the core issue of logical supervenience of consciousness on the physical, we need to fully understand consciousness as there many aspects of consciousness and there is not a single definition that can precisely describe consciousness. Among many arguments around this topic, Chalmers’ opinion on logical supervenience of consciousness on the physical is the following: “I will argue that conscious experience does not supervene logically on the physical, and therefore cannot be reductively explained. (Chalmers, 71).” Chalmers’ argument in support of the above opinion is that we do not fully understand all aspects of consciousness, hence we cannot possibly say that consciousness is logically supervenient on the physical. We argue below that there are certain aspects of consciousness that are indeed logically supervenient on the physical, yet we are still unsure whether other aspects of consciousness are logically supervenient on the physical. Understanding this core issue of logical supervenience of consciousness on the physical may inspire us to improve on the logical supervenience concept of modern day artificial intelligence.

First we show some typical aspects of consciousness as well as what we do not understand about consciousness. To answer the question whether consciousness is logically supervenient to the physical, we first need to understand what consciousness is. Consciousness

can be interpreted as creature consciousness, state consciousness, and consciousness as an entity. While some like to think that consciousness is carried out by an object being in its conscious state, when referring to consciousness we often include a subcategory of consciousness, self-consciousness. If we consider consciousness of an object as the object being capable of receiving external information and internalizing such information (with or without taking future actions in response to the original information), then self-consciousness requires more from the object as to obtaining insight beyond the physicals of its own.

Now we consider the part of consciousness that is logically supervenient to the physical. Suppose we have a pair of identical twins and suppose ideally they are exactly the same in every aspect, that is, they have the exact same physical features and they think alike. Suppose one of the twins gets into an accident and damages part of his/her occipital lobe. As a consequence, this twin's visual processing ability is severely compromised and he/she depends on others to know where he/she is most of the time. This is an example of the fact that consciousness is logically supervenient to the physical. Moreover, since memories shape us to be who we are at the moment, the physical features including the complex and unique combinations of memories have made us all different. If this is the case, how can consciousness not be logically supervenient to the physical?

Consider a pair of identical twins again with the exact same physical features, think alike, and each of them has a soul. The soul, in many religious, philosophical, mythological traditions, exists as the incorporeal essence of a living being. Now the souls of the twins will have the power to change their minds, yet souls do not depend on tweaking the twins' physicals to achieve this change; in fact, since the souls are incorporeal, it is impossible to expect a change in the physicals of the twins before the change in their minds occur. This is similar to what Chalmers mentions in *The Conscious Mind*, "Some high-level properties fail to supervene logically because of a dependence on conscious experience. Perhaps conscious experience is partly constitutive of a property like love, for example. (Chalmers, 72)" What if the change is directly made in one's mind? Like soul and love, things that we as humans are still unable to claim full understandings of, are considered high-level properties by Chalmer. These are also things that seem to only exist without a vehicle of dependence. Yet we cannot claim the opposite to be true either, since these things seem to make change in the mind first before they change the

physicals. In conclusion, it is safest to say that there are certain aspects of consciousness that are logically supervenient on the physical while there are other aspects of consciousness that we are unable to claim their logical supervenience, since we lack understanding of these aspects of consciousness.

In application, we consider this issue of logical supervenience in terms of modern day artificial intelligence (AI). While we have designed AI to have certain physical features just like ours so that they can carry out daily tasks as well as us, the AI today seems to be self-conscious but only on the physical level. We have a fairly clear picture of the connection between our physicals and the brains, so the AI are designed the same way to perform tasks as how we use our bones and muscles. As there are many situations that require AI teamwork, being aware of the surroundings and taking corresponding actions to avoid conflict or collision becomes vital. Therefore, AI is programmed to recognize such conflicts and take actions to avoid them. However, there are no such AI right now that can tell you whether it thinks itself as a moral being, whether it is capable of expressing emotion, or even more, whether it can control its own drives. In other words, even though modern day AI has self-awareness on the physical level, it does not have internal representation of itself. Let us take a modern day reinforcement learning (RL) AI as an example with a human being. Suppose the AI is powerful enough to follow its reinforcement learning models in every aspect of the brain. The human goes on with his/her day and his/her idea of morality might change a little bit over the course of the day because of the learning that has happened inside the brain. The brain initiates the thinking and learning and then changes the behavioral model based on the new learning. Now for comparison, we take a look at the RL AI that has a fixed RL model and this model might get a few updates during the day then improve its performance based on the new model. The difference between the two should be obvious: the AI lacks the thinking part of the work, whereas the brain does not need to be fed with the learning model. The self-consciousness behind the motivation that the brain thinks and learns is what lacks in modern day AI. However, this is not to say that AI cannot get to the point where it functions like a brain one day. This could simply be a matter of time as we continue to study and understand our brains better.

Bibliography

1. Chalmers, David. *The Conscious Mind: in Search of a Fundamental Theory*. Oxford University Press. Oxford, 1996.
2. Tae-Ryang, Kim. *How Does Consciousness Merely Naturally Supervene on the Physical?*. 철학사상. null. 343-375. 10.15750/chss..43.201202.011 (2012).
3. Stanford Encyclopedia of Philosophy, https://plato.stanford.edu/.