LIFELINE HEALTH SERVICES

PREDICT CATEGORIES & SUB CATEGORIES OF INBOUND CALLS

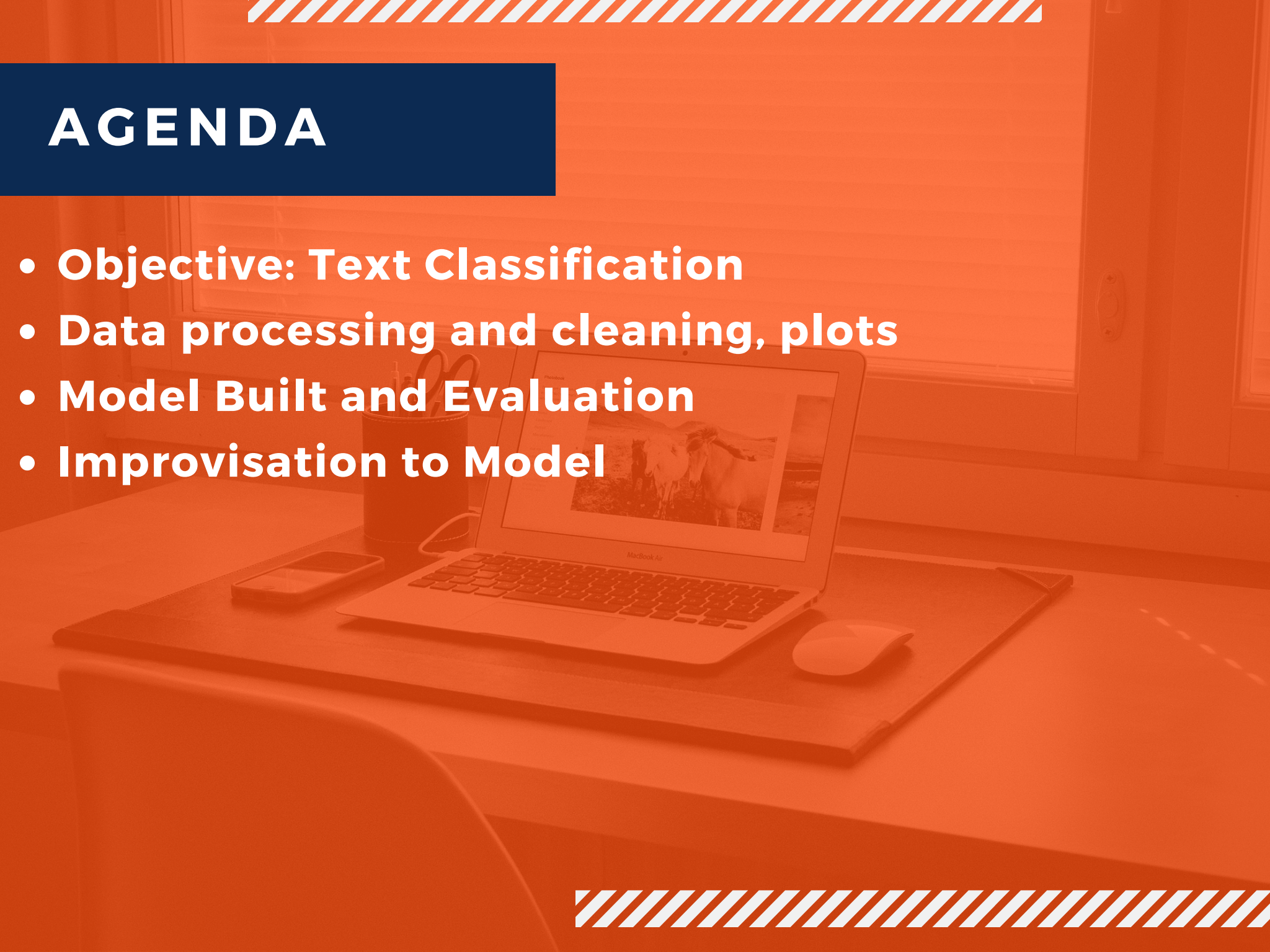
SEPTEMBER 2017

JAYANTH RASAMSETTI

LIFELINE HEALTHCARE SERVICES, LLC

AGENDA

- **Objective: Text Classification**
- **Data processing and cleaning, plots**
- **Model Built and Evaluation**
- **Improvisation to Model**



1 OBJECTIVE

Objectives of current project is:

1) Predicting "categories" (6 levels) and "sub-categories"(21 levels) of inbound calls

2) Classification Problem

3) The given data set has following attributes (57800 rows and 3 columns)

2 DATA PREPROCESSING

1) There are 0 missing values

2) Given data was heavily processed. We conducted tokenization (removed stop words, re.sub("[^a-zA-Z], white spaces, etc)

3) Used Vectorization for unigrams and restricted to 5000 features

4) Wrote a custom "Recall" function in Keras backend (as only accuracy is available). This is not available as keras runs this batch wise so they deprecated it.

In [24]:
```
# Creating a custome "Recall" error function in Keras backend

import keras.backend as K

def recall(y_true, y_pred):
    TP = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    PP = K.sum(K.round(K.clip(y_true, 0, 1)))
    recall = TP / (PP + K.epsilon())
    return recall
```

3 MODEL STEPS

1) Split into train (64%), val (16%) and test (20%)

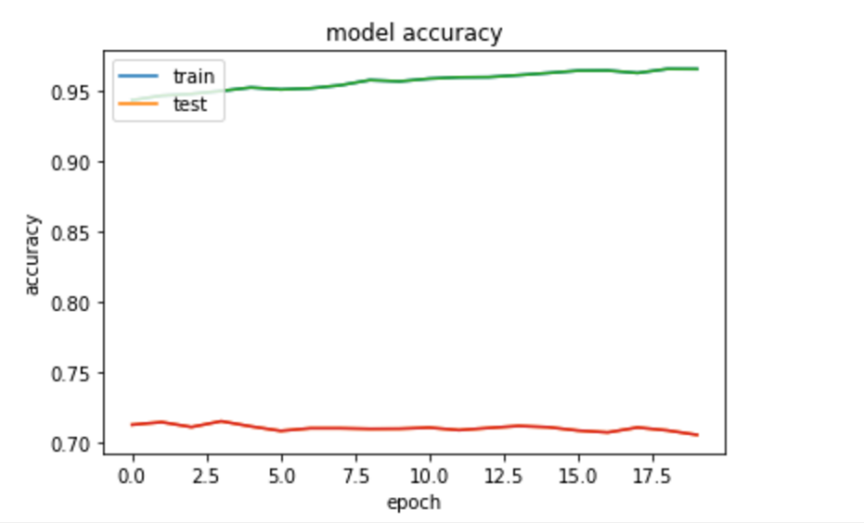
2) Converted the target variables into one_hot_encoding (categorical variables)

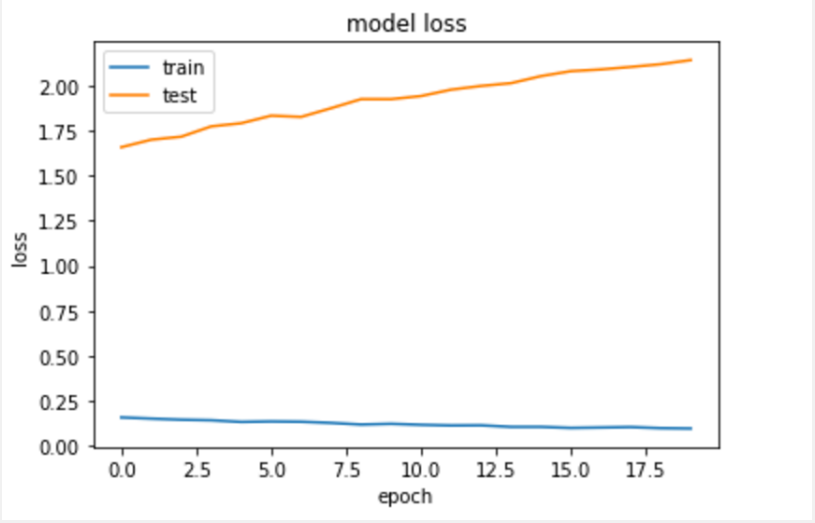
3) Ensured that all were numpy arrays (for feeding into the network)

```
# Split into 64% train, 16% val and 20% test

#First split
train1,test = train_test_split(data_features, test_size = 0.2)
cols = [col for col in data_features.columns if col not in ["categories", "sub_categories"]]

test.x = test[cols]
test.y = test["categories"]
test.z = test["sub_categories"]

#Second split
train,val = train_test_split(data_features[0:45823], test_size=0.2) # 80% of 57280 is 45823
train.x = train[cols]
train.y = train["categories"]
train.z = train["sub_categories"]

val.x = val[cols]
val.y = val["categories"]
val.z = val["sub_categories"]
```

```
8]: # Verify the shape of train.x, train.y
print(train.x.shape, train.y.shape, train.z.shape)
print(val.x.shape, val.y.shape, val.z.shape)
print(test.x.shape, test.y.shape, test.z.shape)
# print(health_data.converse.map(len).max())

((36658, 5000), (36658,), (36658,))
((9165, 5000), (9165,), (9165,))
((11456, 5000), (11456,), (11456,))
```

4 PLOTS


model accuracy
train
test
accuracy
epoch
0.70
0.75
0.80
0.85
0.90
0.95
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


model loss
train
test
loss
epoch
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

5 MODEL BUILT AND EVALUATION - CATEGORIES

Objective is to predict accuracy and recall for each category

Some models worked well with categories

S.No	Text Classification (categories)	Train (Acc, Recall)	Val (Acc, Recall)	Test (Acc, Recall)
1	Naive Bayes	0.84, 0.82	0.84, 0.82	0.84, 0.82
2	MLP Dropout = 0.2 (30 epochs)	0.9862, 9852	0.7815, 0.7788	0.9190, 0.9178
3	MLP Dropout = 0.5, SGD lr = 0.04, decay = 1e-6, momentum = 0.6	0.92, 0.91	0.70, 0.69	0.81, 0.80
4	CNN 1D (10 epochs)	0.4538, 0.4686	0.4131, 0.4112	0.4113, 0.4091
5	LSTM (epoch =10)	0.5714, 0.1985	0.5770, 0.2099	0.5793, 0.2142

6 MODEL BUILT AND EVALUATION- SUB-CATEGORIES

Objective is to predict accuracy and recall for each sub-category

Some models worked well with sub_categories

S.No	Text Classification (sub_categories)	Train (Acc, Recall)	Val (Acc, Recall)	Test (Acc, Recall)
1	MLP Dropout = 0.5 (20 epochs)	0.94, 0.94	0.72, 0.70	0.88,0.88
2	MLP Dropout = 0.5, SGD lr = 0.04, decay = 1e-6, momentum = 0.6	0.90, 0.87	0.72, 0.69	0.87, 0.79
3	CNN 1D (5 epochs)	0.95, 0.04	0.95, 0.12	0.95, 0.16
4	LSTM (epoch =1)	0.91, 0.04	-	-

7 IMPROVISATIONS TO MODELS WITH TIME

1) Explore a bigram and then club categories & sub_categories and then jointly predict

2) There wasn't a heavy class imbalance but

3) An LSTM with word2vec

FIN CORP BANK

PREDICTING CUSTOMER CHURN (SATISFACTION)

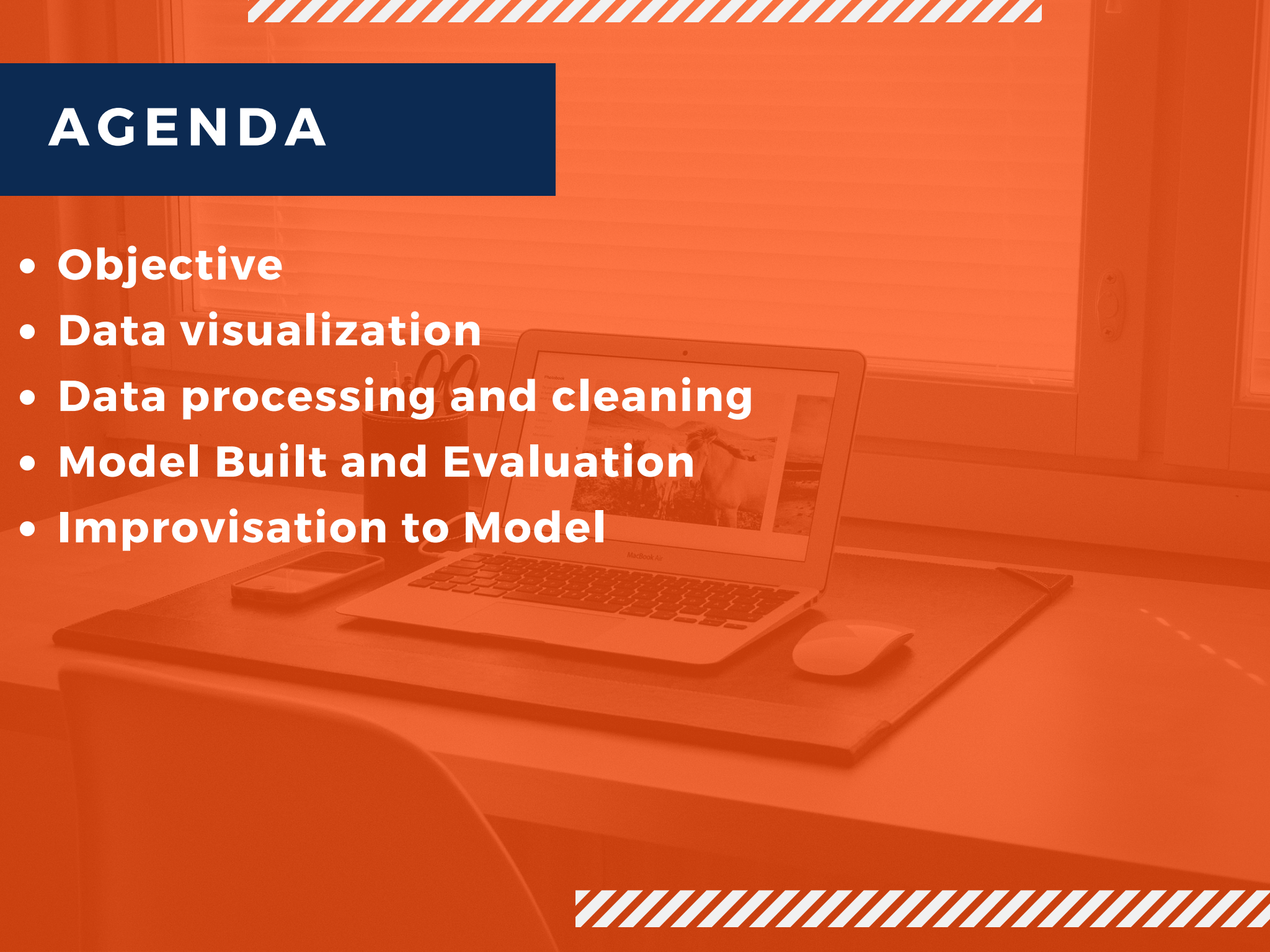
SEPTEMBER 2017

JAYANTH RASAMSETTI


FINCORP
SWAZILAND DEVELOPMENT FINANCE CORPORATION

AGENDA

- Objective
- Data visualization
- Data processing and cleaning
- Model Built and Evaluation
- Improvisation to Model



1 OBJECTIVE & DATA PREPROCESSING

Objectives of current project is:

1) Predicting "class" (1: High energy seismic bump occured) ((target variable V20) is a classification problem

2) The given data set has following attributes (19805 rows and 371 features, 67 values have 0 variance

- Categorical Variables - 61 # 61 vars have values b/n 0, 1, # 34 vars have values b/n 0, 3
- Numerical Variables - 200

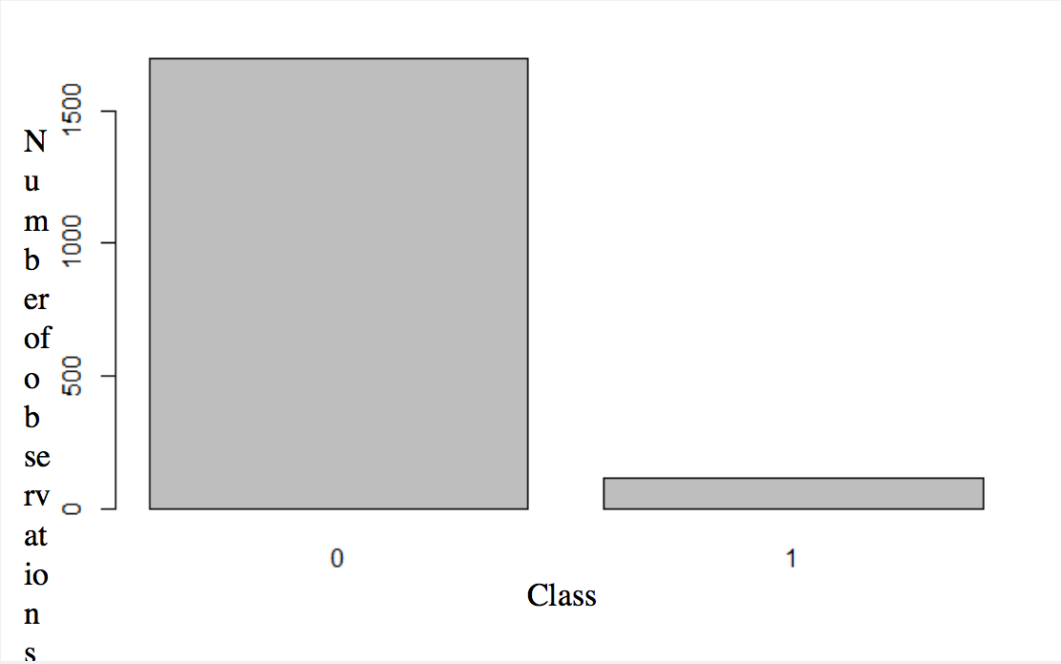
3) 6339 train, 2537 Validation, 3961 test records

1 OBJECTIVE & DATA PREPROCESSING

1) There are 0 missing values Zero

2) Heavy Class imbalance. Used "Smote" on train to prevent class imbalance # 10.04% to 40%

3) Scaled all features



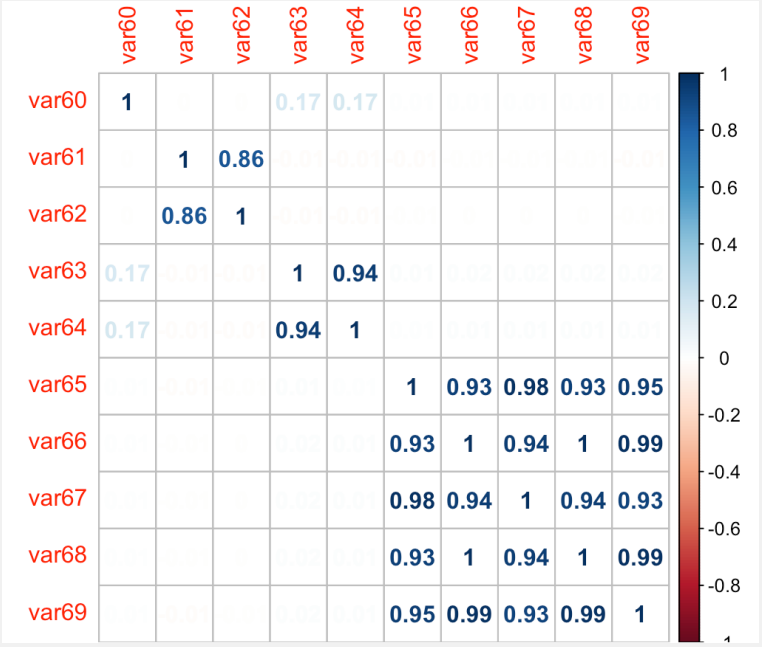
2

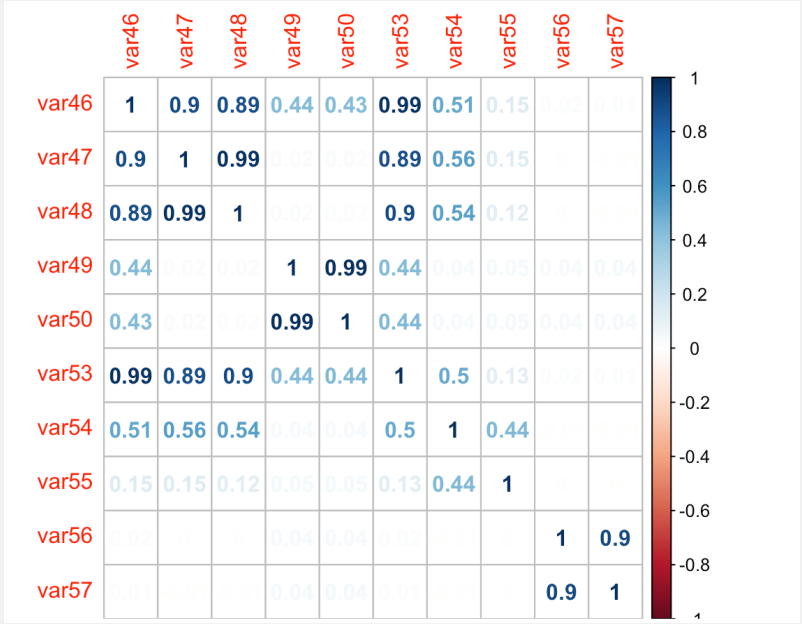
SEISMIC DATAVISUALIZATION - CORRELATION BETWEEN FACTORS

Some trends!

#var 49, var 50 have heavy correlation

var 47, var 49 have heavy correlation





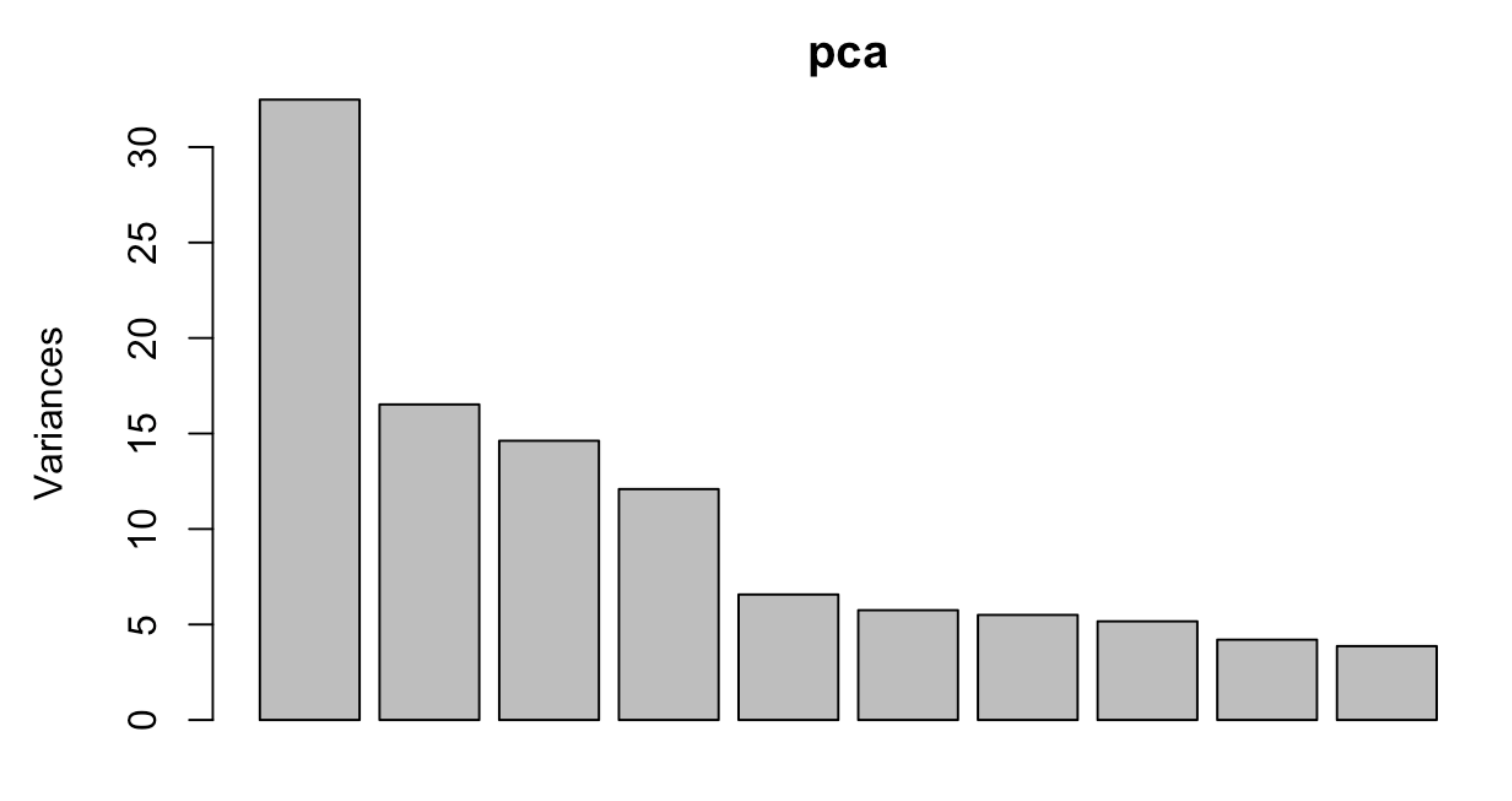
3 DATA PRE PROCESSING

Removed Near Zero Variance and conducted PCA. Retained 15 transformed features

```
> pca <- prcomp(train_rnzv[,!(names(train_rnzv) == "TARGET")], center = T, scale. = T)
> summary(pca)
Importance of components:
                          PC1    PC2     PC3    PC4    PC5     PC6     PC7     PC8
Standard deviation     5.6996 4.0648 3.82372 3.4771 2.5631 2.39748 2.34536 2.27257
Proportion of Variance 0.2166 0.1101 0.09747 0.0806 0.0438 0.03832 0.03667 0.03443
Cumulative Proportion  0.2166 0.3267 0.42419 0.5048 0.5486 0.58691 0.62358 0.65801
                           PC9    PC10    PC11    PC12    PC13    PC14    PC15
Standard deviation     2.05021 1.96627 1.88582 1.78586 1.74281 1.57099 1.51073
Proportion of Variance 0.02802 0.02577 0.02371 0.02126 0.02025 0.01645 0.01522
Cumulative Proportion  0.68603 0.71180 0.73551 0.75678 0.77702 0.79348 0.80869
                          PC16    PC17   PC18    PC19    PC20    PC21    PC22
Standard deviation     1.45195 1.42575 1.3745 1.26762 1.24542 1.19312 1.12843
Proportion of Variance 0.01405 0.01355 0.0126 0.01071 0.01034 0.00949 0.00849
Cumulative Proportion  0.82275 0.83630 0.8489 0.85961 0.86995 0.87944 0.88793
```

3 DATA PRE PROCESSING

Removed Near Zero Variance and conducted PCA. Retained 15 transformed features accounted for over 80 variance



4 MODEL BUILT AND EVALUATION

Typical Accuracy : 81%

AUC value is 0.655

4 MODEL BUILT AND EVALUATION

- We realized that for this seismic domain the AUC is the important aspect. Tried the below models in the stipulated time. KNN along with GBM provided best results

Algorithms	Val Accuracy	Test Accuracy	AUC
Decision Tree	0.8564	0.8634	0.65
KNN (k=1)	0.8868	0.804	***0.67***
Random Forest	0.8699	0.847	0.6355
Bagged DT with tuning	0.8786	0.8687	0.63
GBM	0.9093	0.8705	0.65
SVM - with tuning	0.88	0.85	0.634

5 IMPROVISATIONS TO MODELS WITH TIME

Novel methods to Feature engineering viz. splitting looking at the patterns, binning few of the features etc into one single bin

The Domain knowledge would have made more accurate models

SEISMIC PREDICTION

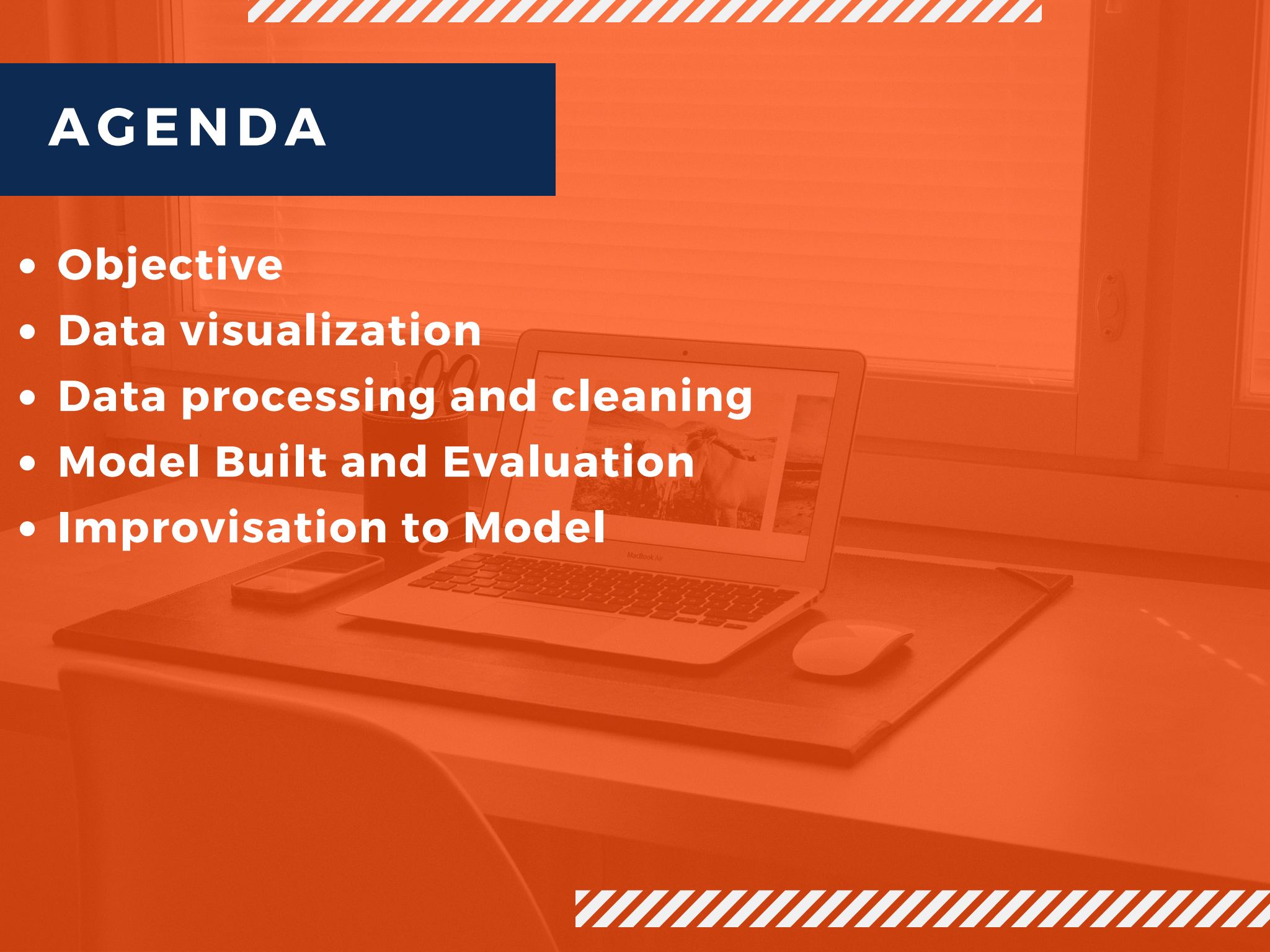
AUGUST 2017

JAYANTH RASAMSETTI

DOLORES OCHOA/AP

AGENDA

- Objective
- Data visualization
- Data processing and cleaning
- Model Built and Evaluation
- Improvisation to Model



1 OBJECTIVE & DATA PREPROCESSING

Objectives of current project is:

1) Predicting "class" (1: High energy seismic bump occured) ((target variable V20) is a classification problem

2) The given data set has following attributes (20)

- Categorical Variables - 5 (V2, V3, V4, V9, class)
- Numerical Variables - 15

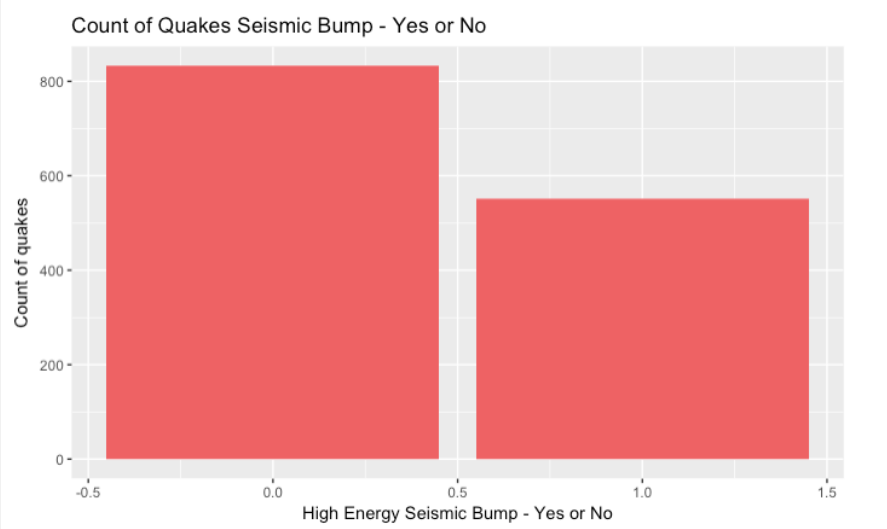
3) 1809 train, 1385 Validation, 1384 test records

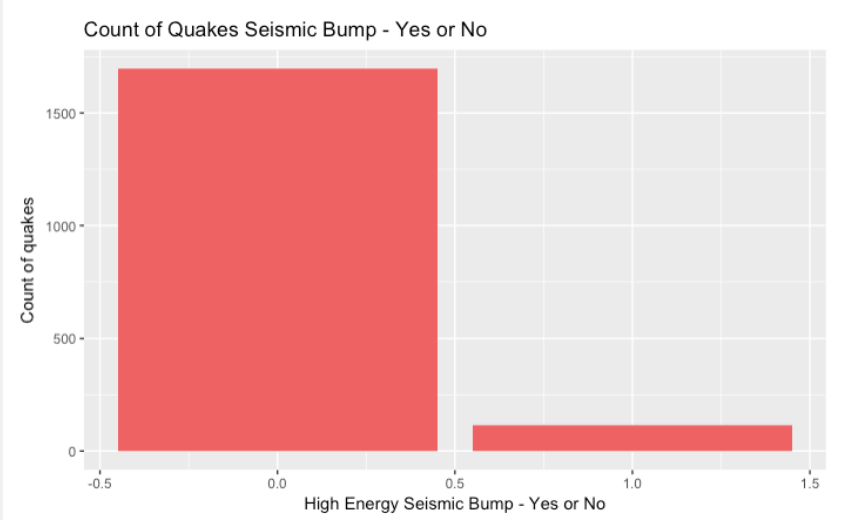
1 OBJECTIVE & DATA PREPROCESSING

1) There are 9 missing values in the original train data set, used "central imputation" for them. Zero missing values in the validation and test data sets

2) Notice that we had to do stratified sampling in the target variable as we had to ensure that the number of High energy seismic bumps and no bumps were similar in both train and validation data

3) Dropped the NULL values columns & converted to numeric the int types





1 OBJECTIVE & DATA PREPROCESSING

Set 3 different seeds!

Before Balancing		
	Train	Validation
0's	1697	833
1's	112	552
Percentage of 1's	**6.19%**	**39.86%**

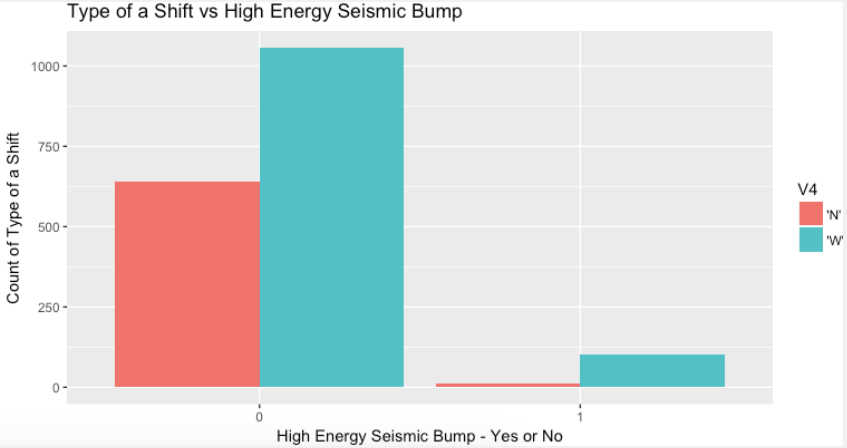
After Balancing	Set.seed (123)	
	Train	Validation
0's	1780	750
1's	456	208
Percentage of 1's	**20.39%**	**21.71%**

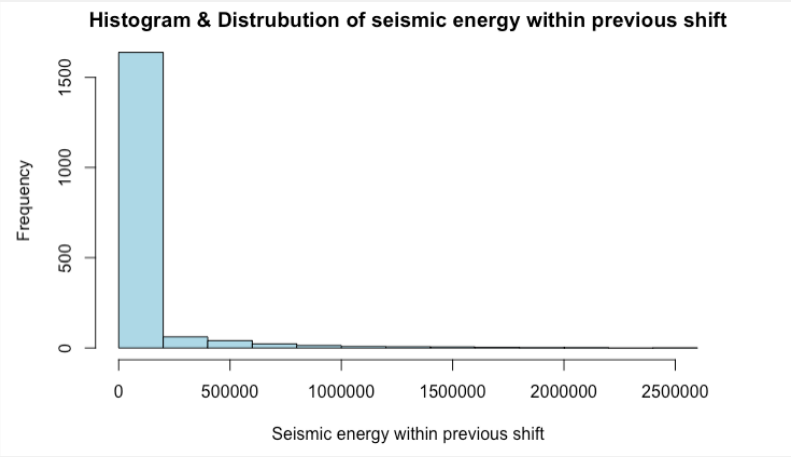
After Balancing	Set.seed (345)	
	Train	Validation
0's	1762	768
1's	474	190
Percentage of 1's	**21.20%**	**19.83%**

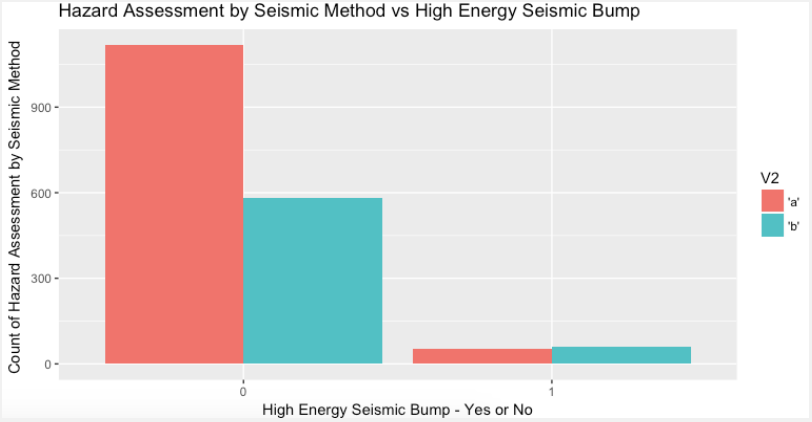
After Balancing	Set.seed (567)	
	Train	Validation
0's	1787	743
1's	449	215
Percentage of 1's	**20.08%**	**22.44%**

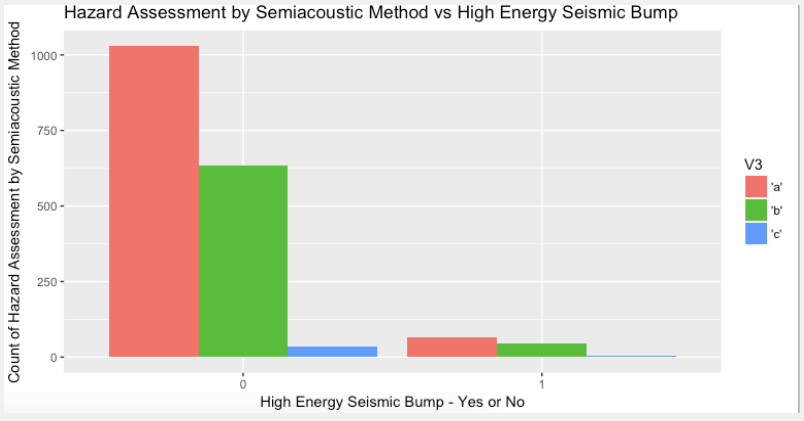
2 SEISMIC DATAVISUALIZATION

Some trends!





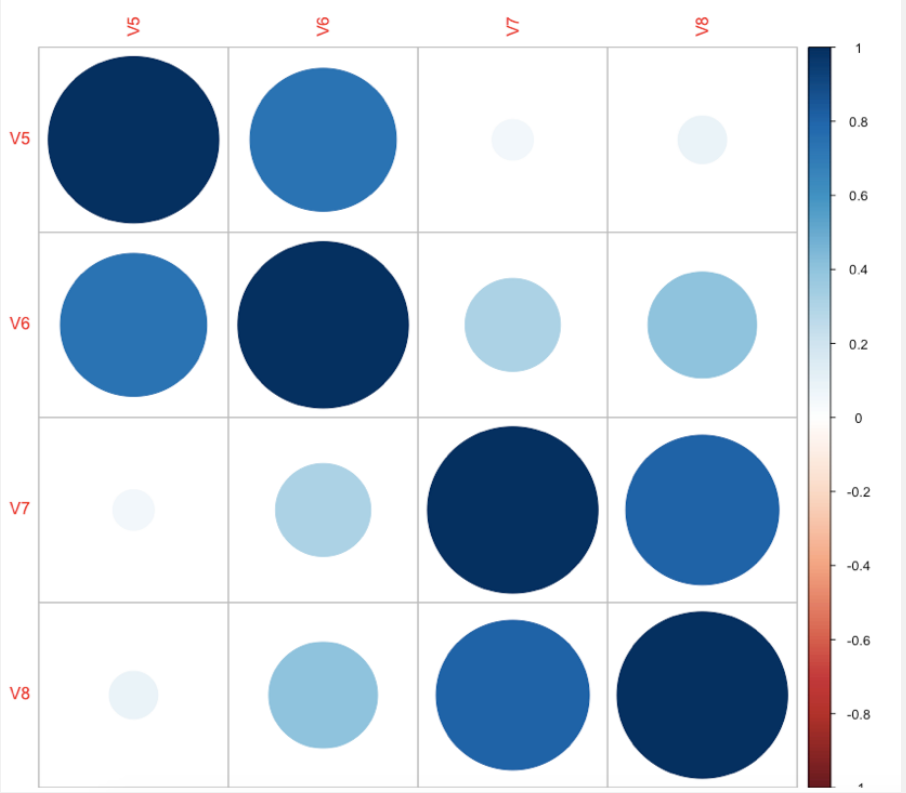




2 SEISMIC DATA VISUALIZATION

Correlation Plots between the different Numerical Attributes:

V6 & V5 ~ Heavily Correlated
V7 & V8 ~ Heavily Correlated



2 SEISMIC DATA VISUALIZATION

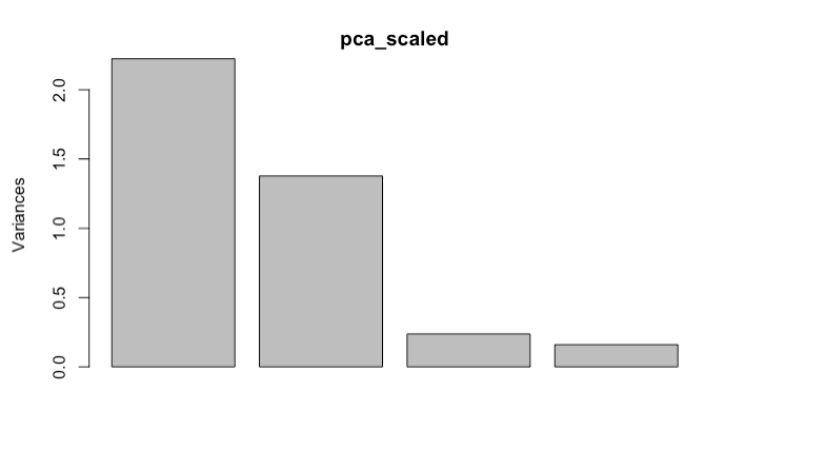
Chi square Independence test on the Categorical Variables

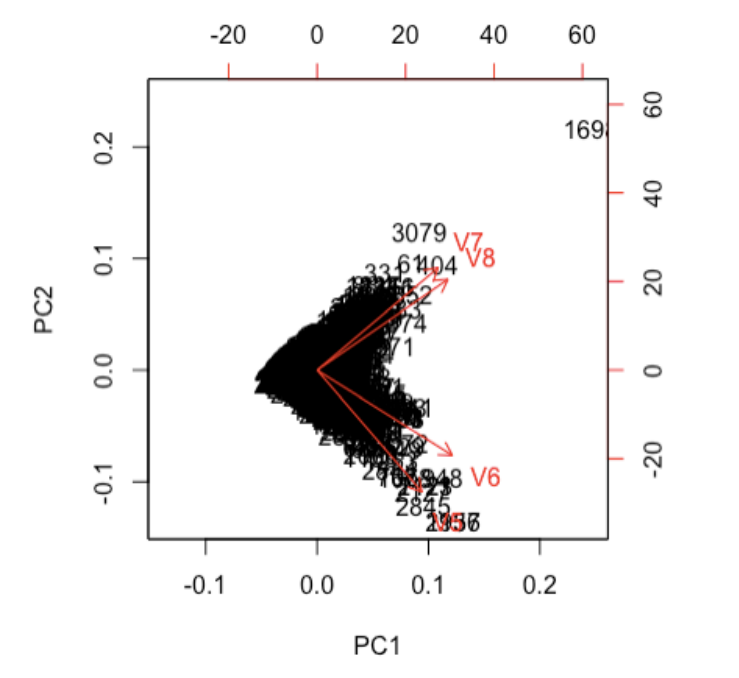
```r
#Chi^2 test for the Categorical attributes V2 and V3

temp<-chisq.test(traindata$V2, traindata$V3, simulate.p.value = TRUE)
temp<-chisq.test(traindata$V2, traindata$V3)
```

```
        Pearson's Chi-squared test

data:  traindata$V2 and traindata$V3
X-squared = 19.221, df = 2, p-value = 6.701e-05
```

3 DATA PRE PROCESSING

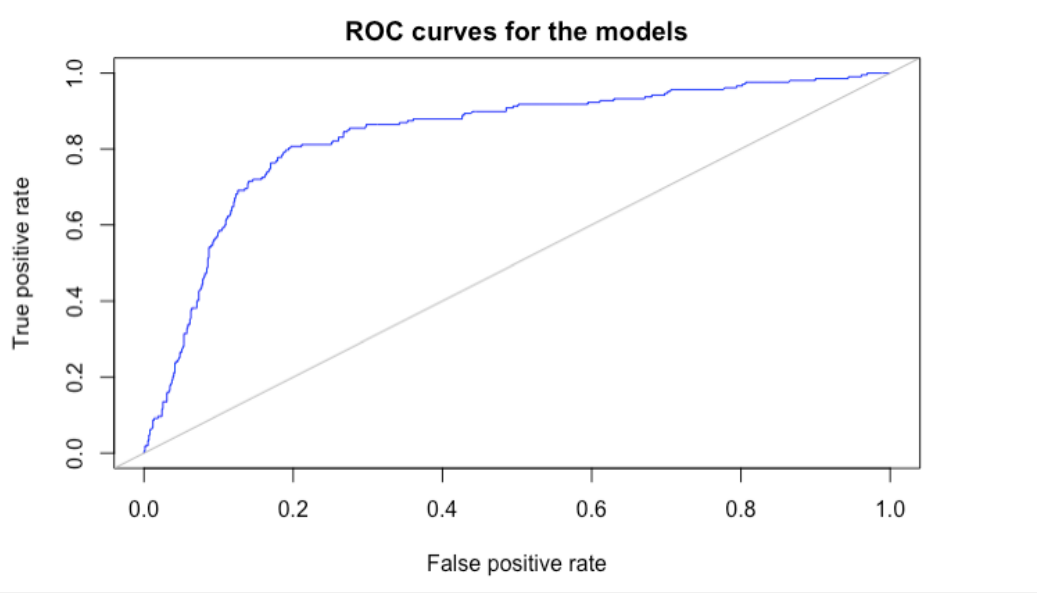
PCA on Numerical features and the important features contributing to the seismic data





4 MODEL BUILT AND EVALUATION

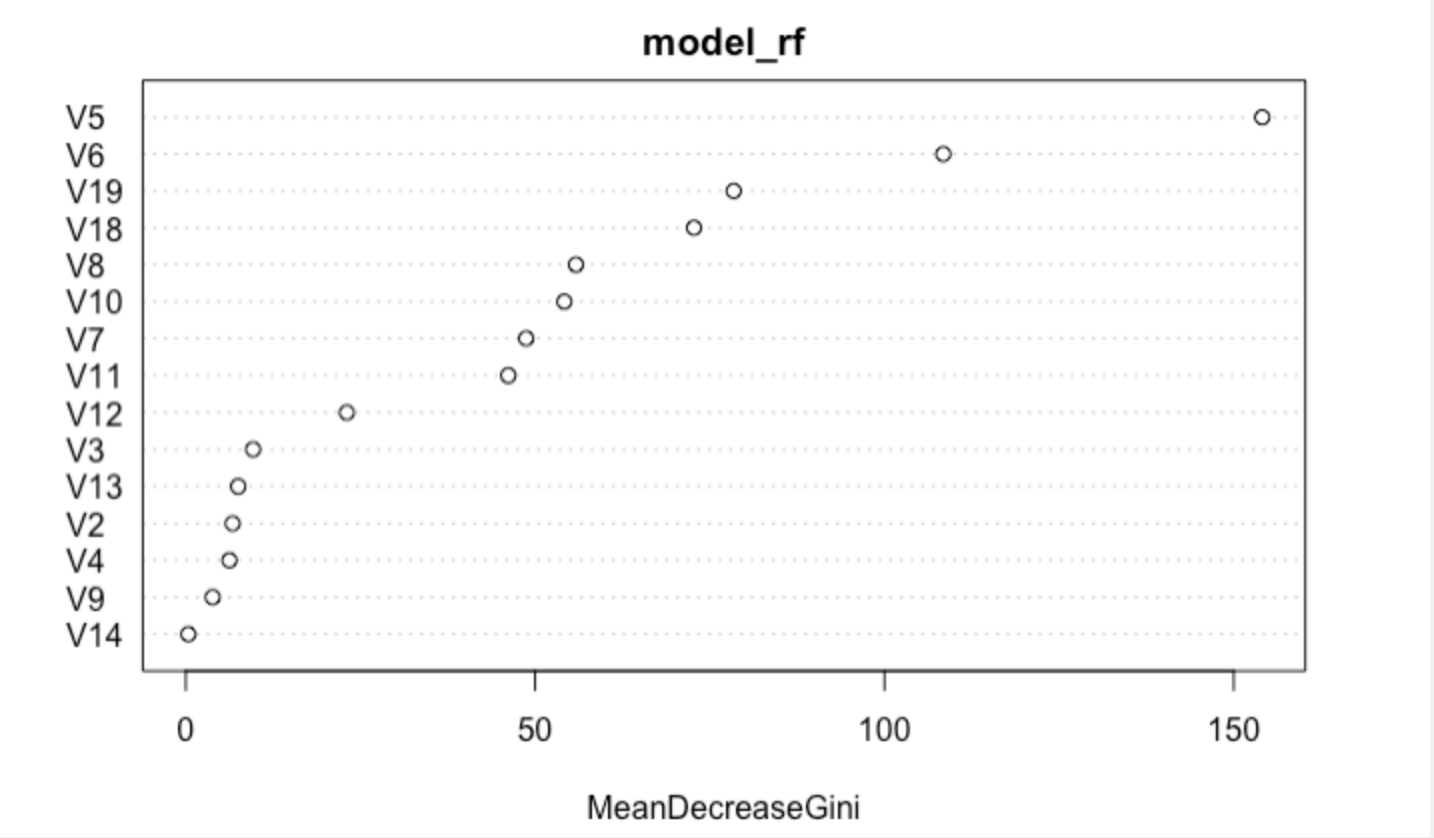
- Logit Confusion matrix resulted in Accuracy : 81.47%
- The AUC value is 0.836




```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 700  48
         1 129  78

               Accuracy : 0.8147
                 95% CI : (0.7885, 0.8388)
    No Information Rate : 0.8681
    P-Value [Acc > NIR] : 1

                  Kappa : 0.3642
 Mcnemar's Test P-Value : 1.819e-09

            Sensitivity : 0.8444
            Specificity : 0.6190
         Pos Pred Value : 0.9358
         Neg Pred Value : 0.3768
             Prevalence : 0.8681
         Detection Rate : 0.7330
   Detection Prevalence : 0.7832
      Balanced Accuracy : 0.7317

       'Positive' Class : 0
```

4 MODEL BUILT AND EVALUATION

- Random Forest



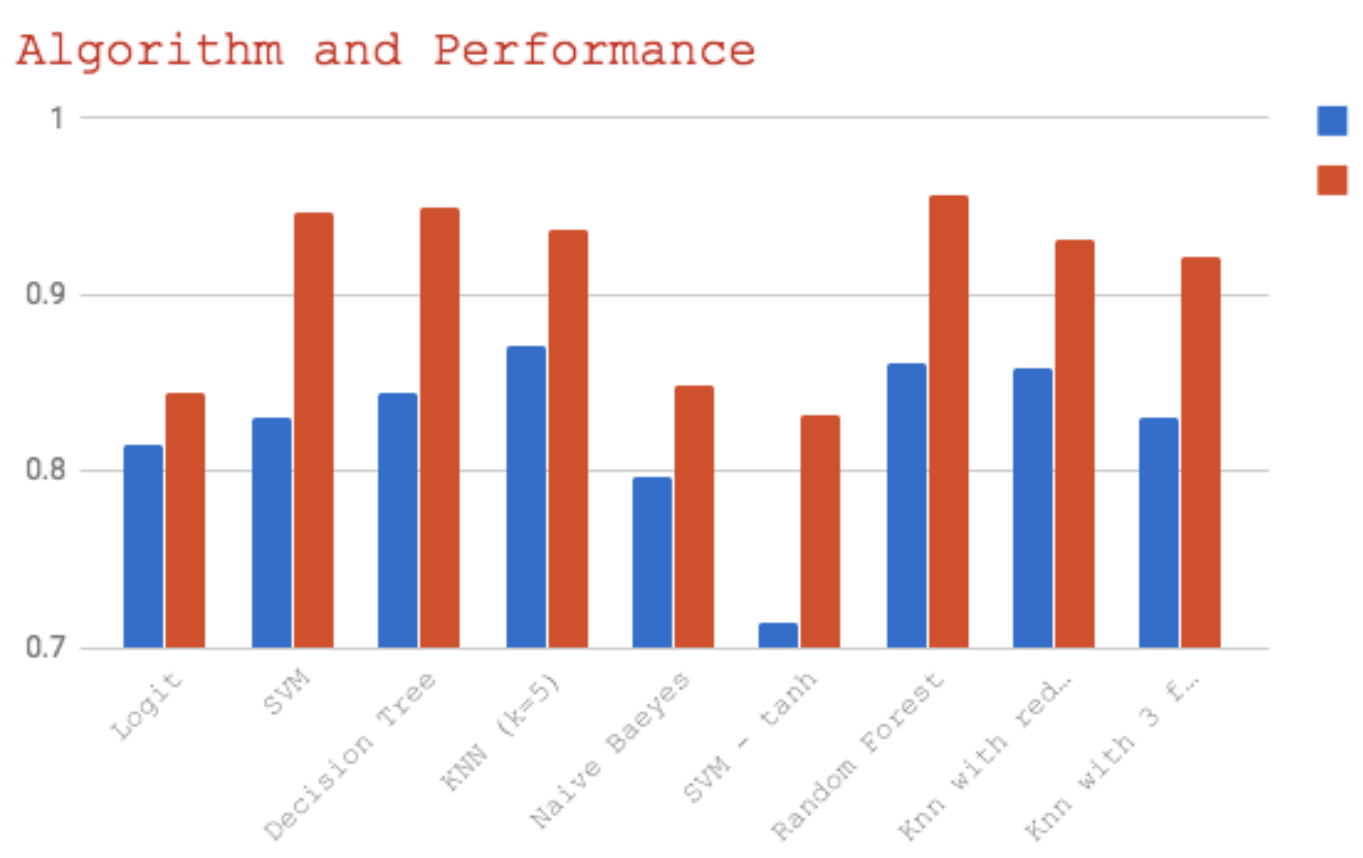
4 MODEL BUILT AND EVALUATION

- We realized that for this seismic domain the Accuracy is the important aspect. Tried the below 11 models in the stipulated time

Algorithms	Train Accuracy	Test Accuracy (Kaggle Score)	Train Sensitivity
Logit	0.8147	0.70087	0.844
SVM	0.8309	NA	0.9467
Decision Tree	0.8445	NA	0.9493
KNN (k=5)	***0.8716***	***0.80491***	***0.9373***
Naive Baeyes	0.7965	NA	0.849
SVM - tanh	0.714	NA	0.832
Random Forest	0.8612	<0.80491	0.956
Knn with reduced features	0.858	NA	0.9307
Knn with 3 features V18, V6, V8	0.8309	NA	0.9208
GBM	In code	In code	In code
Bagged Decision Tree	In code	In code	In code
Stacked Ensemble	In code	In code	In code

4 MODEL BUILT AND EVALUATION

- We realized that for this seismic domain the Accuracy is the important aspect. Tried the below 11 models in the stipulated time; Parameter Tuning Tried Cp = 0.01 and 0.05



5 IMPROVISATIONS TO MODELS WITH TIME

- PCA before the Classification
- Keep positive class reords with missing values and use smote
- Novel methods to Feature engineering viz. splitting looking at the patterns, binning few of the features etc into one single bin
- Either include Bagging to decrease bias, boosting to reduce variance
- The Domain knowledge would have made more accurate models