June
2017

Readmission of Diabetes Patients – A data analysis project

PROJECT REPORT
BY BRUNDA CHOUTHOY

CSC 478 | DePaul University

Introduction

The objective of this project is to predict the occurrence of a diabetic patient being readmitted. The results will help hospitals identify vulnerable patients with a higher likelihood of readmission. Readmission rates in hospitals are a key indicator on quality of patient care and a clear indication of total cost or inconvenience related to the treatment. Patients with serious medical conditions such as diabetes mellitus are key drivers of readmission rates owing to the complexity of their illness. Therefore, being able to predict based the most important features to determine whether a patient will need readmission can help doctors and hospitals to provide better care.

The goal is to apply the knowledge discovery process on the Diabetes readmission data to discover most important factors contributing to hospitals readmissions and provide an effective prediction model on readmissions and enable hospitals to identify and target patients at the highest risk. The project explores various machine learning techniques on data such classification using Random forest, Decision tree, Naïve Bayes and SVM and clustering with k-means and PCA for reduced dimensionality in clustering. For classification algorithms, Accuracy along with Precision and Recall are used as key performance indicators.

Dataset summary

The dataset was obtained from the UCI Machine Learning Repository. It is listed under the name Diabetes 130 – US Hospitals. According to the dataset description, the data has been prepared to analyze factors related to readmission as well as other outcomes pertaining to patients with diabetes. The dataset represents 10 years (1999-2008) of clinical care at 130 U.S. hospitals and integrated delivery networks.

Data Source: The data was obtained from the Center for Machine Learning and Intelligent Systems at University of California, Irvine.
https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008

Number of Instances: 101766
Number of attributes: 50

Encounters – Record data: Each row in the data represents an encounter and information was extracted from the database for encounters that satisfied the following criteria:

(1) It is an inpatient encounter (a hospital admission).
(2) It is a diabetic encounter, that is, one during which any kind of diabetes was entered to the system as a diagnosis.

(3) The length of stay was at least 1 day and at most 14 days.
(4) Laboratory tests were performed during the encounter.
(5) Medications were administered during the encounter.

Features – Attributes: The attributes represent patient and hospital outcomes. This data set mostly contains categorical attributes such as medical specialty, race and gender, but also includes a few ordinal attributes such as age and weight and continues attributes such as time(days) in hospital, number of lab procedures and number of medications.
There is a total of 13 numeric attributes and 37 categorical/nominal attributes in the dataset.

Input variables: The following table lists each feature, type of the attribute and its description:

Feature name	Type	Description and values
Encounter ID	Numeric/ Continuous	Unique identifier of an encounter
Patient Number	Numeric/Continuous	Unique identifier of a patient
Race	Nominal/Categorical	Values: Caucasian, Asian, African American, Hispanic, and other
Gender	Nominal/Categorical	Values: male, female, and unknown/invalid
Age	Nominal/Categorical	Grouped in 10-year intervals: 0, 10), 10, 20), …, 90, 100)
Weight	Numeric/Continuous	Weight in pounds.
Admission Type	Nominal/Categorical	Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available
Discharge disposition	Nominal/Categorical	Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available
Admission Source	Nominal/Categorical	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital
Time in hospital	Numeric/Continuous	Integer number of days between admission and discharge
Payer code	Nominal/Categorical	Integer identifier corresponding to 23 distinct values, for example, Blue Cross/Blue Shield, Medicare, and self-pay

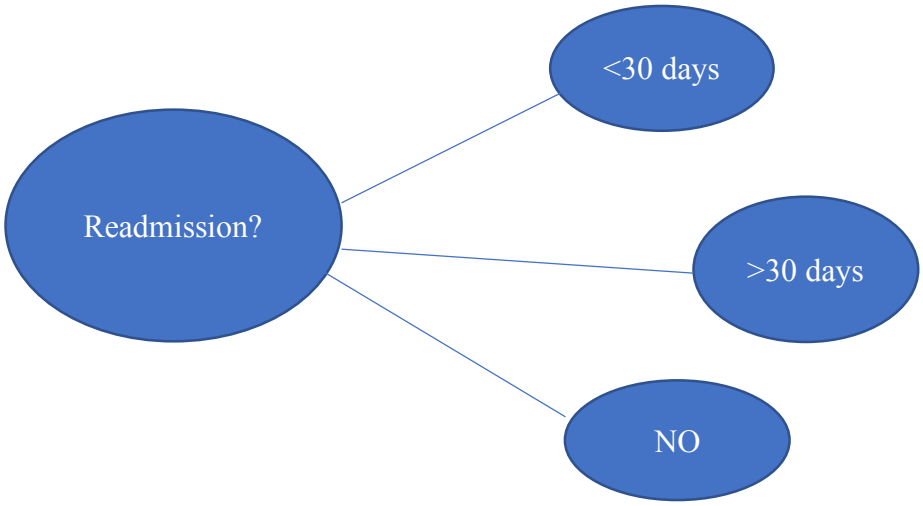
Medical specialty	Nominal/Categorical	Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon
Number of lab procedures	Numeric/Continuous	Number of lab tests performed during the encounter
Number of procedures	Numeric/Continuous	Number of procedures (other than lab tests) performed during the encounter
Number of medications	Numeric/Continuous	Number of distinct generic names administered during the encounter
Number of outpatient visits	Numeric/Continuous	Number of outpatient visits of the patient in the year preceding the encounter
Number of emergency visits	Numeric/Continuous	Number of emergency visits of the patient in the year preceding the encounter
Number of inpatient visits	Numeric/Continuous	Number of inpatient visits of the patient in the year preceding the encounter
Diagnosis 1	Nominal/Categorical	The primary diagnosis (coded as first three digits of ICD9); 848 distinct values
Diagnosis 2	Nominal/Categorical	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values
Diagnosis 3	Nominal/Categorical	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values
Number of diagnoses	Numeric/Continuous	Number of diagnoses entered to the system
Glucose serum test result	Nominal/Categorical	Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured

A1c test result	Nominal/Categorical	Indicates the range of the result or if the test was not taken. Values: ">8" if the result was greater than 8%, ">7" if the result was greater than 7% but less than 8%, "normal" if the result was less than 7%, and "none" if not measured.
Change of medications	Nominal/Categorical	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: "change" and "no change"
Diabetes medications	Nominal/Categorical	Indicates if there was any diabetic medication prescribed. Values: "yes" and "no"
24 features for medications	Nominal/Categorical	For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: "up" if the dosage was increased during the encounter, "down" if the dosage was decreased, "steady" if the dosage did not change, and "no" if the drug was not prescribed

Target variable:

Readmitted	Nominal/Categorical	Days to inpatient readmission. Values: "<30" if the patient was readmitted in less than 30 days, ">30" if the patient was readmitted in more than 30 days, and "No" for no record of readmission.

Readmitted is the class variable and the goal is to predict the occurrence of a diabetic patient being readmitted within 30 days.



Tools used for analysis:
IPython, Jupyter notebook, matplotlib, NumPy, scikit-learn, Pandas

Data Cleaning and final dataset:

Preliminary analysis and preprocessing of the data were performed on the data resulting in retaining only these features and encounters that could be used in further analysis, that is, contain sufficient information.

Irrelevant data:
As mentioned in the research article linked with this dataset, the preliminary dataset contained multiple inpatient visits for the same patient – there is more than one encounter for many of the patients and the observations could not be considered as statistically independent. Thus, only one patient per encounter is used for further analysis. The first encounter for each patient as the primary admission is considered and determined whether they were readmitted within 30 days.

Furthermore, the attribute, discharge disposition id, corresponds to 29 distinct values that indicate patients are discharged to home or another hospital, to hospice for terminally ill patients, or indicate that the patients have passed away. To correctly include only patients who are alive and not in hospice, the records that had discharge disposition codes of 11, 13, 14, 19, 20, and 21were removed, to avoid biasing the analysis. These discharge codes matched the instances of patients who were deceased or sent to hospice.

Dataset reduced to size: 69973 instances, 50 features

Missing value treatment:
The dataset contains incomplete, redundant and noisy information as expected in any real-world data. There are many features that could not be treated directly since they have a high percentage of missing values.

The following table lists the features with the number and percentage of missing values:

Feature name	Number of missing values	Percentage of missing values
Weight	67185	0.96 or 96%
Payer code	30415	0.56 or 56%
Race	1918	0.027 or 2.7%
daig_1	10	0.00014 or 0.0014%
diag_2	293	0.0041 or 0.041%
diag_2	1224	0.017 or 1.7%
Medical specialty	33639	0.48 or 48%

There were several features that could not be treated directly since they had a high percentage of missing values. These features were weight (96% values missing), payer code (56%), and medical specialty (48%). Weight attribute is considered to be too sparse and it was not included in further analysis. Payer code was removed since it had a high percentage of missing values and it was not considered relevant to the outcome. Medical specialty attribute was maintained, adding the value "missing" to account for missing values. Race attribute was maintained as well, adding the value "missing" to account for missing values. Furthermore, attributes diag_1, diag_2, diag_2 were treated by removing the instances containing NAs for further analysis.

There were also four features: 'acetohexamide', 'examide', 'citoglipton', 'glimepiride-pioglitazonethat' that have only one unique value for all instances in the input data. All the instances contain the value 'NO' which does not contribute to the outcome of the target variable and hence were dropped for further analysis.

'Patient_nbr' and 'encounter_id' are two features that represent the key – and every instance represents one encounter per patient and are all unique values. These 2 attributes were removed as they will not contribute to the outcome of the target variable as well.

Features dropped for analysis:
Weight, Payer code, acetohexamide, examide, citoglipton, glimepiride-pioglitazonethat, Patient_nbr, encounter_id

Dataset reduced to size: 68689 instances, 42 features

Examining the target variable:

Since we are primarily interested in factors that lead to early readmission, data sets with two different sizes and classes were considered for further analysis.

(1) **Dataset A:** Pseudo 2 classes for the target variable - the readmission attribute (outcome) is defined as having two values: "<30," if the patient was readmitted within 30 days of discharge or "NO," which covers both readmission after 30 days and no readmission at all.
(2) **Dataset B:** 2 classes for the target variable - the readmission attribute (outcome) is defined as having two values: "<30," if the patient was readmitted within 30 days of discharge or "NO," which includes patients who were NOT readmitted at all. Patients who were readmitted after 30 days were dropped for further analysis.

The class – '>30' i.e. patients readmitted after 30 days of discharge appeared to be a little fuzzy and noisy, so just to make sure the data is separable and consists of two different classes – both the datasets were used for the knowledge discovery process.

	Number of instances/samples	**Number of features**
Initial dataset – Preliminary data	101766	49 Input + 1 target <u>Target class distribution:</u> Class 'NO': 40534 Class '>30': 21944 Class '<30': 6211
Final dataset A	68689	41 Input + 1 target <u>Target class distribution:</u> Class 'NO': 62478 Class '<30': 6211
Final dataset B	46745	41 Input + 1 target <u>Target class distribution:</u> Class 'NO': 40534 Class '<30': 6211

As shown above, the dataset is highly imbalanced – only 9% of the data contain the class: '<30'.

Data Exploration and visualization:

Following the data cleaning process, the next stage is to the exploratory analysis of the data that is performed on the Final data set A i.e. with 68689 instances and 42 features. Data exploration is an important step in the process of knowledge discovery as there is an extensive need to find the spread of the variables, if the variables are correlated or if outliers are present in the data before we proceed to apply machine learning techniques and to avoid error in prediction.

Univariate analysis:
All the variables were examined individually. In case of continuous variables, to understand the central tendency and spread of the variable – Histograms, boxplots are used as shown in figure 1 and figure 2. For categorical variables, frequency tables and Bar charts were used for exploration.

The histogram results from Figure 1 are interesting because many machine learning techniques assume a Gaussian univariate distribution on the input variables. Some attributes like number_lab_procedures, number_inpatient, time_in hospital may have a Gaussian or nearly Gaussian skewed distribution and others like num_procedures may have a bimodal distribution.

Density plots are another way of getting a quick idea of the distribution of each attribute and are displayed in Figure 2. The distribution of each attribute is clearer than histograms.

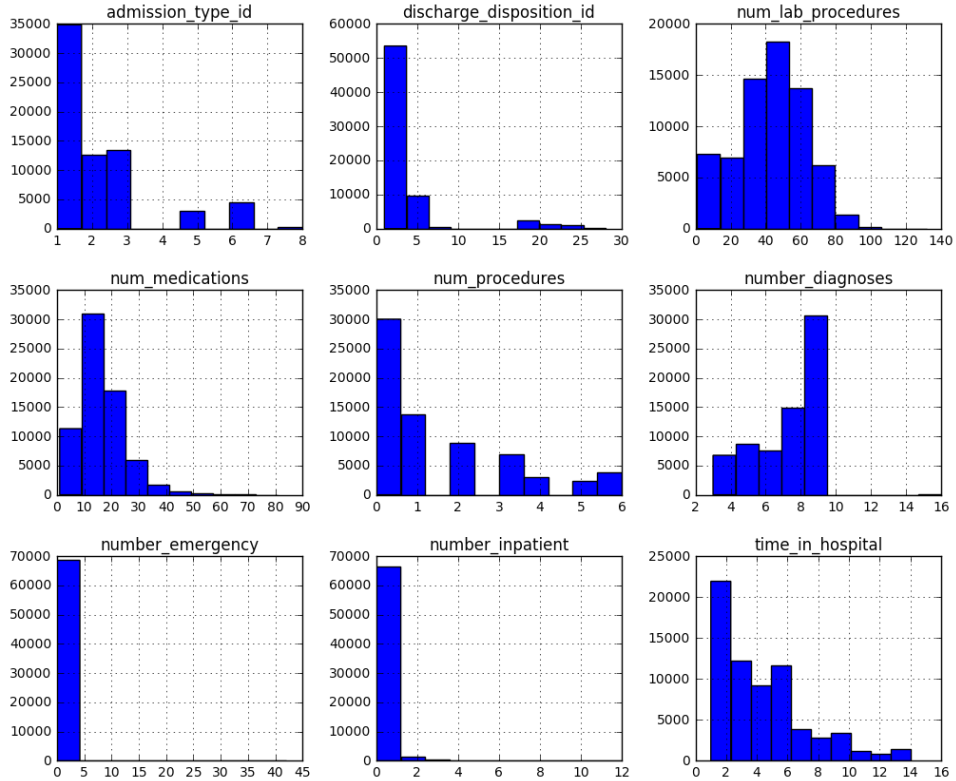


Figure 1: Univariate Histograms for continuous data

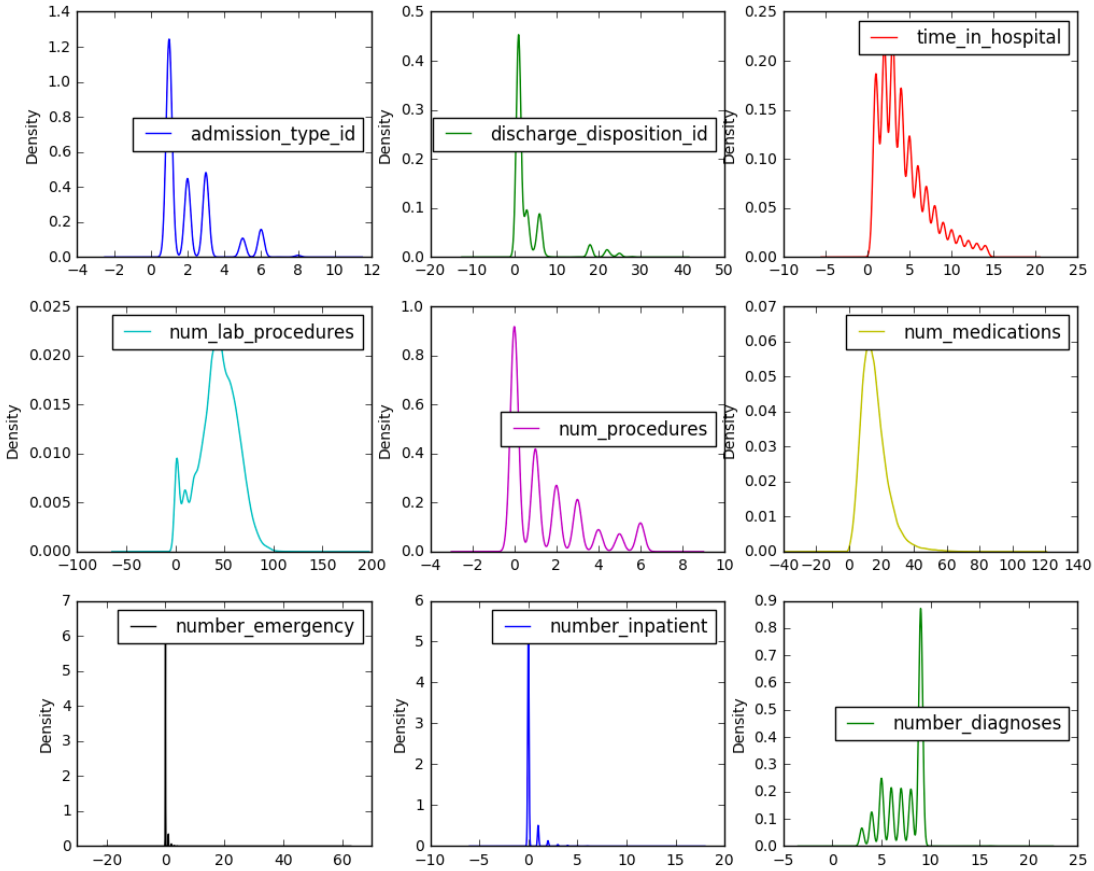


Figure 2:Univariate Density plots

Box and whisker plots were used to understand the spread of the variable and detect outliers and are displayed in Figure 3. The spread of attributes is quite different. For time in hospital, it can be observed that most patients within 1-6 days and very few patients have stayed in the hospital for longer than 12 days. Some attributes like number_inpatient and number_emergency are quite skewed with smaller values.

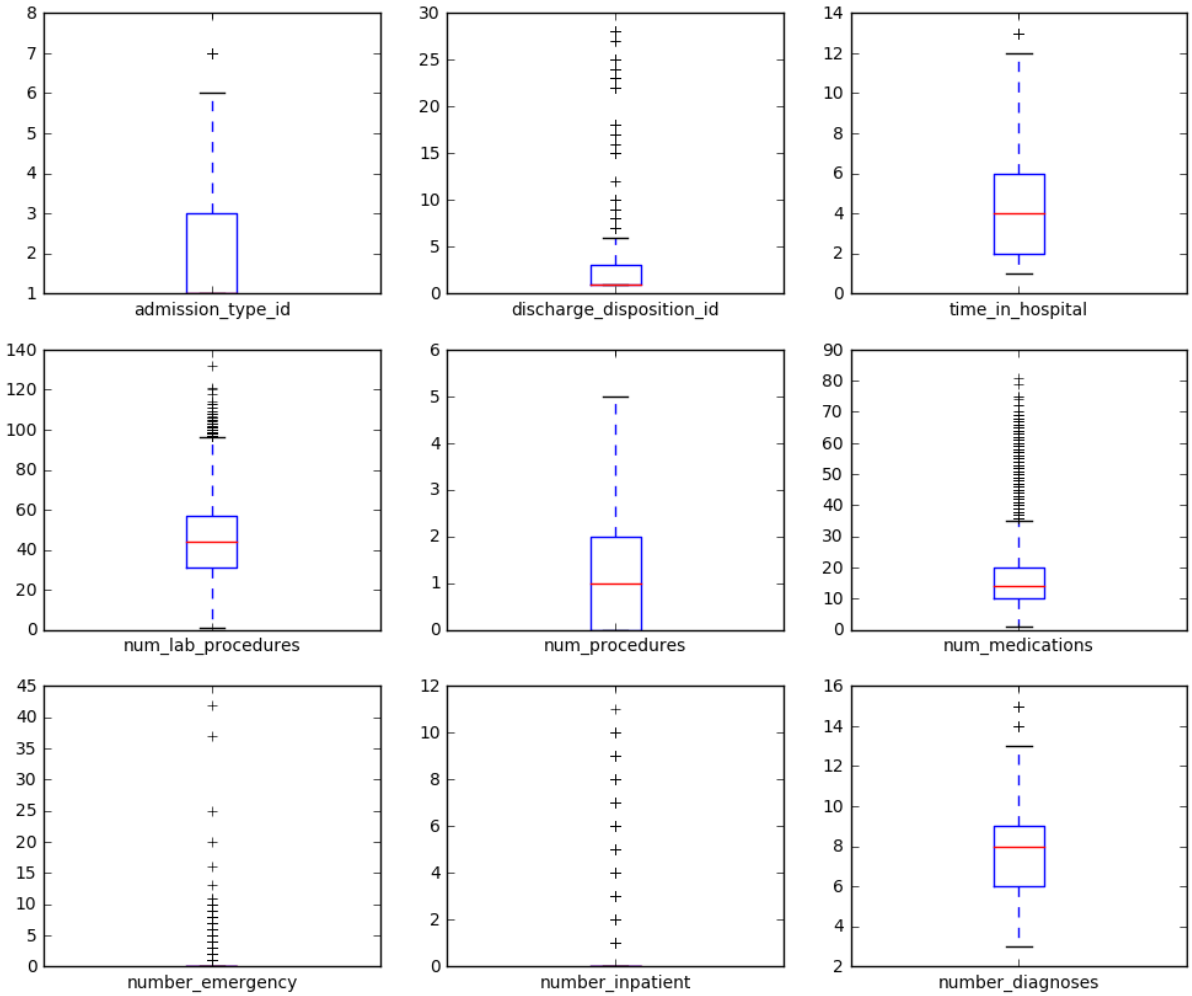


Figure 3:Univariate box and whisker plots

Figure 4 represents Bar plots for categorical data. It can be observed that Caucasians dominate the race attribute and there are more female patients. Since this is a diabetes encounter, most of the patients already have a diabetes medicine assigned and it's interesting to see and A1Cresult for more than 55000 patients is not available i.e. the test is not conducted.

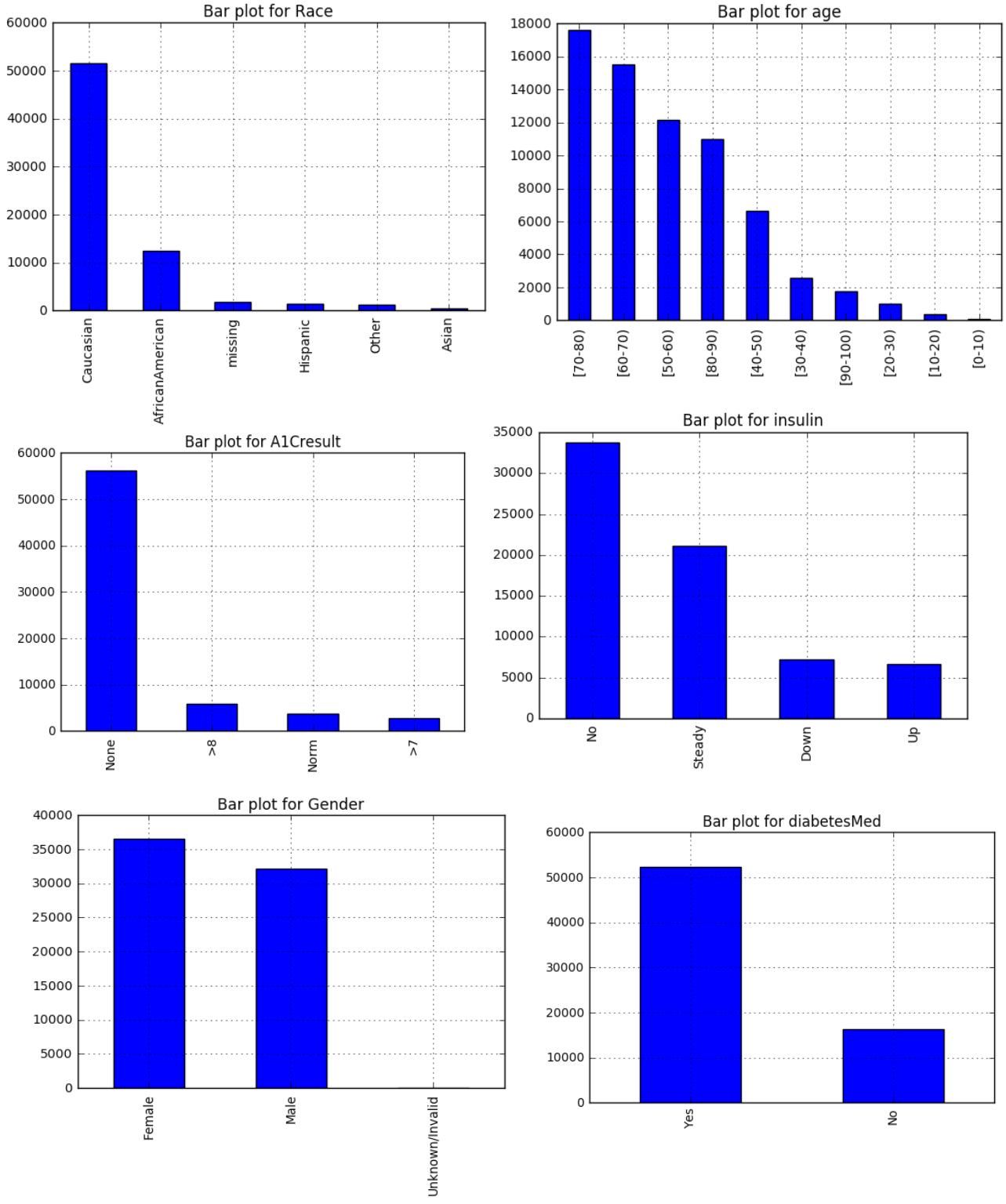


Figure 4:Bar plots for categorical data

Multivariate analysis:

Correlation matrix is used to calculate the correlation between each pair of attributes and the plot is displayed in Figure 5. It can be observed that no two variables are highly correlated with each other. This is useful to know, because some machine learning algorithms like logistic regression can have poor performance if there are highly correlated input variables in your data.

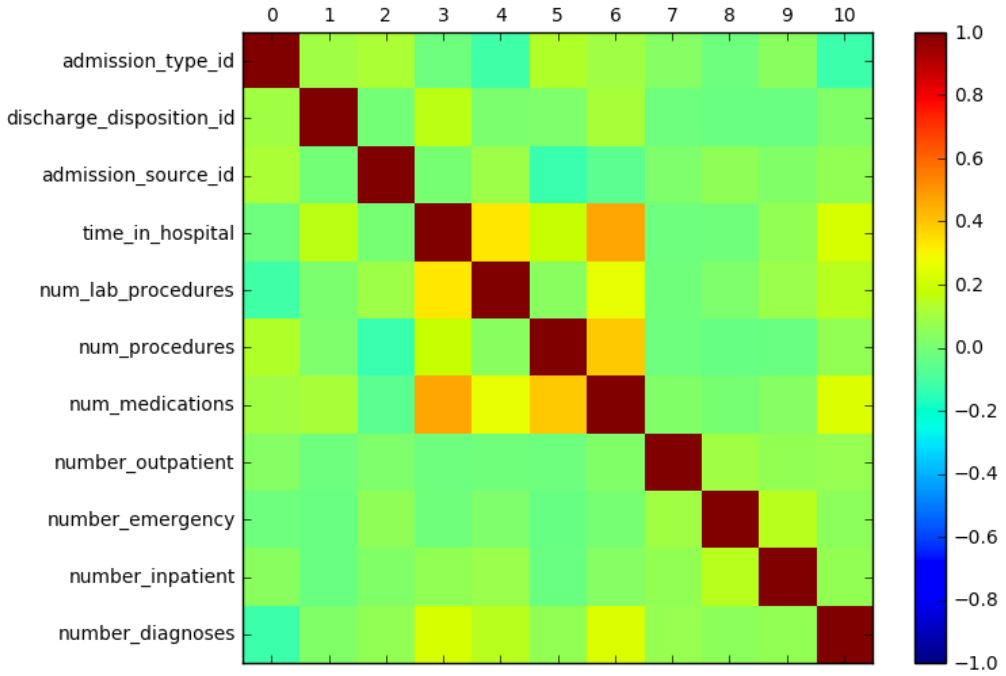


Figure 5: Correlation matrix plot

Scatter plots for bivariate analysis:

Scatter plot matrix is also created to examine the relationship between two variables from different perspectives. Figure 6 depicts the scatter plot matrix. It can further be seen that no two variables are highly correlated.

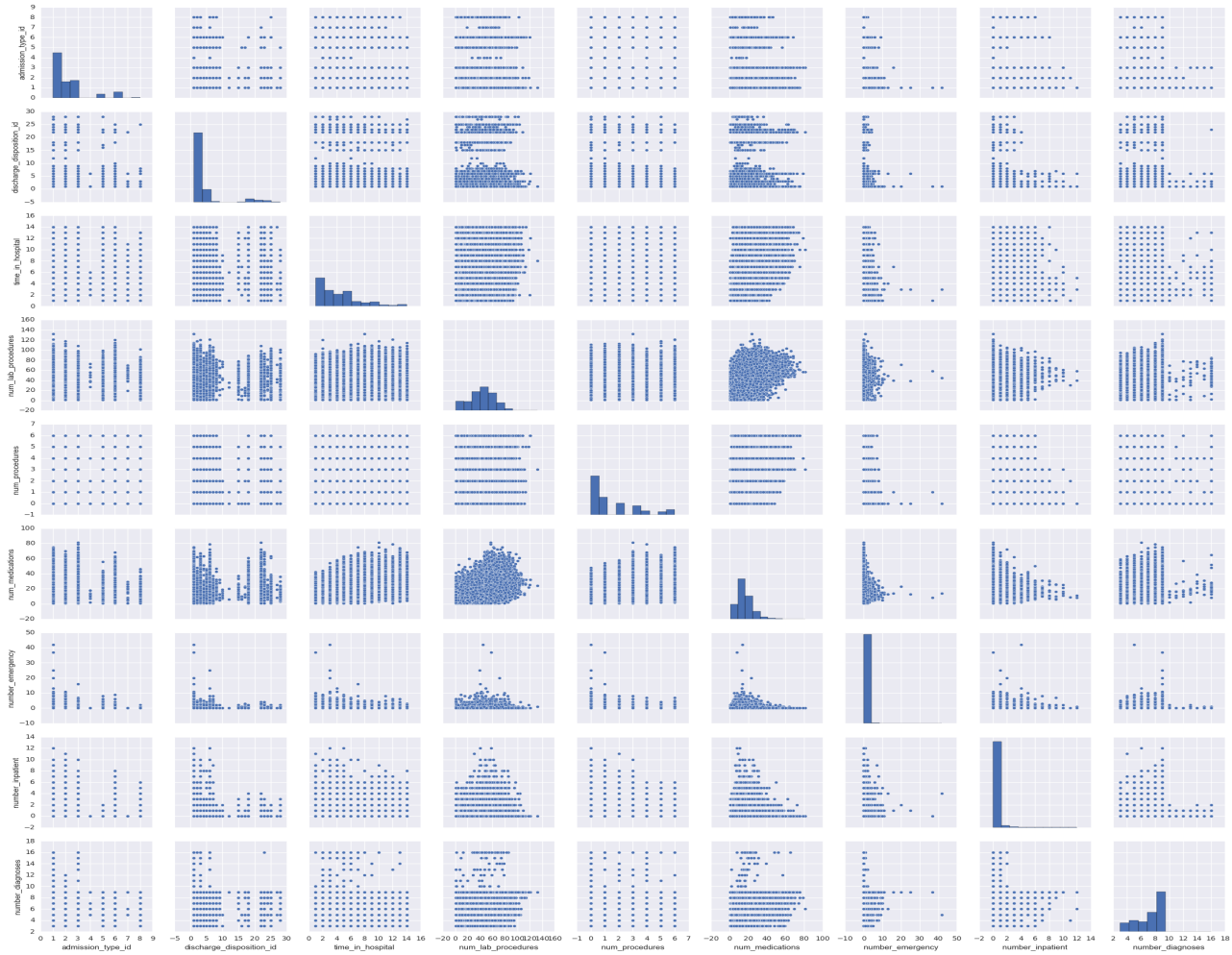


Figure 6: Scatter plot matrix for continuous variables

Data preparation for Machine learning algorithms:

Label encoder from the sklearn preprocessing module is used to convert categorical to nominal categorical variables for further processing and preparing input for machine learning algorithms.

For the application of machine learning algorithms such as Support vector machines, the encoded data is z-score normalized using the scale method from the sklearn preprocessing module. SVMs assume that the data it works with is in a standard range, usually either 0 to 1, or -1 to 1. So, the normalization of feature vectors prior to feeding them to the SVM is very important. Also, to make sure that for each dimension, the values are scaled to lie roughly within this range.

Different feature selection/reduction approaches such as Recursive feature elimination and PCA are used for each of the techniques(models) applied on data and is described in the Experimental results section.

Experimental Results:

As mentioned, datasets of 2 different sizes (dataset A and dataset B) were used to build, evaluate and compare different machine learning models.

Two different Machine learning techniques were applied on each of the datasets:

1. **Classification**
   a. Decision tree
   b. Random forest – Ensemble method
   c. Support Vector Machine
   d. Naive Bayes

2. **Clustering**
   a. K-Means clustering

1a. Classification – Decision Tree Classifier

Decision tree classifier from the sklearn tree package is used for two main purposes –

i) Feature selection - Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables. Feature importance generated by the model are very useful for further analysis.
ii) Model evaluation - to build the model to compare and learn the key performance indicators

Decision tree classifier was applied on the cleaned dataset with 41 input features and feature importance and ranking were obtained. Further, the best 20 and 10 features were selected by setting a threshold value to choose the best features for model building and performance evaluation.

Preliminary parameters used by the decision tree model for feature selection:
criterion: 'Entropy'
class_weight='balanced'
max_depth=10
min_samples_leaf=10

Feature selection results:

	# of features	Accuracy (in %)	Precision (in %)	Recall (in %)
Dataset A (68K features)	41 features	64.6	88	65
	20 features	63.8	88	64
	11 features	62.1	88	62
Dataset B (40k features)	41 features	65	84	65
	20 features	65.2	84	65
	11 features	64.2	83	64

As per the results table above, the dataset with 20 features were considered as optimal to be used for further analysis.

Grid search with Cross validation was performed to choose the best parameters for the classifier model. Parameters chosen by GridSearchCV method from sklearn grid_search module:
({'class_weight': 'balanced',
'criterion': 'gini',
'max_depth': 20,
'min_samples_leaf': 5,
'min_samples_split': 10},
0.6332655899026634)

Model Performance based on the best parameters selected by Grid search function:

20 features	Accuracy (in %)	Precision (in %)	Recall (in %)
Dataset A (68K features)	78.3	93	78
Dataset B (40k features)	78.7	91	79

1b. Classification – Random Forest Classifier

Random Forest is a versatile machine learning ensemble method that undertakes dimensionality reduction methods, handling missing and outlier values as well.

Random forest classifier from the sklearn is used on this data for:

i. Feature selection: One of best benefits of Random forest is the power to handle large data set with higher dimensionality. It can identify the most important variables so it is

also considered as one of the **dimensionality reduction** methods. Further, the model outputs Importance of variable, which is a very handy feature.

ii. It has methods for balancing errors in data sets where classes are **imbalanced**.

iii. Random Forest involves sampling of the input data with replacement called as **bootstrap sampling**. Error estimated on these out of bag samples is known as out of bag error. Study of error estimates by Out of bag, gives evidence to show that the out-of-bag estimate is as accurate as using a test set of the same size as the training set. Therefore, using the out-of-bag error estimate removes the need for a set aside test set.

Random forest classifier from the sklearn ensemble package is used for model building and evaluation. The classifier was applied on the cleaned dataset with 41 input features and feature importance and ranking were obtained. Further, the best 20 and 10 features were selected by setting a threshold value to choose the best features for model building and performance evaluation.

Preliminary parameters used:
n_estimators=500
criterion='entropy'
class_weight='balanced_subsample'
max_depth=10
min_samples_leaf=10

Feature Selection results:

	# of features	**Accuracy (in %)**	**Precision (in %)**	**Recall (in %)**
Dataset A (68K features)	41 features	70.3	88	70
	20 features	71.5	88	72
	11 features	71.2	88	71
Dataset B (40k features)	41 features	71.3	84	71
	20 features	72.3	85	72
	11 features	72.2	84	76

As per the results table above, the dataset with 20 features were considered as optimal to be used for further analysis. The feature importance graph for the 20 features is shown in Figure 6.

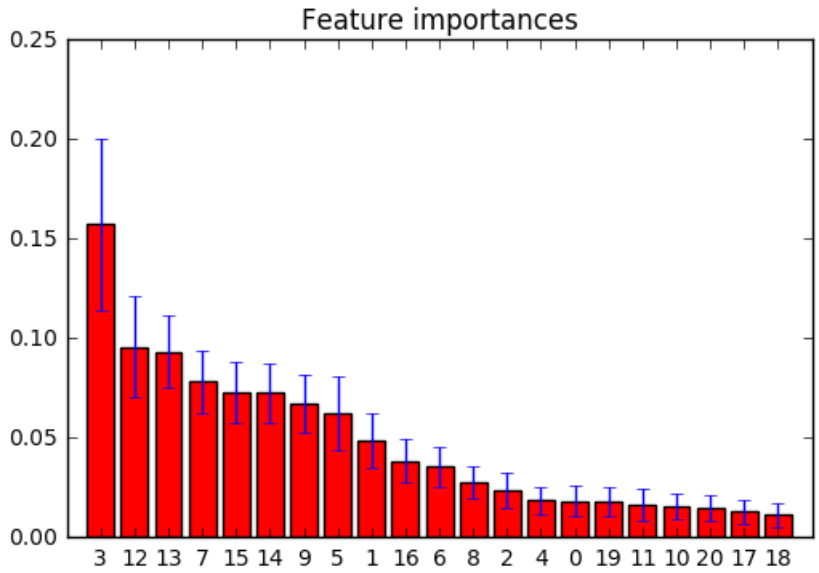


Figure 7: Feature importance Graph for 20 best attributes

Number_inpatient, discharge_disposition_id, diag_1, num_lab_procedures, num_medications are identified as most important features in the prediction of readmissions.

Grid search with Cross validation was performed to choose the best parameters for the Random forest classifier model. Parameters chosen by GridSearchCV method from sklearn grid_search module:
({n_estimators=1000,
criterion='gini',
class_weight='balanced_subsample',
max_depth=20,
min_samples_leaf=5,
min_samples_split=10},
0.6332655899026634)

Model Performance based on the best parameters selected by Grid search CV method:

20 features	**Accuracy (in %)**	**Precision (in %)**	**Recall (in %)**
Dataset A (68K features)	98.7	99	99
Dataset B (40k features)	96.8	97	97

1c. Classification – Support vector machine (SVM)

SVM works well with clear margin of separation and is effective in high dimensional spaces. It uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.

i. Input data to SVM is normalized z-score values, the encoded data is obtained using the scale method from the sklearn preprocessing module. SVMs assume that the data it works with is in a standard range, usually either 0 to 1, or -1 to 1. So, the normalization of feature vectors prior to feeding them to the SVM is very important. Also, to make sure that for each dimension, the values are scaled to lie roughly within this range.
ii. It works well with imbalanced input data, by setting the training weight factors for the positive and negative classes in inverse proportion using the class_weight parameter, so that the trained model is not more sensitive to the class for which you have more samples.
iii. Feature selection was performed using Recursive feature elimination technique from the preprocessing module in sklearn and SVC was used as the base classifier to select the 20 best features for the model.
iv. Train and test data sets were obtained using the sklearn cross validation module and the split ratio was set to 80:20.

The following table summarizes the result on test data before and after feature selection:

TEST DATA	# of features	Accuracy (in %)	Precision (in %)	Recall (in %)
Dataset A (68K features)	41 features	71.6	86	72
	20 features	72.5	88	72
Dataset B (40k features)	41 features	68.8	81	69
	20 features	69.3	81	69

As per the results table above, SVM does provide some interesting values for precision and overall accuracy but classifiers such as Decision tree and Random forest did work better for this imbalanced feature set.

1d. Classification – Naïve Bayes

Naïve Bayes algorithm is simple and effective and a good choice for any classification problem.

i. Naive Bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method.
ii. Naive Bayes for multinomial models in the sklearn module- MultinomialNB is used as the classifier. The multinomial Naive Bayes classifier is suitable for classification with discrete features. The multinomial distribution normally requires integer feature counts.
iii. Feature selection was performed using Recursive feature elimination technique from the preprocessing module in sklearn and MultinomialNB was used as the base classifier to select the 20 best features for the model.
iv. Train and test data sets were obtained using the sklearn cross validation module and the split ratio was set to 80:20.

The following table summarizes the result on test data before and after feature selection:

TEST DATA	# of features	Accuracy (in %)	Precision (in %)	Recall (in %)
Dataset A (68K features)	41 features	75.3	85	75
	20 features	90.6	85	91
Dataset B (40k features)	41 features	66.3	79	66
	20 features	86.1	81	82

As per the table above, Naïve Bayes gave some interesting results but based on the class probabilities for Precision and recall – the model didn't perform well with imbalanced data.

Classification report and confusion matrix for the 20 features with Dataset A is shown below:

```
Accuracy:0.906
Classification report
             precision    recall  f1-score   support

        <30       0.24      0.02      0.04      1224
         NO       0.91      0.99      0.95     12514

avg / total       0.85      0.91      0.87     13738

Confusion matrix
[[   30  1194]
 [   96 12418]]
```

As per the classification report, we can see that the Precision and recall values for the class '<30' is very low and there are 1194 false positives. Hence, it can be concluded the Naïve Bayes model did not perform well with imbalanced data.

2a. Clustering: k-Means and PCA for reduced dimensionality in clustering

Clustering techniques such as kmeans was used to gain more knowledge about the data and further learn the different classes and clusters.

Data preparation: For the application of kmeans and PCA, the input data was normalized using the MinMaxScaler function from the sklearn preprocessing module. All features were normalized to the value between 0 and 1 so that they are in the same range.

Principal component analysis was performed on the normalized input data using PCA method from the sklearn decomposition module. 20 components were used to capture 95.95% of the data and hence the data was reduced from 41 features to 20 principal components which is not very ideal. Principal components and the percentage of variance captured by each component is shown in Figure 8.

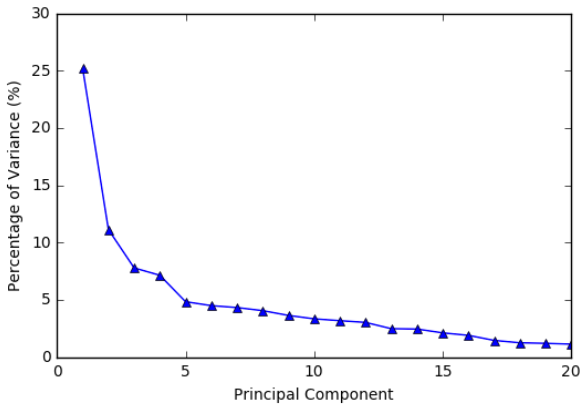


Figure 8: % of variance captured by each component

Given that the variation in the data is not a lot and there is not a good amount of correlation between features, the results from PCA were not great.

Kmeans clustering was performed before and after applying a dimensionality reduction technique such as PCA. The results are summarized in the table below:

kmeans	# of Features used	Completeness	Homogeneity
Dataset A (68K features)	41 features	0.000160165	0.00036319

	PCA reduced components (20)	0.000160165	0.00036319
Dataset B (40K features)	41 features	0.000493588	0.000864397
	PCA reduced components (20)	0.000493588	0.000864397

As per the results from the table, we can see that the completeness and homogeneity scores are very low and k-means clustering did not perform well on this dataset. It can also be concluded that PCA dimensionality reduction did not contribute to improving the performance of kmeans clustering.

To summarize, the feature set does not appear ideal for clustering or dimensionality reduction using PCA – possibly due to the imbalanced nature of the data for the response variable and predictor variables as well.

Experimental Analysis

Based on the data analysis and experimentation, the best model selected for this dataset is using the Random forest classifier. Table 1 shows the classification report and confusion matrix for the random forest classifier with 20 best features:

Accuracy:0.987

Classification report

	precision	recall	f1-score	support
<30	0.95	0.91	0.93	6211
NO	0.99	0.99	0.99	62478
avg / total	**0.99**	**0.99**	**0.99**	68689

Confusion matrix

```
[[ 5632   579]
 [ 327    62151]]
```

Table 1: Random forest model with 20 important features

- Individual classification algorithms like Decision tree, SVM and Naïve Bayes were applied to conduct a performance assessment with an ensemble classification technique like Random Forest.
- An ensemble method – Random forest provided best results for this imbalanced data compared to other individual classification algorithms.
- A subset of 20 most important features were obtained that are especially significant in determining the outcome of target variable. Seems that these 20 features can help predict the outcome of readmission within 30 days and the results will be better than random guess.
- Number_inpatient, discharge_disposition_id, diag_1, num_lab_procedures, num_medications are identified as most important features in the prediction of readmissions.
- From the selected best model, the proportion of actual positive cases which are correctly identified i.e. recall or sensitivity is 99%, precision is 99% and the proportion of the percentage of instances correctly classified and total number of instances i.e. accuracy is 98.7%.
- Results from this data analysis concludes in the direction of successfully predicting whether a patient will be readmitted with 30 days of discharge or not.

Conclusion

- The rate of hospital readmissions of patients is a key measure that is tracked for numerous reasons. The goal of this project was to apply the knowledge discovery process on the Diabetes readmission data to discover most important factors contributing to hospitals readmissions and provide an effective prediction model on readmissions to enable hospitals to identify and target patients at the highest risk.
- The project explored various machine learning techniques on data such classification using Random forest, Decision tree, Naïve Bayes and SVM and clustering with k-means and PCA for reduced dimensionality in clustering.
- To make sure the classes of the target variable are separable into distinct classes – 2 datasets of different sizes were used for analysis. Since the performance of the models were almost similar – dataset A with 68689 instances could be used further.
- Random forest classifier performed with the best values for accuracy, precision and recall and the best model selected has an accuracy of 98.7%.
- The feature set does not appear ideal for clustering or dimensionality reduction using PCA – possibly due to the imbalanced nature of the data for the response variable and predictor variables as well.
- The readmission groups are most significantly related to number of inpatient visits, discharge disposition id, diag_1 i.e. the first diagnosis of the patient, number of lab procedures, number of medications.

- The readmission of a patient does not solely depend on any single variable, but the interactions of related variables.
- Instead of tracking all the 50 features, hospitals can be to focus on the 20 features identified as significant. Future work will be to identify the right threshold and values for each of these features that would indicate the patient would not be readmitted within 30 days.

References

- Beata Strack et al., "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records." – https://www.hindawi.com/journals/bmri/2014/781670/

- Machine Learning Repository - https://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+1999-2008#