**SCHOOL OF APPLIED TECHNOLOGY**

Illinois Institute of Technology

**PROJECT REPORT**

**Group Number-297: Diabetic Medication & Patient Re-admission Prediction using different Classification Algorithms**

UTKARSHA VIDHALE – A20448993

ITMD 525

Topics in Data Management: Data Mining

Table of Contents

[**1. Introduction** 3](#_Toc39853014)

[**2. Data** 3](#_Toc39853015)

[**3. Problems to be Solved** 5](#_Toc39853016)

[**3. KDD** 5](#_Toc39853017)

[4.1. Data Processing 5](#_Toc39853018)

[1. Data Cleaning 5](#_Toc39853019)

[2.Data Integration 7](#_Toc39853020)

[3.Data Transformation 9](#_Toc39853021)

[4.2. Data Splitting 12](#_Toc39853022)

[4.3. Treating Imbalance Data 12](#_Toc39853023)

[A. Predicting Re-admissions 12](#_Toc39853024)

[B. Predicting Diabetic’s Medication Requirement 13](#_Toc39853025)

[4.3. Data Mining Methods and Processes 13](#_Toc39853026)

[A. Predicting Re-admission 13](#_Toc39853027)

[B. Predicting Diabetic’s Medication Requirement 23](#_Toc39853028)

[**5. Evaluations and Results** 33](#_Toc39853029)

[5.1. Evaluation Methods 33](#_Toc39853030)

[5.2. Results and Findings 33](#_Toc39853031)

[A. Predicting Re-admissions 33](#_Toc39853032)

[B. Predicting Diabetic’s Medication Requirement 34](#_Toc39853033)

[**6. Over-fitting concerns** 35](#_Toc39853034)

[**7. Conclusions and Future Work** 37](#_Toc39853035)

[6.1. Conclusions 37](#_Toc39853036)

[6.2. Limitations 37](#_Toc39853037)

[6.3. Potential Improvements or Future Work 37](#_Toc39853038)

[8. References 38](#_Toc39853039)

# **1. Introduction**

The management of sugar level in the hospitalized patient has a significant bearing on outcome, in terms of both morbidity and mortality. This recognition has led to the development of formalized protocols in the intensive care unit (ICU) setting with rigorous glucose targets in many institutions. However, the same cannot be said for most non-ICU inpatient admissions. This analysis of a large clinical database is to be undertaken to examine historical patterns of diabetes care in patients admitted to a US hospital and to inform future directions which might lead to improvements in patient’s medical condition and help to save medical resources and valuable time of medical staff. Our application will provide general idea of patient’s medical necessity based on diabetic prescription and number of times patient is re-admitted in hospital. Since, there has always been shortage of resources in medical industry whether it is hospital bed, medicines and other equipment due to large number of patients, so it is always advisable to keep the tab of patients that might use the resources in future. Our model will provide the predictions in terms of whether patient will be prescribed diabetic medication or not can really help the doctors and medical staff to keep the proper tab of patient’s health and avoid diabetic condition. This model will also determine if patient needs to be readmitted in hospital based on clinical history, medication and other factors which can help the medical staff to properly maintain and utilize medical resources as per the necessity and availability.

# **2. Data**

The dataset represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It is sourced from UCI Machine Learning Repository and has been prepared to analyze factors related to readmission as well as other outcomes pertaining to patients with diabetes. ( <http://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+19992008> ).

The data contains more than 1,00,000 instances and 50 attributes. The dataset is multivariate in terms of its characteristics, whereas the attributes are numerical and nominal.

Attribute list and their datatype (Numerical or Nominal)

encounter\_id : numerical

patient\_nbr : numerical

race : nominal

gender : nominal

age : nominal

weight : nominal

admission\_type\_id : numerical

discharge\_disposition\_id : numerical

admission\_source\_id : numerical

time\_in\_hospital : numerical

payer\_code : nominal

medical\_speciality : nominal

num\_lab\_procedures : numerical

num\_medications : numerical

number\_outpatient : numerical

number\_emergency : numerical

numbert\_inpatient : numerical

diag\_1 : nominal

diag\_2 : nominal

diag\_3 : nominal

number\_diagnoses : numerical

max\_glu\_serum : nominal

A1Cresult : nominal

metformin : nominal

repaglinide : nominal

nateglinide : nominal

chlorpropamide : nominal

glimepiride : nominal

acetohexamide : nominal

glipizide : nominal

glyburide : nominal

tolbutamide : nominal

pioglitazone : nominal

rosiglitazone : nominal

acarbose : nominal

miglitol : nominal

troglitazone : nominal

tolazamide : nominal

examide : nominal

citoglipton : nominal

insulin : nominal

glyburide-metformin : nominal

glipizide-metformin : nominal

glimepiride-metformin : nominal

metformin-rosiglitazone : nominal

metformin-pioglitazone : nominal

change : nominal

diabetesMed : nominal

readmitted : nominal

# **3. Problems to be Solved**

Based on the above dataset, following are some of the interesting research problems:

1. Based on the clinical condition, patient’s physical features and medical history of patient, we are going to predict whether the patient should be treated with diabetic’s medication or not. This can be helpful for both patients and doctors as they can keep the tab of the health of their patients with this prediction model and take some early steps to avoid the condition of diabetic medication.

2. In order to improve patient’s safety and doctor’s valuable time, we are going to determine whether the patient will be required to readmit in the hospital within 30 days, after 30 days or never, based on the clinical history, diabetics prescription and other factors. This can provide the proper timeline of patient’s medical history and will help the clinics and hospitals to utilize proper medical resources based on availability of patient and the critical condition of patient, thus avoiding the chaos.

# **4. KDD**

Knowledge Discovery in Databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Following are the basic steps in KDD which were followed to accomplish the goals.

## 4.1. Data Processing

Basic operations such as the removal of noise if appropriate, collecting the necessary information to model or account for noise, deciding on strategies for handling missing data fields, accounting for time sequence information and known changes which were performed are briefly explained below.

### 1. Data Cleaning

Data cleansing or data cleaning is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. Following are the major steps taken to overcome the dirty data.

#### Unique Values

Two columns named “encounter\_id” and “patient\_number” have unique values for each record mentioned. These columns clearly have no significance in the analysis process and hence were not considered in the further tasks.

#### Single Values

“Examide” and “Citoglipton” consists of same value for all the entries in the dataset. These columns with single value for every possible situation were removed as they do not contribute in analyzing the dataset.

#### Missing Values

Columns such as “Weight”, “Medical\_Speciality”, “Payer\_code” have missing values of 96.9%, 49.1% and 36.6% respectively. Such huge number of null values cannot be replaced therefore, it was finalized to remove these columns.

On the other hand, columns named “Race”, “Diagnosis\_1”, “Diagnosis\_2” and “Diagnosis\_3” had significantly low missing values with 2.2%, 0.02%, 0.35% and 1.39%. These missing values were replaced with the most frequent factor in their respective columns. These were replaced by “Caucasian”, “428”, “276” and “259” respectively.

|  |
| --- |
| Checking Missing Values in Dataset: |
| Filling missing values for “Diagnosis\_1”, “Diagnosis\_2” and “Diagnosis\_3”: |

|  |
| --- |
| Filling missing values for “Race”: |
| Dropping unnecessary columns: |

### 2.Data Integration

Data integration involves combining data residing in different sources and providing users with a unified view of them. This can lead to issues such as schema integration, redundancy, detection and resolution of data value conflicts. To identify and resolve such issues, following measures were performed.

#### Correlation Analysis

Correlation analysis is a statistical method used to evaluate the strength of relationship between two variables. A high correlation means that two or more variables have a strong relationship with each other, while a weak correlation means that the variables are hardly related.

Feedback:

As our targeted variable is binary in nature, we used spearman correlation method.

Previously, Pearson Correlation was used.

As shown in the following figure, there no significant issues related to correlations of the variables.

|  |
| --- |
| Spearman correlation method: |

|  |
| --- |
| Heatmap Correlation: |

#### Grouping

Columns named “Diagnosis\_1”, “Diagnosis\_2” and “Diagnosis\_3” have values represented in icd9 codes. In order to make these codes more readable, we converted them into 9 groups as per the pre-defined medical categories. ( <https://www.hindawi.com/journals/bmri/2014/781670/tab2/> )

|  |
| --- |
| Grouping of “Diagnosis\_1”, “Diagnosis\_2” and “Diagnosis\_3”. |

### 3.Data Transformation

Data transformation is the process of converting data from one format or structure into another format or structure. The following data transformation methods were performed in order to get the entire dataset in one single form.

#### Data Normalization

The dataset consists of some numerical columns such as “time\_in\_hospital”, “number\_lab\_procedures”, “number\_medications”, ets. In order to get these numbers in affixed range, Min-Max normalization was performed.

|  |
| --- |
| Min-Max normalization: |
| Data after normalization: |

#### Data Discretization

Some of the columns such as “Race”, “Gender”, “Age”, etc. with nominal values were required to be converted into numeric data and as a result, LabelEncoder() was used.

The pd.cut() was also used to convert numeric data into required groups.

In order to get the dataset in binary form, we converted the numeric data into nominal and created dummy variables.

|  |
| --- |
| Data before LableEncoder(): |
| LabelEncoder() used: |
| Data After LabelEncoder(): |

|  |
| --- |
| pd.cut():  A screenshot of a social media post  Description automatically generated |
| Dummy Variables: |

## 4.2. Data Splitting

Data splitting is the act of partitioning available data into. two portions, usually for cross-validating purposes. One. portion of the data is used to develop a predictive model. and the other to evaluate the model's performance. As the dataset has more 1,00,000 instances, we used hold-out evaluation to split the data into train and test set in 8:2 ratio.

|  |
| --- |
| Data Splitting: |

## 4.3. Treating Imbalance Data

To ensure better performance, we checked for imbalance data with respect to both target variables.

Feedback:

We applied SMOTE technique on training data after hold-out evaluation to obtain data balancing.

Previously, data balancing was performed on whole dataset before splitting.

### A. Predicting Re-admissions

The data with respect to variable “Re-admissions” was distributed in 53.92%, 34.93% and 11.96%. After applying SMOTE technique for over sampling, the improved distribution obtained is 49.7%, 32.2% and 18.11%.

|  |
| --- |
| Imbalanced data: |
| Data after oversampling: |

### B. Predicting Diabetic’s Medication Requirement

The data with respect to Diabetic’s Medication Requirement was distributed in 22.99% and 77.01%. After applying SMOTE technique for over sampling, the improved distribution obtained is 35.48% and 64.52%.

|  |
| --- |
| Imbalanced data: |
| Data after oversampling: |

## 4.3. Data Mining Methods and Processes

Classification is the most common task involved in data mining. To solve the proposed research problems, various classification tasks were performed.

### Predicting Re-admission

The following classification tasks were performed to predict the re-admissions in the hospital based on patient’s data.

#### 1. Naïve Bayes

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.

After initially applying Naïve Bayes classifier, accuracy of 42% was obtained along with 0.43 precision and 0.42 recall.

|  |
| --- |
| Naïve Bayes before balancing data:    Accuracy: 0.14 Precision: 0.145 Recall: 0.145 |
| Naïve Bayes after balancing data:    Accuracy: 0.42 Precision: 0.43 Recall: 0.42 |

To further improve the model, PCA technique was used to get top 20 significant variables.

|  |
| --- |
| PCA applied: |
| Naïve Bayes after PCA:    Accuracy: 0.56 Precision: 0.566 Recall: 0.566 |

After balancing the data and performing PCA, the accuracy, precision and recall for the model improved. Thus, the best algorithm is obtained with highest accuracy of 0.56.

#### 2.Decision Tree

A decision tree is a simple representation for classifying examples. Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. The accuracy obtained after initially applying decision tree was 48%. To further improve the results, we tried applying PCA for top 20 significant variables. As a result, we obtained accuracy of 46%.

|  |
| --- |
| Decision Tree:    Accuracy: 0.48 Precision: 0.48 Recall: 0.48 |
| Decision tree after PCA (top 20):    Accuracy: 0.46 Precision: 0.46 Recall: 0.46 |

Even after performing PCA for top 20 variables, the accuracy was not improved. Thus the highest accuracy obtained is 0.48.

#### 3. Bagging- Decision Tree Classifier

Bootstrap aggregating, also called bagging (from bootstrap aggregating), is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach.

Feedback:

We applied bagging algorithm to the decision tree classifier to get better accuracy. Previously, the classifier used for bagging was not mentioned.

Initial accuracy obtained with bagging was 55%. After applying PCA for top 20 variables, we got an accuracy of 55% as well.

|  |
| --- |
| Bagging – Decision Tree Classifier :    Accuracy: 0.55 Precision: 0.56 Recall: 0.56 |
| Bagging – Decision Tree Classifier: After PCA:    Accuracy: 0.55 Precision: 0.55 Recall: 0.55 |

Thus, the best working algorithm has accuracy of 0.55, precision of 0.56 and recall of 0.56

#### 4. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

We tried applying Random Forest algorithm to observe the results before and after applying PCA for top 20 variables. And thus, got the following accuracies:

|  |
| --- |
| Random Forest:      Accuracy: 0.57 Precision: 0.53 Recall: 0.57 |

|  |
| --- |
| Random Forest after PCA:      Accuracy: 0.54 Precision: 0.54 Recall: 0.54 |

As a result, the highest accuracy obtained is 0.57.

#### 5. Gradient Boosting- Decision Tree Classifier

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

Feedback:

We used gradient boosting method for decision tree classifier and observed the outcomes. Previously, the name of the algorithm on which gradient boosting is applied was not mentioned.

Gradient Boosting was also applied to the pre-processed data to observe the behaviors. Initially we obtained an accuracy of 58% but after applying PCA for top 20 most significant variables, it was changed to 54%.

|  |
| --- |
| Gradient Boost- Decision Tree Classifier:      Accuracy: 0.58 Precision: 0.58 Recall: 0.58 |

|  |
| --- |
| Gradient Boost- Decision Tree Classifier after PCA:      Accuracy: 0.54 Precision: 0.54 Recall: 0.54 |

The best working algorithm using gradient boost for decision tree classifier has an accuracy of 0.58.

#### 6. SVM- Support Vector Machine

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Lastly, SVM was performed and following results were obtained.

|  |
| --- |
| SVM - Support Vector Machine:    Accuracy: 0.57 Precision: 0.58 Recall: 0.58 |
| SVM- Support Vector Machine after PCA:    Accuracy: 0.56 Precision: 0.56 Recall: 0.56 |

Thus, the highest accuracy obtained for SVM is 0.57.

### Predicting Diabetic’s Medication Requirement

The following classification tasks were performed to predict the diabetic’s medication requirements for the patients based on their profile.

#### Naïve Bayes

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set.

After initially applying Naïve Bayes classifier, accuracy of 80% was obtained along with 0.77 precision and 1.00 recall.

|  |
| --- |
| Naïve Bayes before balancing data:    Accuracy: 0.82 Precision: 1 Recall: 0.766 |
| Naïve Bayes after balancing data:    Accuracy: 0.80 Precision: 0.77 Recall: 1.00 |

To further improve the model, PCA technique was used to get top 20 significant variables.

|  |
| --- |
| PCA applied: |

|  |
| --- |
| Naïve Bayes after PCA:    Accuracy: 0.79 Precision: 0.76 Recall: 1.00 |

The best working algorithm using Naïve Bayes classifier has accuracy of 80%.

#### 2. Decision Tree

A decision tree is a simple representation for classifying examples. Decision tree learning is a method commonly used in data mining. The goal is to create a model that predicts the value of a target variable based on several input variables. The accuracy obtained after initially applying decision tree was 1.00. Further, we tried applying PCA for top 20 significant variables. As a result, we obtained accuracies as 1.00 and 1.00 respectively.

|  |
| --- |
| Decision Tree:    Accuracy: 1.00 Precision: 1.00 Recall: 1.00 |
| Decision tree after PCA (top 20):    Accuracy: 1.00 Precision: 0.79 Recall: 0.77 |

As a result, highest accuracy obtained is 1.

#### 3. Bagging- Decision Tree Classifier

Bootstrap aggregating, also called bagging (from bootstrap aggregating), is a machine learning ensemble meta-algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach.

Feedback:

We applied bagging algorithm to the decision tree classifier to get better accuracy. Previously, the classifier used for bagging was not mentioned.

Initial accuracy obtained with bagging was 0.99. After applying PCA for top 20 variables, we got an accuracy of 0.78.

|  |
| --- |
| Bagging:    Accuracy: 0.99 Precision: 1.00 Recall: 1.00 |
| Bagging After PCA:    Accuracy: 0.78 Precision: 0.78 Recall: 0.93 |

#### 4. Random Forest

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes or mean prediction of the individual trees.

We tried applying Random Forest algorithm to observe the results before and after applying PCA for top 20 variables. And thus, got the following accuracies:

|  |
| --- |
| Random Forest:      Accuracy: 0.99 Precision: 1.00 Recall: 1.00 |

|  |
| --- |
| Random Forest after PCA:      Accuracy: 0.766 Precision: 0.79 Recall: 0.93 |

#### 5. Gradient Boost- Decision Tree Classifier

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.

Feedback:

We used gradient boosting method for decision tree classifier and observed the outcomes. Previously, the name of the algorithm on which gradient boosting is applied was not mentioned.

Gradient Boosting was also applied to the pre-processed data to observe the behaviors. Initially we obtained an accuracy of 1.00 but after applying PCA for top 20 most significant variables, it was changed to 0.79.

|  |
| --- |
| Gradient Boostng:      Accuracy: 1.00 Precision: 1.00 Recall: 1.00 |
| Gradient Boost after PCA:      Accuracy: 0.79 Precision: 0.78 Recall: 0.86 |

Here, the accuracy obtained without performing PCA is higher.

#### 6. SVM- Support Vector Machine

In machine learning, support-vector machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.

Lastly, SVM was performed and following results were obtained.

|  |
| --- |
| SVM:    Accuracy: 1.00 Precision: 1.00 Recall: 1.00 |

|  |
| --- |
| SVM after PCA:    Accuracy: 0.79 Precision: 0.76 Recall: 1.00 |

The accuracy obtained without applying PCA is higher than that obtained after performing PCA.

# **5. Evaluations and Results**

## 5.1. Evaluation Methods

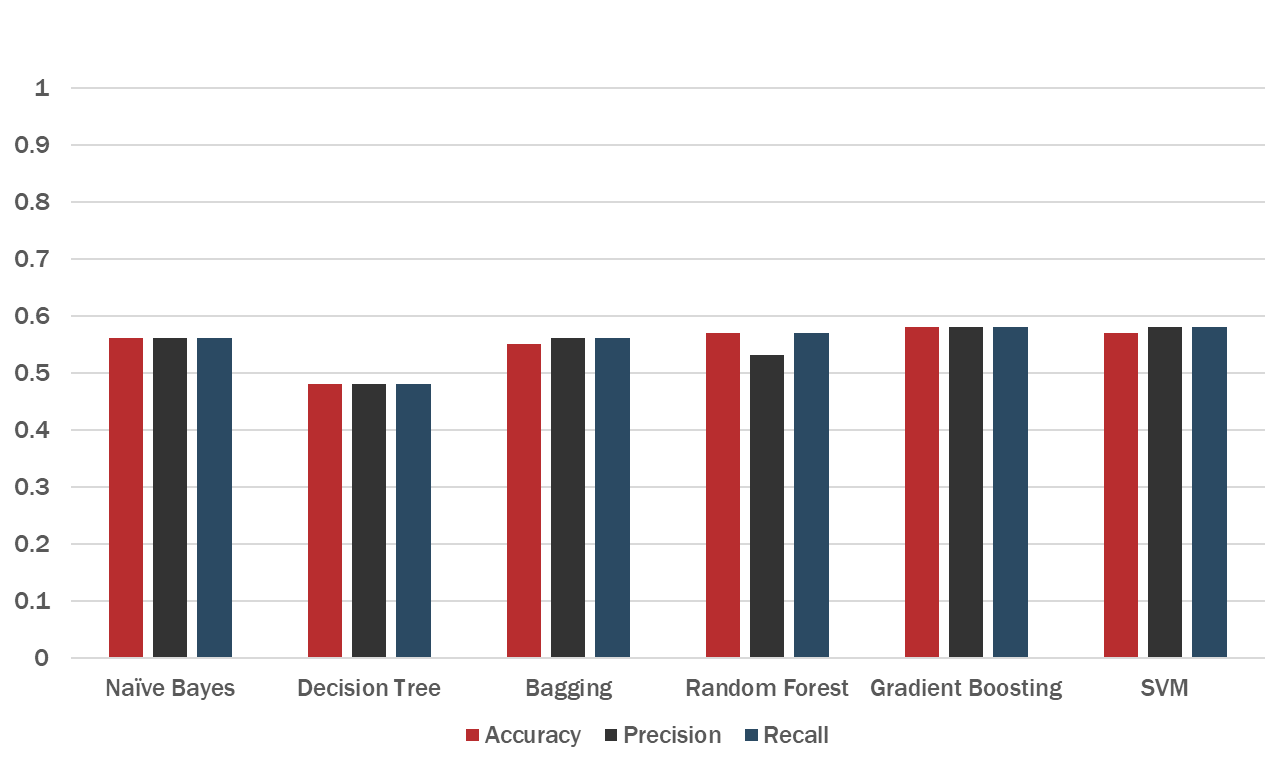
The observed models were evaluated based on accuracy scores, precisions and recalls, to find the most significant algorithm for both the proposed problems.

## 5.2. Results and Findings

### A. Predicting Re-admissions

The following findings indicate which algorithm worked best based on its accuracy, precision and recall for predicting the re-admissions of the patients.



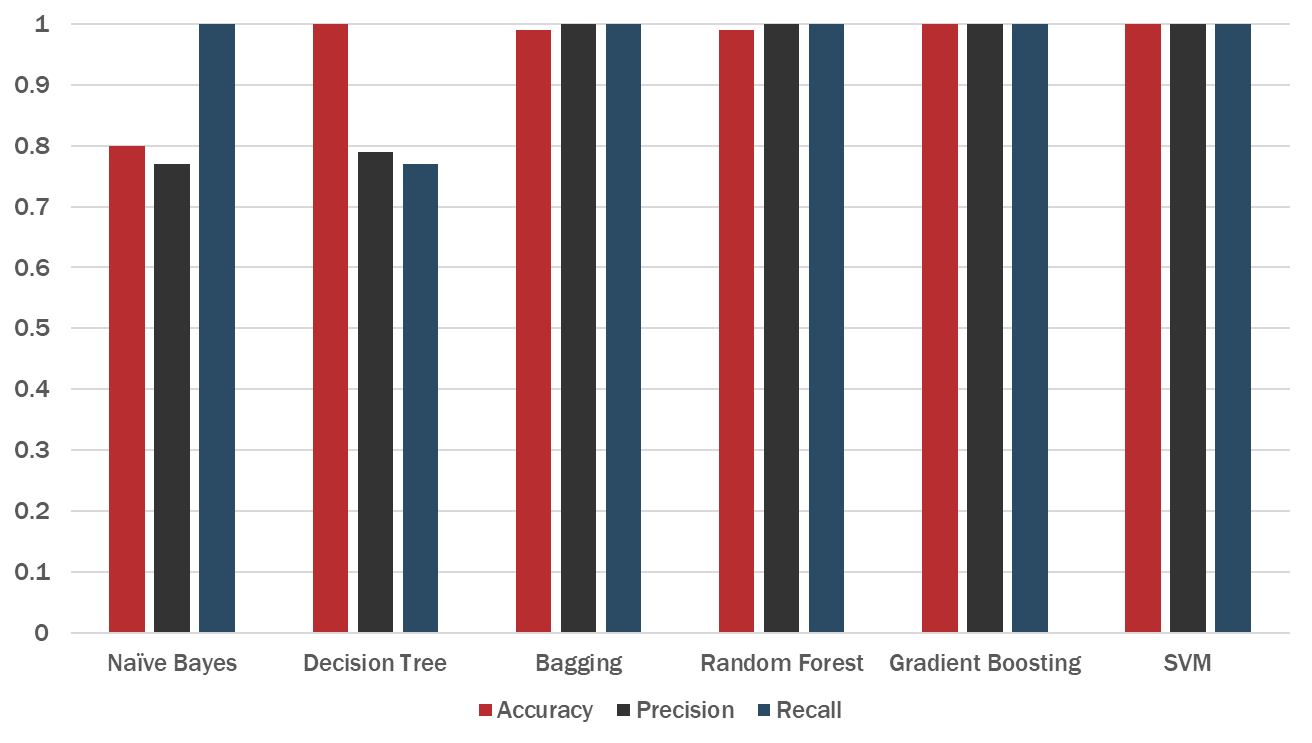


Predicting Re-admissions:

### B. Predicting Diabetic’s Medication Requirement

For predicting diabetic’s medication requirements, various algorithms were observed. The following data shows comparison of all the models based on accuracy, precision and recall.





Predicting Diabetic’s Medication:

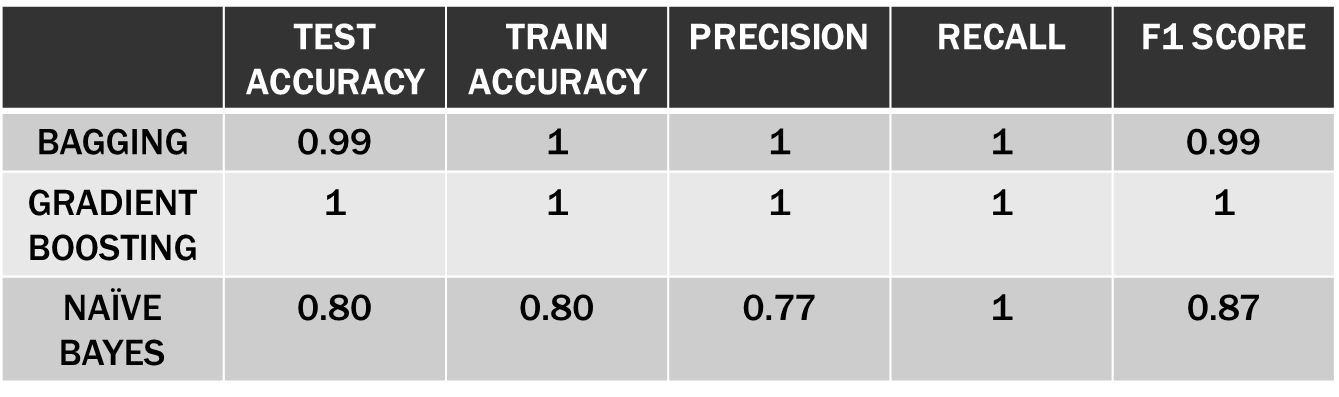
# **6. Over-fitting concerns**

The best solution obtained after comparing all the algorithms to predict diabetic’s medication requirement, seems to have too high accuracy and other evaluating factors. In order to confirm the trueness of it’s working we checked for over-fitting issues.

We obtained the training accuracy and test accuracy for Gradient Boosting and Bagging for Decision Tree algorithm as well as for Naïve Bayes.

|  |
| --- |
| Train and test accuracy for Gradient Boosting with decision tree classifier: |
| Train and test accuracy for Bagging with decision tree classifier:  A screenshot of a social media post  Description automatically generated |

|  |
| --- |
| Train and test accuracy for Naïve Bayes classifier:  A screenshot of a social media post  Description automatically generated |



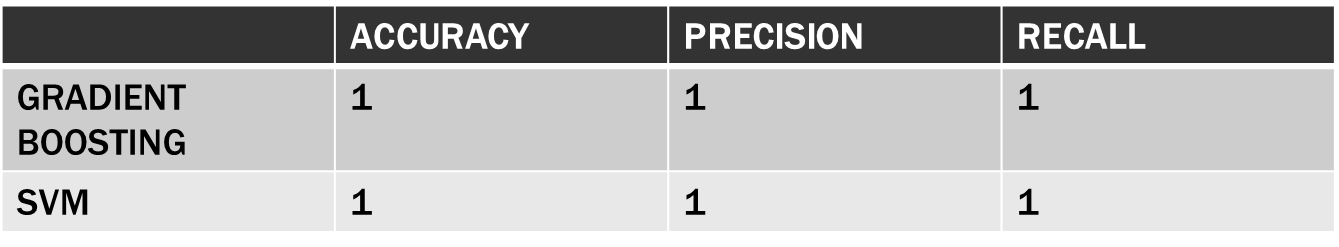
From the results obtained, it appears that there is no over-fitting problem.

# **7. Conclusions and Future Work**

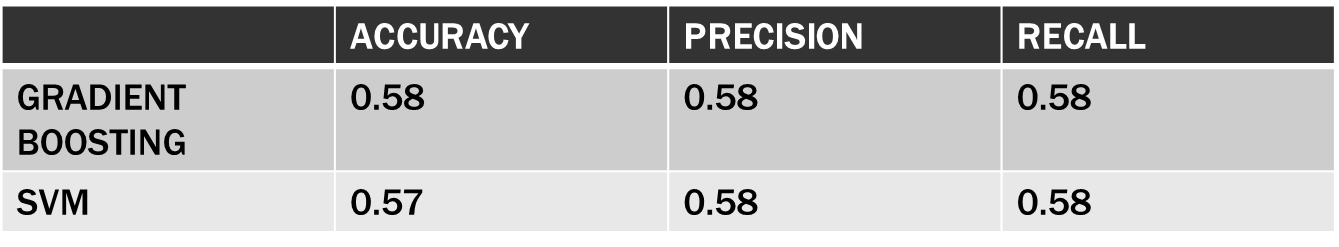
## 6.1. Conclusions

After comparing the outcomes of all the models and algorithm, following conclusions were drawn.

Highest accuracy obtained for predicting Diabetic’s Medication Requirements is of Gradient Boosting and SVM.



In order to predict the re-admissions of patients in hospitals, the best algorithms obtained are Gradient Boosting and SVM.



## 6.2. Limitations

The best accuracy obtained for predicting re-admissions of patients in hospital is very low and hence not much reliable.

## 6.3. Potential Improvements or Future Work

The accuracy for predicting Re-admissions of patients in the hospital can be further improved by processing more advanced data mining techniques.

If provided with more precise data in terms of time period (month, day, year), time series analysis can be performed to determine the entire period the patient might be required to be under medication in the hospital.

The data can be further used to build a recommendation system to help doctors to determine the type of medications to be prescribed to the patients.

# 8. References

1. Hindawi BioMed Research International : <https://www.hindawi.com/journals/bmri/2014/781670/#copyright>

2. UCI Machine Learning Repository:

<http://archive.ics.uci.edu/ml/datasets/Diabetes+130-US+hospitals+for+years+19992008>