Southern Methodist University

CSE -7331: Introduction to Data Mining

Dr. Michael Hahsler

Classification

Project 3

Sevil D’monty   47568070
Pritheesh Panchmahalkar - 47524741

Contents

Executive summary

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. In this project, we are use a subset of hospital data where we are going to predict the hospitalization prediction based on the member's past hospitalization claims and other related data. A classification task begins with a data set in which the class assignment is known. For example, a classification model that predicts hospitalization risk could be developed based on observed data for many claims filed by members over a period of time. In addition to the historical hospitalization claims, the data might track drug count, lab count or paydelays, charlson index value and length of stay, and so on. Days in hospital would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case. To predict if a member based on predictors stays in the hospital, we created three different models, trained them with the same dataset and measured the performance of each model using the same test data. These three models are K-Nearest Neighbors, Random Forest, and deep neural network model with 4 dense layers. These models are initially trained with 13 features defined in the data preparation and compared with the model trained with 5 features selected using the Recursive Feature Elimination algorithm. As the hospital data set is similar to randomly generated data, it is difficult to achieve high accuracy, but the models perform a decent job in predicting if a patient stays in the hospital. We have observed that for except the neural network model, the other two train almost in a similar way when trained with all the 13 features and with 5 selected features. The neural network model overfits with the 5 featured subset and the model needs to be modified in order for it perform well with this training data.

Data Preparation

For this classification project we are going to use similar data as we used for previous project, project 2 cluster analysis. Many of the attributes are unnecessary in order to carry out classification so we are not taking them into consideration. The useful attributes for classification project are as follows.

Attribute	**Description**
Claim	Total number of claims filled by member within duration of year 1,2 and 3.
Paydelay	It is number of days between the date of service and date of payment
Charlson Index	A measure of the affect diseases has on overall illness, grouped by significance, that generalizes additional diagnoses.
DrugCount	Count of unique prescription drugs filled by DSFS (Days since First Service). No count is provided if prescriptions were filled before DSFS zero. Values above 6, the 95% percentile after excluding counts of zero, are top-coded as “7+”.
Lab Count	Count of unique laboratory and pathology tests by DSFS (Days Since First Service). Values above 9, the 95% percentile after excluding counts of zero, are top-coded as “10+”.
Length of Stay	Length of stay (discharge date – admission date + 1), generalized to: days up to six days; (1-2] weeks; (2-4] weeks; (4-8] weeks; (8-12 weeks]; (12-26] weeks; more than 26 weeks (26+ weeks).
ClaimsTruncated	ClaimsTruncated (a flag for members who have had claims suppressed. If the flag is 1 for member xxx in DaysInHospital_Y2, some claims for member xxx will have been suppressed in Y1).
DaysInHospital	Days in hospital, the main outcome, for members with claims in Y1. Values above 14 days (the 99% percentile) are top-coded as “15+”.
Heart count	Number of claims filed by the member having primary condition under heart diseases.
Injured count	Number of claims filed by the member having primary condition under MISCELLANEOUS #2 which means that the member got injured by accident.
Renal Count	Number of claims filed based on the PrimaryCondition Renal2 which is related to chronic renal failures.
AgeAtFirstClaim	Age in years at the time of the first claim’s date of service computed from the date of birth; Generalized into ten-year age intervals.

Claims: - The claims attribute gives the number of claims that each member has filed. This gives us total number of claims filed by each member for year 1. The minimum value of claims is 1 and maximum is 43. By this we can assume that if a member has maximum claims could be very sick person with very high charlson index value or there is a possibility that the member was diagnosed with chronic disease such as diabetes, asthma etc. which require regular checkup and follow-up with the doctor. Eighty-eight percent of Americans over 65 years of age have at least one chronic health condition. It is possible that a patient who has just one claim might be suffering from a life-threatening disease too, if the claim is recorded at the end of Year 1. The claims attribute can be grouped along with the primary group condition and help the health care providers understand the claims per each condition, based on which the patient can be given medication. From insurance companies' point of view, the members who have less claims could be profitable most of the time when compared to the members with high claims.

PayDelay: - The attribute PayDelay is number of days between the date of service and date of payment to the vendor. It has value starts from 0 day up till 162 + days. The delay in payment could be due to various reasons like financial situation of the patients, lack of communication of the payment due date or conflicts with the insurance providers. An average PayDelay for each member is calculated using the claims data. This attribute gives an overview of the timeline of the payment process of the members. The health care providers would be happier with the patients who have less average PayDelay when compared to ones with higher average PayDelay. For the sake of this project, all the values in the PayDelay are converted to numeric values, i.e., the value "162+" is converted to 162. Based on the new attribute, the mean of the means of the pay delay of the members is around "52.26%".

Charlson Index: - According to Wikipedia, "The Charlson comorbidity index predicts the one-year mortality for a patient who may have a range of comorbid conditions, such as heart disease, AIDS, or cancer (a total of 22 conditions). Each condition is assigned a score of 1, 2, 3, or 6, depending on the risk of dying associated with each one. Scores are summed to provide a total score to predict mortality."

The patients with less Charlson Index are healthy and the one's with high Charlson index are not healthy i.e., suffering from life threatening diseases. This attribute is important to know about the health condition of a patient. The patients with high Charlson Index are likely to visit the hospitals more often when compared to the patients with low Charlson Index. Also, the hospital bills for the patients with high Charlson Index are likely to be higher. From the insurance companies point of view, it could be profitable for them to focus on patients with low Charlson index, as they are likely to pay less and visit the hospitals less often. The doctors/researchers based on Charlson index can focus their research on diseases that result in high Charlson index and design medicines that can cure such diseases.

The values in the claims data is modified as show in the table below.

Charlson Index	Value taken for charlson index
0	0
1-2	1.5
3-4	3.5
5+	5.5

Table: Charlson Index values and modified values

Drug count: - The Attribute Drug count is total number of drugs prescribed by the doctor to the member during the year 1. In the drugs data, the values above 6 are top-coded as “7+”. To keep the data numeric, this value has been changed to 7. The minimum value of Drug count is 1 and maximum value is 84. The drug count gives information about the condition of the patient. The more the drug count value of a member, the higher the probability that a patient is suffering from a disease that needs long time to be cured or can never be cured. The lower the drug count of a patient, the healthier is the patient which is the objective of the health care providers. The patients should be careful when using drugs, as the drugs could be a reason for illness too. The patients who take regular prescriptions should try to eliminate few drugs, so they can avoid the effects of the drugs.

Lab count: - The Attribute Lab count is the number of Laboratory tests prescribed by the doctor to the member in the year 1 data. In the labs data, the values above 9 are top-coded as “10+”. To keep the data numeric, this value has been changed to 10. The minimum value of lab counts is 1 and maximum is 80. Lab tests like blood tests, urine tests, MMR, X-rays etc. are necessary to determine the health condition of the patient. We can assume that the member having maximum number of Lab count are the least healthy people when compared others. Also, there is possibility that members with higher Lab count could be suffering from chronic disease for which that member must visit doctor on regular basis and get the tests done.

Length of stay: The Attribute length of stay is the total number of days a member spent in the hospital of health care facility. There are missing values in the length of stay attributes which we have changed to 0 days, we have considered that a member whose length of day is missing might not admitted to hospital but instead that member was admitted in morning and later discharged in evening, or some members got treated in ambulance and got discharged on same day. we have changed values for length of stay attribute from string to numeric is as follows,

Length of Stay	Value taken for Length of Stay
Missing data	0
1 day	1
2 days	2
3 days	3
4 days	4

5 days	5
6 days	6
1-2 Weeks	7
2-4 weeks	14
4-8 weeks	28
8-12 weeks	56
12-26 weeks	84
26+ weeks	182

The sum of length of stay of each member is calculated based on their claims in the claims data for the clustering methods. The longer the length of stay, the higher is the probability that the patient is suffering from a disease that requires more medical care. It could also be possible that the missing values in the data could be due to the reason of privacy, but for the sake of simplicity, we have chosen a value 0 instead of NA. The highest length of stay was observed as 562 days, which results due to the way the data was changed. It is possible that the patient has been admitted for 2-4 weeks throughout the year or may be the claims were recorded multiple times each kind of Primary Condition group, place of service, specialty.

Dih_levels: The attribute days in hospital gives days spent in hospital by the member. The data in this attribute ranging from 0 to 15 maximum. Let us consider the Histogram plotted using the data which omits the count when days in hospital is 0. It is clear that the frequency is very high for a day in the hospital and drastically low for the rest of the days except for the last value which is 15. It is possible that the value 15 is the result of encoding “15+ days” as 15.

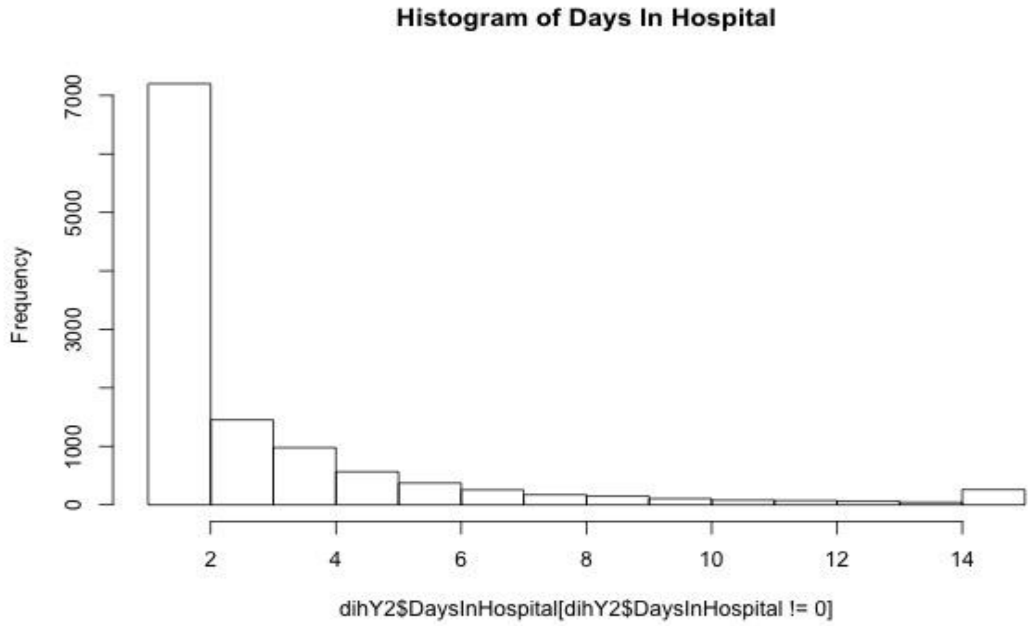

Figure: Histogram of Days in Hospital

We decided to divide this data in 3 classes namely as ‘No Stay’, ‘Short Stay’ and ‘Long stay’. To divide the days in hospital among three classes we chose the median of days in hospital whose value is not zero. In this way, the classes could be nearly half way divided among short stay and long stay. If a member was admitted for 0 days then he/she is in No Stay level, same for the member who was admitted in hospital for 1 to 2 days then he is part of Short Stay level and lastly the member who were admitted for more than 2 days are categories as Long Stay. By this we can assume that the person who visits doctor for checkup or a surgery which do not require hospitalization comes under No Stay. The Member who need medical treatment up to 2 days in hospital are comes under Short stay such as Members who had minor accident, fractures etc. Long stay category there are members who might require long stay mostly more than 2 days such as member with very high charlson index and old age members or members who had serious accident, deliveries and might take long time for recovery. Approximately we got 83 % of data comes under the class ‘No Stay’ and nearly 10% of data comes under ‘Short Stay’ and 7% of data comes under ‘Long Stay’.

AgeAtFirstClaim: This attribute gives age of the member when he/she filed for their first claim or in other words the age of the member at which he/she filed their first claim. This attribute has values with the interval of 10 years such as 0-9, 10-19 till 80+. This attribute is very complicated as you cannot be sure if the member who filed for their first claim what was his/her actual age as this attribute has range of ages.

Heart claims count: We have created this attribute named as Heart count. The Heart Claims count attribute gives the count of claims a member made when suffering from an ailment related to heart. To get the count of such claims the primary condition group could be AMI (ACUTE MYOCARDIAL INFARCTION), HEART4 (ATHEROSCLEROSIS AND PERIPHERAL VASCULAR DISEASE), CATAST (CATASTROPHIC CONDITIONS which includes cardiac arrest), CHF (CONGESTIVE HEART FAILURE), MISCHRT (MISCELLANEOUS CARDIAC), HEART2 (OTHER CARDIAC CONDITIONS), PERVALV (PERICARDITIS) or STROKE. For the models, as the objective is to predict the number of days a member spends in the hospital, it is possible that the members suffering from problems related to heart are likely to spend more in hospital, say ICU after the cardiac arrest. The mean of length of stay of the members based on claims on above mentioned PrimaryConditionGroups is 0.46.

Injured claims count: The injured claims count attribute gives the count of claims a member has made when suffering from injuries. The injured claims count is calculated by using the PrimaryConditionGroup ‘MSC2a3’ (Misc#2) which deals with external causes of injuries. The patient is likely to spend more time in the hospital due to injuries as the patient might have broken ribs, limbs etc. and needs to spend some time in the hospital.

Renal claims count: The renal claims count attributes gives the count of claims a member has made when suffering from problems related to kidney like chronic renal failure, end-stage renal disease and kidney transplants. These problems are likely to make the patient spend more time in the hospital due to dialysis, or the patient may be under observation for a few days etc.

Dataset: The dataset has a class imbalance problem as most of the patient do not really stay in the hospital unless they have a serious health issue.

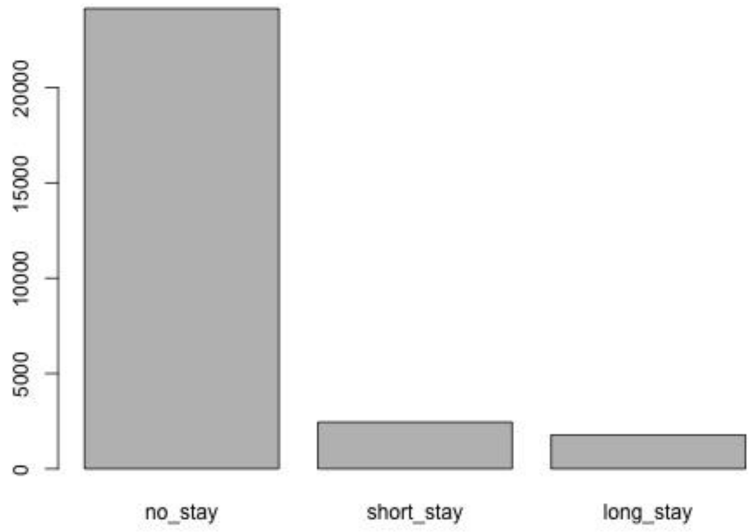

Figure: Histogram of Class levels have class imbalance problem

It is clear that there is class imbalance problem and the model would just predict no_stay even if the label for the class would be "short stay" or "long stay". Hence, we will work on a subset of this data which contains 2500 rows of each of the three classes which is divided into test and train data. The 2500 rows for each of the classes can be obtained using strata method of the package sampling. This method oversamples the data wherever required to balance the classes.

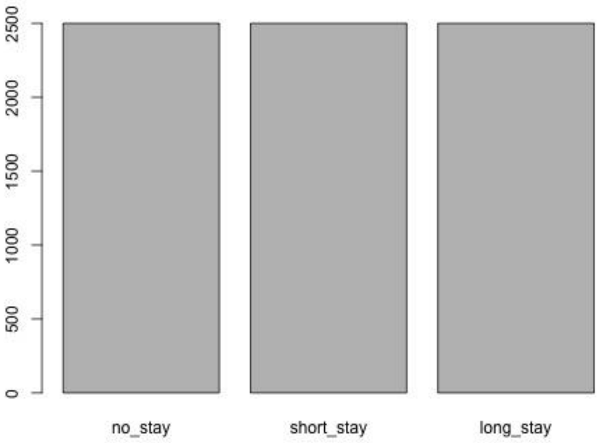

Figure: Resolved class imbalance problem

The train data is 80% of the subset and the rest is used as test data. The model initially works on all the features of the dataset like AgeAtFirstClaim, claims, PayDelay, CharlsonIndex, LengthOfStay, DrugCount, LabCount, ClaimsTruncated, injury_count, heart_count, renal_count and dih_levels.

Later, we choose features based on result of feature selection algorithms and see how well the selected features work.

The train data which is 80% of the data is divided among the three classes as follows

No_stay	Short_stay	Long_stay
2003	2010	1987

The test data which is 20% of the data is as follows

No_stay	Short_stay	Long_stay
497	490	513

Feature subset selection

Recursive Feature Elimination (RFE), a method for feature subset selection is provided by the caret R package. It is a feature selection method that fits a model to all the features and removes the weakest ones until the specified number of features is reached. Each of the features is ranked according to its importance to the model.

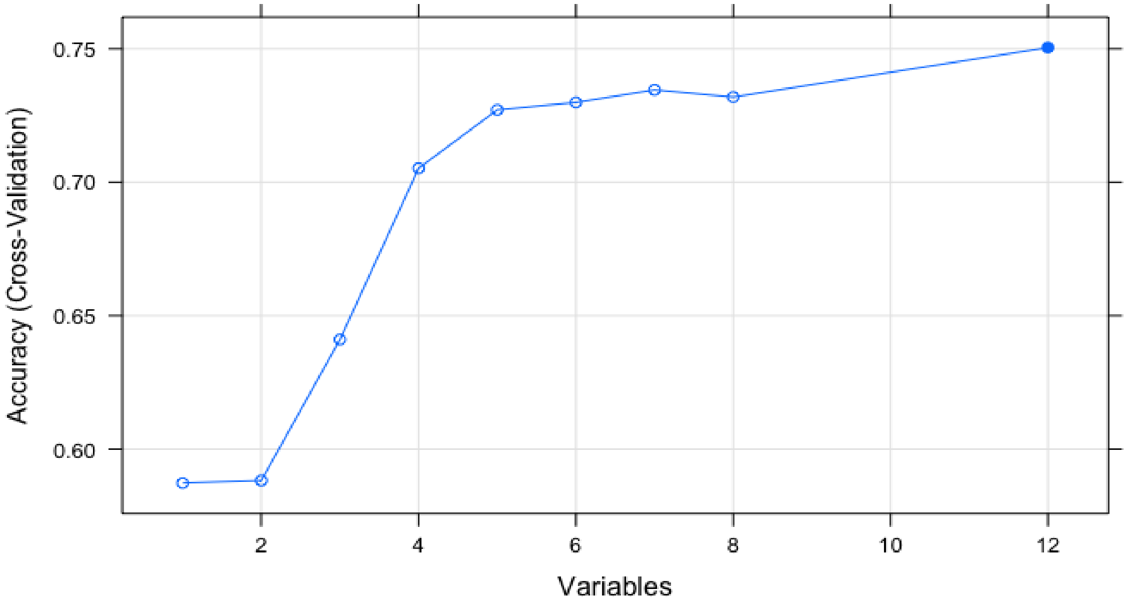

Figure: Feature subset selection

Based on the result of RFE, we select the features which are important for the model and try to train the model based on the data and assess it. The top 5 features are PayDelay, claims, DrugCount, injury count and LabCount.

Modeling

K Nearest Neighbors

KNN is a non-parametric supervised learning technique in which we try to classify the data point to a given category with the help of training set. In simple words, it captures information of all training cases and classifies new cases based on a similarity.

Predictions are made for a new instance (x) by searching through the entire training set for the K most similar cases (neighbors) and summarized the output variable for those K cases. In classification this is the mode (or most common) class value.

The subset of the data i.e., training data which has 6000 rows has been used to train the knn model. Before the data can be passed to the knn model, the data has to be scaled. The table below shows results of k-means algorithm, knn model gives a good accuracy and the kappa values for k = 1 and keeps decreasing as the value of k keeps increasing.

K	Accuracy	Kappa
1	0.96	0.94
2	0.73	0.60
3	0.69	0.54
4	0.65	0.47
5	0.62	0.43
6	0.59	0.39
7	0.59	0.38

Table: Results of k-means algorithm

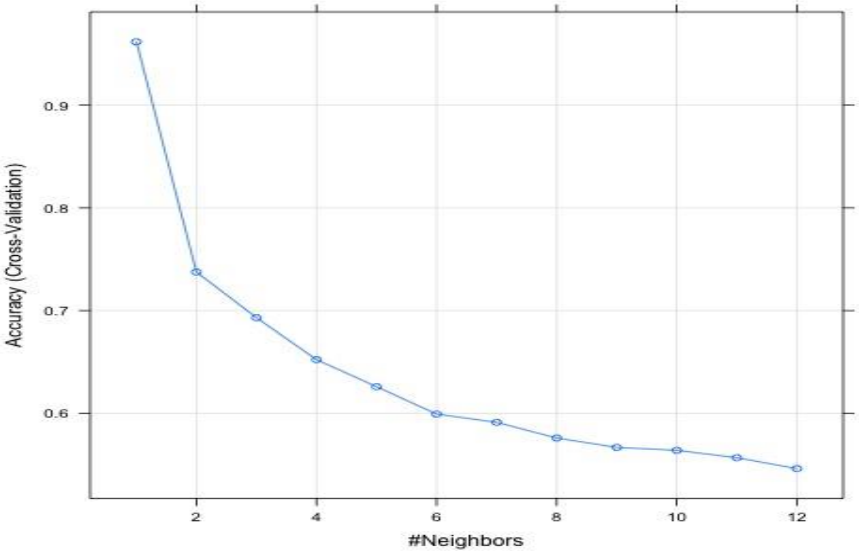

Figure: Result of knn algorithm plotted using #neighbors and accuracy

The figure above shows plot of knn model. It is clear that as the count of neighbors gets higher, the accuracy of the model decreases.

Assessing the model's performance

The assessment of the model uses the test data which was 20% of the subset. Table below shows the test data we are using for validating a model that trained on the train data.

No_stay	Short_stay	Long_stay
497	490	513

The knn model predicts the class labels for the test data and we construct a confusion matrix and statistics based on these predictions.

Confusion Matrix

Prediction	No_stay	Short_stay	Long_stay
No_stay	172	55	23
Short_stay	80	184	54
Long_stay	73	49	210

Overall Statistics

Accuracy	0.68
95% CI	(0.6584, 0.7062)
No Information Rate	0.342
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.5238
Mcnemar's Test P-Value	0.0002343

CI - a confidence interval (CI) is a type of interval estimate, computed from the statistics of the observed data, that might contain the true value of an unknown population parameter.

The value of accuracy is greater than no information rate. Hence, the model is doing good job on predicting class label for test data.

	No Stay	**Short Stay**	**Long Stay**
Sensitivity	0.54	0.72	0.78
Specificity	0.86	0.82	0.85
Pos Pred Value	0.66	0.66	0.73
Neg Pred Value	0.79	0.86	0.88
Prevalence	0.33	0.33	0.34
Detection Rate	0.18	0.23	0.27
Detection Prevalence	0.27	0.36	0.37
Balanced Accuracy	0.70	0.77	0.81

Sensitivity: The Sensitivity gives the true positive rate which measures the actual positives that are correctly identified as positives.

Sensitivity = TP / (TP + FN)

Specificity: Specificity gives the true negative rate which measures the actual positives that are correctly identified as negatives.

Specificity = TN / (TN + FP)

Pos Pred Value: The positive predictive value is defined as the percent of predicted positives that are actually positive

Pos Pred Value = TP/ (TN + FP)

Neg Pred Value: the negative predictive value is defined as the percent of negative positives that are actually negative.

Neg Pred Value = TN / (TN + FN)

Prevalence: a numeric value for the rate of the "positive" class of the data

Prevalence = TP+FP/(TP+FP+TN+FN)

KNN model performs well on this data in predicting the length of stay of a patient in the hospital taking into account the data records are. Based on the statistics, the model predicts that a person stays in hospital and hence the sensitivity of no stay is low, and specificity is high. It is possible that a person may have many claims or high charlsonIndex and the model predicts that he stays in hospital but, the person might not have to stay in the hospital.

The sensitivity for short_stay and long_stay is relatively higher than the no_stay class label. Hence, the model performs well in predicting that a member has to stay in the hospital when compared to predicting not staying in hospital. It is better for the model to predict that a member stays in hospital in case of emergency to get healthier. However, the data has records which have class labels that could be predicted with great difficulty. But here, the objective of the model should be to reduce the prediction where result in no stay but actually it is long_stay or short_stay.

KNN trained with selected features

Based on the result of RFE, we select the features which are important for the model and try to train the model based on the data and assess it. The top 5 features are PayDelay, claims, DrugCount, injury count and LabCount.

K	Accuracy	Kappa
1	0.97	0.95
2	0.75	0.63
3	0.69	0.54
4	0.65	0.47
5	0.61	0.42
6	0.60	0.40
7	0.58	0.37
8	0.57	0.36
9	0.57	0.35
10	0.56	0.34

Table: Results of KNN model on feature selected dataset

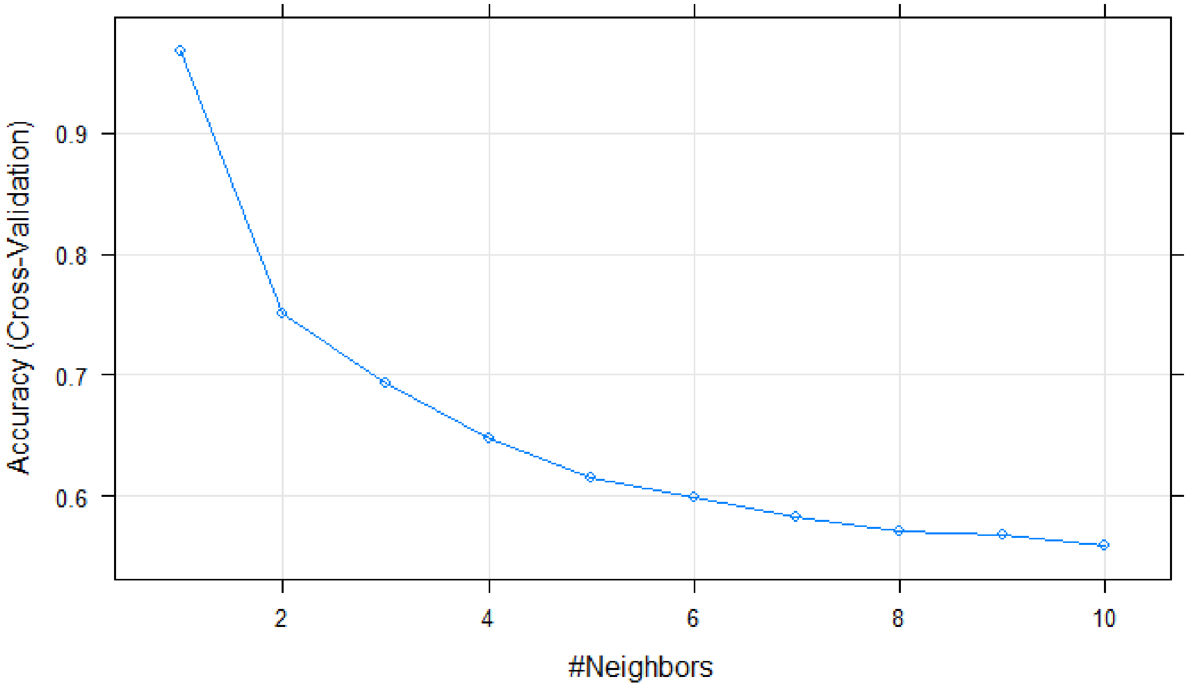

Figure: Plot showing the result of KNN model

Assessing the model’s performance

To assess the performance of the knn model trained using the training data with 6000 rows, we use the test data which consists of 1500 rows consisting of approximately 500 each with class labels no_stay, short_stay and long_stay.

We use the trained model to predict the class labels for the test data and validate them against the actual class labels. The results of prediction are shown below.

Confusion Matrix

Prediction	No_stay	Short_stay	Long_stay
No_stay	265	85	54
Short_stay	133	350	58
Long_stay	99	55	401

Overall Statistics

Accuracy	0.68
95% CI	(0.653, 0.701)
No Information Rate	0.342
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.51
Mcnemar’s Test P-Value	2.642e-05

	No Stay	**Short Stay**	**Long Stay**
Sensitivity	0.53	0.71	0.78
Specificity	0.86	0.81	0.84
Pos Pred Value	0.66	0.65	0.72
Neg Pred Value	0.79	0.85	0.88
Prevalence	0.33	0.32	0.34
Detection Rate	0.17	0.23	0.27
Detection Prevalence	0.27	0.36	0.37
Balanced Accuracy	0.70	0.77	0.81

The sensitivity of No_stay class is low. Hence, it is obvious that the model is predicting less True Positives when compared to False Negatives for No_Stay class. The objective of the model would be to reduce the sensitivity for short_stay and long_stay class as the model predicts that a member doesn’t stay in the hospital, but it is not the case. But the model does a pretty decent job with the data provided, as it is difficult to perform prediction on the hospital dataset. This model is comparable to the model which used all the features in the table and the accuracy is only 0.5% less.

	No Stay		Short Stay		Long Stay	
	KNN1	KNN2	KNN1	KNN2	KNN1	KNN2
Sensitivity	0.53	0.46	0.64	0.61	0.75	0.73
Specificity	0.86	0.84	0.78	0.78	0.78	0.80

Table: Comparing the results of model before and after feature selection

Model 1 which used 10 columns is not very dissimilar to model 2 which used 5 columns. KNN2 has a slightly less accuracy when compared to KNN1. It would be better to go with KNN1 which predicts the result of short_stay and long_stay as they are more important to be predicted right than no stay class.

Random Forest

Random Forest is one such very powerful ensembling machine learning algorithm which works by creating multiple decision trees and then combining the output generated by each of the decision trees. Decision tree is a classification model which works on the concept of information gain at every node. For all the data points, decision tree will try to classify data points at each of the nodes and check for information gain at each node. It will then classify at the node where information gain is maximum. It will follow this process subsequently until all the nodes are exhausted or there is no further information gain. Decision trees are very simple and easy to understand models; however, they have very low predictive power. In fact, they are called weak learners.

This model uses the same train data which we used to train the knn model.

The table below shows the results of the random forest algorithm, the final value used for the model was mtry = 7 with an accuracy of 97.2%. Here mtry is number of variables randomly sampled as candidates at each split.

mtry	Accuracy	Kappa
2	0.92	0.88
4	0.97	0.96
7	0.97	0.96
9	0.97	0.96
12	0.97	0.96

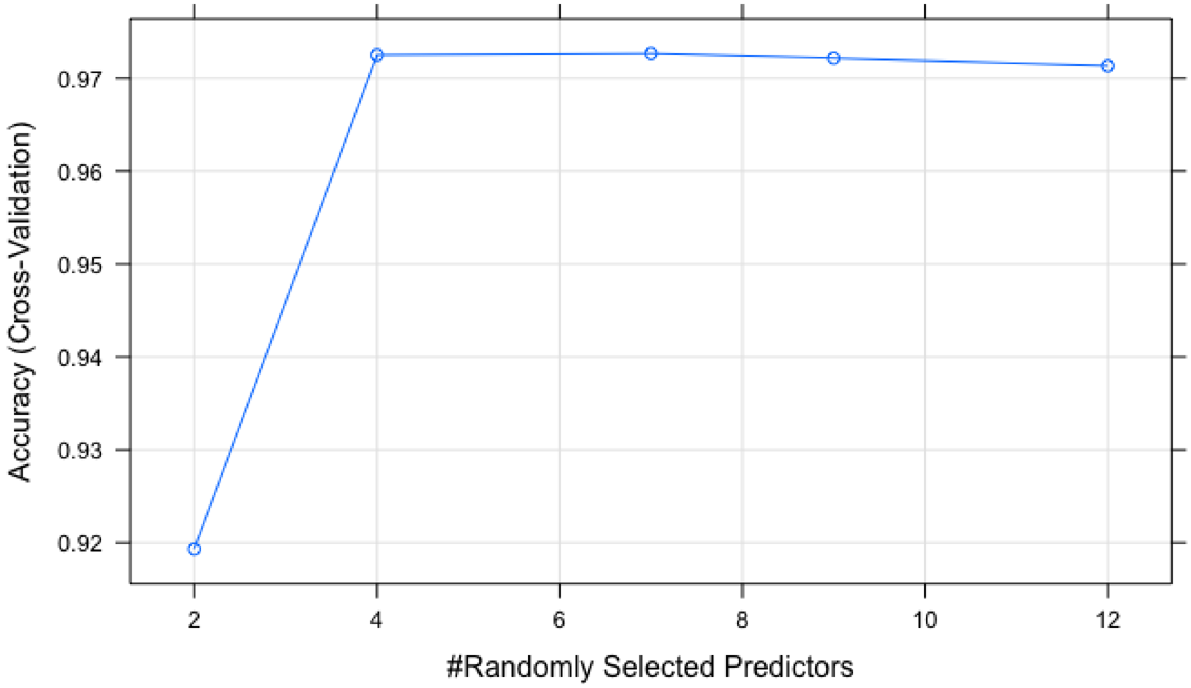

Figure: Plot showing the result of random forest fit on train data

Assessing the model's performance

The assessment of the random forest model uses the test data which is 20% of the subset which used approx. 1500 rows. Table below shows confusion matrix which we have constructed from the test data.

Confusion Matrix

Prediction	No_stay	Short_stay	Long_stay
No_stay	338	105	51
Short_stay	93	327	47
Long_stay	66	58	415

Based on predictions of the random forest model we have constructed statistics shown below.

Overall Statistics

Accuracy	0.72
95% CI	(0.696, 0.742)
No Information Rate	0.342
P-Value [Acc > NIR]	< 2e-16
Kappa	0.57
Mcnemar's Test P-Value	0.2836

	No Stay	**Short Stay**	**Long Stay**
Sensitivity	0.68	0.67	0.81
Specificity	0.84	0.86	0.87
Pos Pred Value	0.68	0.70	0.77
Neg Pred Value	0.84	0.84	0.90
Prevalence	0.33	0.33	0.34
Detection Rate	0.22	0.22	0.28
Detection Prevalence	0.32	0.31	0.36
Balanced Accuracy	0.76	0.76	0.84

The statistics from above table we can assume that the model predicts that the class no_stay and short_stay have low sensitivity value compare with the long_stay class which means a member is either not staying or short staying at the hospital. We could say that our model is able to find members who will not stay in hospital with 68% accuracy, same with Short stay 67% and long stay with 81% which is highest of all. The specificity values for all three classes are above 84% which means this model is able to find members who will not stay in the hospital with the accuracy of 84%, which gets increase for Short stay and Long stay classes.

Random forest trained with selected features

Based on the result of RFE, we select the features which are important for the model and train the model based on the data and assess it. The top 5 features are PayDelay, claims, DrugCount, injury count and LabCount.

mtry	Accuracy	Kappa
2	0.97	0.95
3	0.97	0.95
4	0.97	0.95
5	0.97	0.95

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was mtry = 5

Assessing the model’s performance

The trained model is now assessed using the test data. The table below shows the confusion matrix.

Confusion Matrix

Prediction	No_stay	Short_stay	Long_stay
No_stay	308	96	58
Short_stay	113	341	43
Long_stay	76	53	412

The model predicts that the member does not stay in the hospital for 154 times while they were supposed to stay in hospital. The model need to reduce this error rate.

Overall Statistics

Accuracy	0.71
95% CI	(0.683, 0.730)
No Information Rate	0.342
P-Value [Acc > NIR]	< 2e-16
Kappa	0.56
Mcnemar's Test P-Value	0.1837

	No Stay	**Short Stay**	**Long Stay**
Sensitivity	0.62	0.70	0.80
Specificity	0.85	0.84	0.87
Pos Pred Value	0.67	0.69	0.76
Neg Pred Value	0.82	0.85	0.89
Prevalence	0.33	0.33	0.34
Detection Rate	0.20	0.22	0.27
Detection Prevalence	0.31	0.33	0.36
Balanced Accuracy	0.73	0.77	0.84

The sensitivity of the class No_stay is relatively lower than the class Short_stay and Long_stay. This is because the model predicts a member would stay in the hospital but in reality, the patient doesn't. This is better when compared to the model predicting that the patient stays at their home, but the patient stays in hospital.

	No Stay		Short Stay		Long Stay	
	RF1	RF2	RF1	RF2	RF1	RF2
Sensitivity	0.68	0.62	0.66	0.70	0.81	0.80
Specificity	0.84	0.85	0.86	0.84	0.87	0.87

Both the models perform almost similarly with a minute difference in accuracy of 1%.

Neural network

Neural network is an information-processing machine and can be viewed as analogous to human nervous system. Just like human nervous system, which is made up of interconnected neurons, a neural network is made up of interconnected information processing units. The information processing units do not work in a linear manner. In fact, neural network draws its strength from parallel processing of information, which allows it to deal with non-linearity. Neural network becomes handy to infer meaning and detect patterns from complex data sets. The data that we have used for the model is the subset of members data which have at least 1 claim based on the physical injury.

The following model has been designed using the keras library to train the data. It is deep network with 4 dense layers and 3 dropout layers in between each dense layer to stop the network from overfitting the data.

Activation functions used in the model

ReLU: ReLu stands for rectified Linear Unit. The function of ReLU is given by the formula

$$f(x) = \max(0, x)$$

ReLU improvises the neural networks by speeding up training which is achieved by simple gradient computation.

Sigmoid: A sigmoid function is a mathematical function having a characteristic "S"-shaped curve. The sigmoid function is given by the formula

$$S(x) = \frac{1}{1 + e^{-x}}$$

Relu activation function has been used for first two dense layers, sigmoid has been used for third dense layer and softmax has been used for the last dense layer.

```
Model
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_17 (Dense)             (None, 256)               3328
_________________________________________________________________
dropout_13 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_18 (Dense)             (None, 128)               32896
_________________________________________________________________
dropout_14 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_19 (Dense)             (None, 64)                8256
_________________________________________________________________
dropout_15 (Dropout)         (None, 64)                0
_________________________________________________________________
dense_20 (Dense)             (None, 4)                 260
=================================================================
Total params: 44,740
Trainable params: 44,740
Non-trainable params: 0
```

The same training data which has been used for other two models has been used to train the deep neural network as well. The training data has approximately 2000 records each of class no_stay, short_stay or long_stay. It has been trained using 6000 rows.

Results of deep neural network model

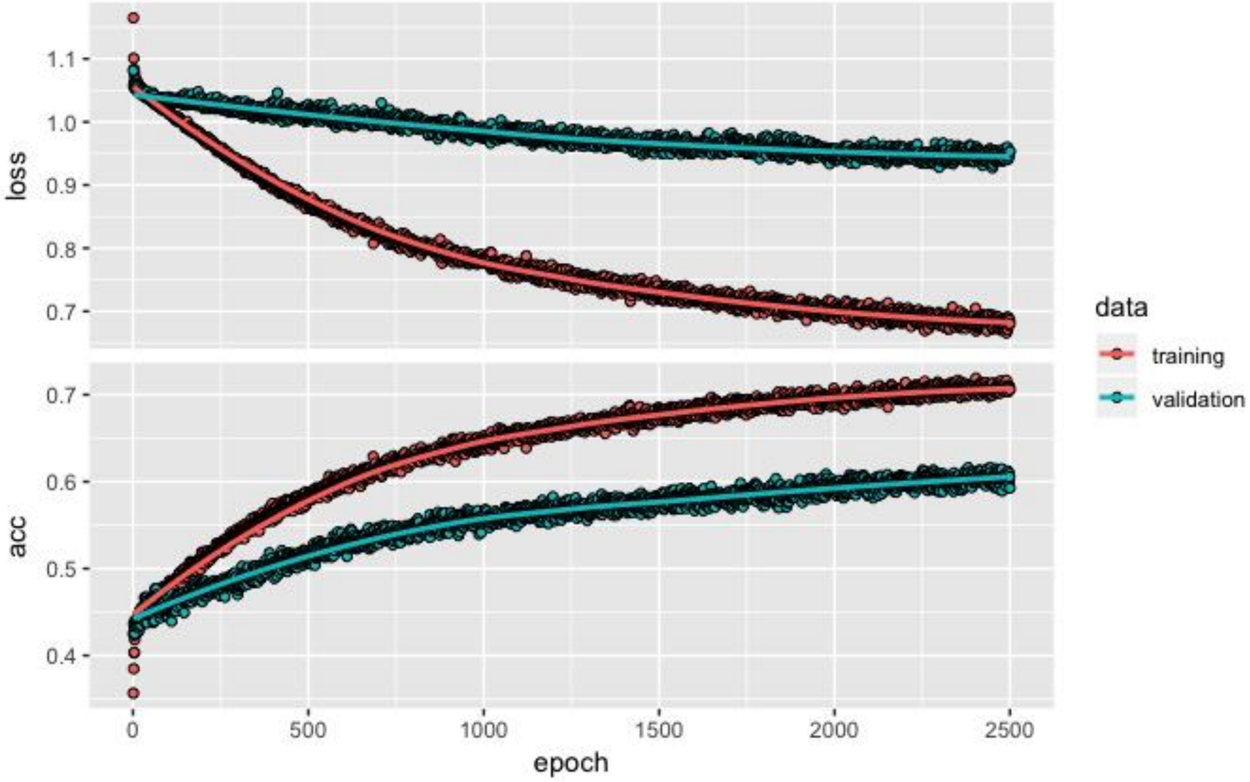

Figure: Resulting validation and training accuracy and loss on neural network trained with all features

Based on the image above plotted using the history returned by the fit method used on the model constructed above, it is clear that the model performs well as the model's training loss and validation loss are decreasing and the validation and training accuracies are increasing. If the model runs for longer time by increasing the number of epochs to, say, 4000 from 2500, the model could have performed better, and the accuracy could have been increased.

Validation loss	0.94
Validation accuracy	0.60
Training loss	0.68
Training accuracy	0.70

Neural network model trained with selected features

Now, the same model is trained using the subset of the data which used only the selected features based on the RFE algorithm.

Results of deep neural network model based on subset of data

Validation loss	1.042
Validation accuracy	0.46
Training loss	1.021
Training accuracy	0.47

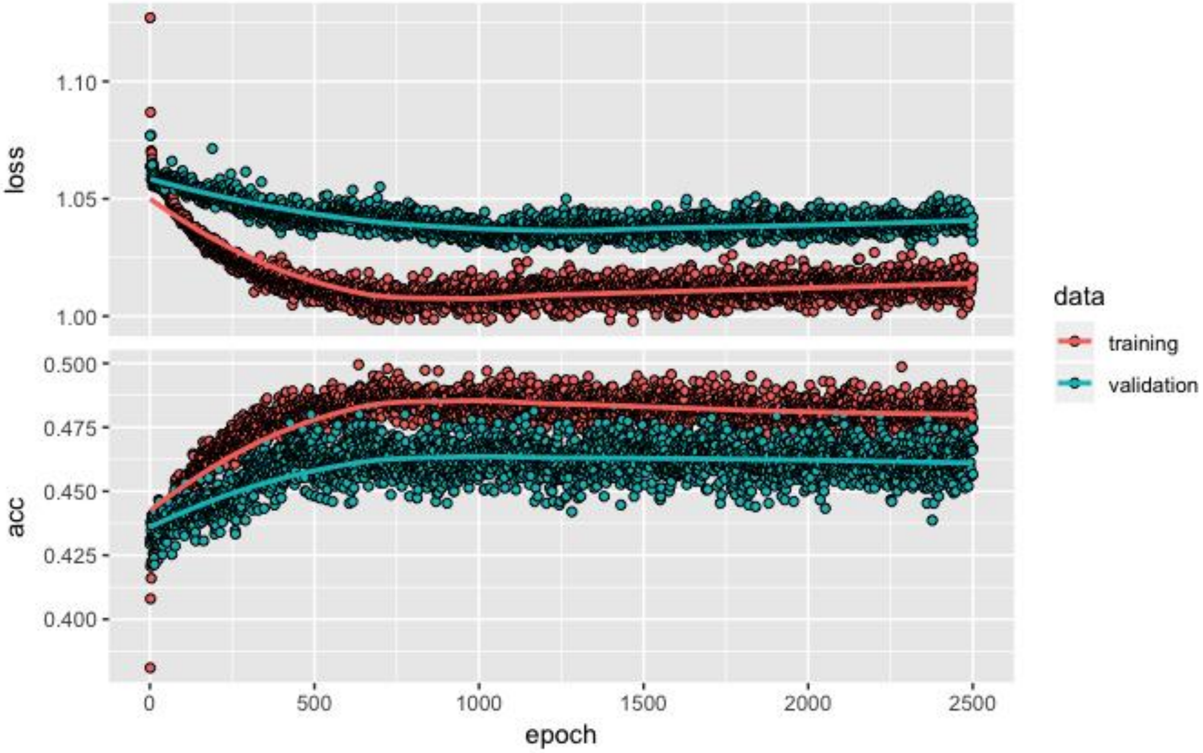

Figure: Resulting validation and training accuracy and loss on neural network trained with selected features

Based on the results of the deep neural network model trained on the subset of the data, it is clear that the model itself selects the features required for it to train well. After a few epochs, the training loss seems to increase and hence, the model is overfitting. The model has to be changed to accommodate the changes in the data it is being trained with and the number of epochs has to be likely reduced to prevent the model from overfitting again.

Evaluation and Deployment

The above discussed models give an overall idea if the patient stays in the hospital or not. If a patient stays in the hospital, the model also predicts how long a patient is likely to stay in the hospital for the treatment. Hence, using the above discussed models, the health care industry can predict the length of stay of patients in their hospitals. Based on the analysis of prediction of the members data, the health care providers can predict the number of beds, the required equipment in their hospitals to treat the patients and appropriate medication as well. If a model predicts that a patient is likely to stay in the hospital, then it is possible that the patient's health could be in a serious condition and based on this prediction, the patient could be diagnosed and then could be treated. This model could also be used by the insurance providers to predict if a member stays in the hospital. The insurance providers would be profitable by investing on the members who are not likely to spend more time in the hospital.

We can measure the value of the model based on its accuracy of predicting the length of stay of a member in the hospital. For a correctly predicted short stay or long stay the model could be awarded some points based on the length of stay. The model should be penalized if it predicts that a member is likely to not stay in the hospital but in reality, the length of stay is long because, the patient is predicted to be healthy based on the prediction, but this is not the case.

The model can be updated, based on the features that decide the stay of the patients in the hospital. If in the new data, there is a feature that helps the model in predicting the stay, then, this feature can be selected along with the other features to train the model. If any other feature with which the model is currently trained becomes useless for the new data, then this feature can be removed, and the model can be trained again.

Conclusion

The models perform well predicting that a patient stays in the hospital, but their accuracy drops as they predict that the patient stays in hospital but actually, they don't. KNN model and the random forest model perform very similarly when trained with all the features discussed in the data preparation section and the five selected features using the Recursive Feature Elimination algorithm. The neural network model however doesn't train well with the selected features and might need some changes in the layers, activation function or dropout or number of epochs for it to perform well and stop overfitting on the training data. Random Forest algorithm with all the features out performs the other two models. The neural network model could perform well if the parameters are tuned and as the accuracy kept increasing gradually and the loss kept decreasing too, the number of epochs can be increased in order to increase the validation accuracy.

References

http://www.scikit-yb.org/en/latest/api/features/rfecv.html

http://topepo.github.io/caret/recursive-feature-elimination.html#rfe

https://en.wikipedia.org/wiki/Confidence_interval

https://en.wikipedia.org/wiki/Sigmoid_function

https://stats.stackexchange.com/questions/82162/cohens-kappa-in-plain-english

https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

https://en.wikipedia.org/wiki/Random_forest

https://machinelearningmastery.com/feature-selection-with-the-caret-r-package/

https://en.wikipedia.org/wiki/Confusion_matrix