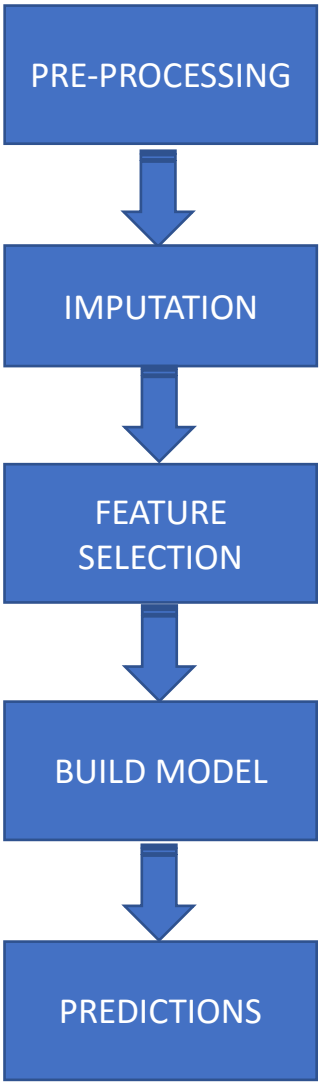
Data Mining Classification
Diabetic Patients Readmission Prediction

Report By:

1. **Abhinav Maurya**
2. **Nandini Malempati**
3. **Samarth Raghuram Shetty**
4. **Ushang Thakker**
5. **Vikas Janardhanan**

Under guidance of:
Prof. Yijun Zhao

Data Flow Transformation:



Preprocessing Steps:

Label Encoding:

- Features (diag_1, diag_2, diag_3) which have predefined meaning for each string entry in the record are replaced with the appropriate Integer values as per the data set documentation provided.
- Rest of the features of string data type are label encoded to convert them into integer values.
- Range values are converted to mean of their boundaries.

Normalization:

- All the features are Z-score normalized.
- This is required in case of SVM as we are trying to minimize the distance between the separating plane and the support vectors. If one feature (i.e. one dimension in this space) has very large values, it will dominate the other features when calculating the distance. If you rescale all features (e.g. to [0, 1]), they all have the same influence on the distance metric.

Imputation

- We tried predicting the missing values using KNN method. The imputation cost was higher with negligible increase in accuracy. So we decided to proceed with mean, median, mode strategy.
- Continuous values: We are computing the mean of each feature and filling the missing values with the same.
- Categorical values: We are computing the mode of each feature and filling the missing values with the same.

Features selection

- The features with more than 50% threshold of missing values are eliminated because the prediction would result in incorrect correlation with the label. Weight, payer code and Medical specialty are eliminated on this basis.
- encounter_id and patient_nbr were also eliminated as there was no correlation with the label.

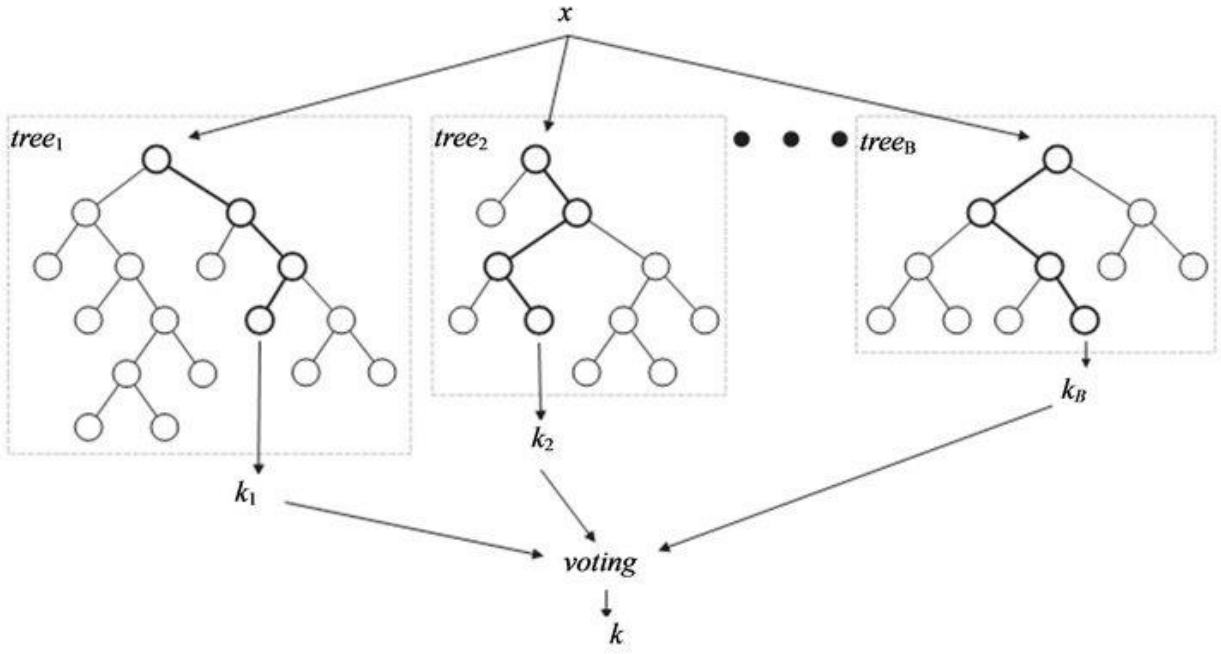
Data Mining Techniques

Handling imbalanced data:

We found that the given dataset is imbalance. Number of instances of class '<30' was just 10% of the entire dataset.
Techniques tried to fit the data:

- Under sampling: After undersampling we found the accuracy to be 73%.
- Oversampling: After oversampling the accuracy increased to about 93%. So, we are using oversampling of the data.

Random forest Classifier:



Ensemble Sampling method: Bootstrapping

The random forest classifier implements the Bootstrap aggregating, also called bagging.
Given a standard training set X of size n, bagging generates m new training sets each of size n', by sampling from X uniformly and with replacement. By sampling with replacement, some observations may be repeated in each bags(Xi) If n'=n, then for large n the set Xi is expected to have the fraction (1 - 1/e) (≈63.2%) of the unique examples of X, the rest being duplicates.
This kind of sample is known as a bootstrap sample. The m models are fitted using the above K bootstrap samples and combined by voting.

Parameters:
Number of decision tree used to generate the ensemble: 50
Sampling rate: 1.0 (Size of each bootstrapped bag is |N|. These bags are individually used to create decision tree within the random forest)

Experiment results:
Overall It reduces variance and helps to avoid overfitting. This model has high accuracy because the final prediction is done by choosing the majority of vote from 50 different decision trees. Each tree is created by considering different bags and different features within the algorithm. This makes sure that the correct correlation gets more weight over the undesired outliers.

SVM:

We also ran experiments using SVM classifier with linear Kernel.
We observed that SVM Performed poorly in our dataset. The reason could be:

- Use of inappropriate kernel (linear kernel for nonlinear problem)

Decision Tree:

Decision tree performs well for categorical features and not for continuous features which was best suited in our dataset. We can see this improvement when we compare Decision tree accuracy with SVM accuracy. There was a huge improvement.
This better performance of Decision Tree drove us towards the Random Forest.

Statistics:

Given data set is split into training data and test data in (70,30) ratio. Model is built using training data and the same is used for predicting the label in test data. The accuracies are listed below:

Method	Train Accuracy	5-fold CV	Test Accuracy
SVM	~60%	~60%	~60%
Decision Tree	100%	~92%	~93%
Random Forest	100%	~99%	~99%
Ensemble using above methods	100%	~94%	~96%

Output:

```
Train Accuracy: 1.00 [Decision Tree]
Train Accuracy (5 Fold CV): 0.92 (+/- 0.00) [Decision Tree]
Test Accuracy: 0.93 [Decision Tree]
Train Accuracy: 0.60 [Linear SVM]
Train Accuracy (5 Fold CV): 0.60 (+/- 0.00) [Linear SVM]
Test Accuracy: 0.60 [Linear SVM]
Train Accuracy: 1.00 [Random Forest]
Train Accuracy (5 Fold CV): 0.99 (+/- 0.00) [Random Forest]
Test Accuracy: 0.99 [Random Forest]
Train Accuracy: 1.00 [Ensemble]
Train Accuracy (5 Fold CV): 0.95 (+/- 0.00) [Ensemble]
Test Accuracy: 0.96 [Ensemble]
```