Predicting Risk for Diabetes based on the Urine Sample using Different Algorithms

Dipesh Chand   
*Faculty of Computer Science*   
Østfold University College1783 Halden,Norway   
dipesh.chand@hiof.no

Bibek Acharya   
*Faculty of Computer Science*   
Østfold University College1783 Halden,Norway   
bibek.acharya@hiof.no

*Abstract*—Even though there are many advanced and sophisticated techniques that are being used to enhance the quality of health services, many of them still fail to produce accurate predictions when applied to real-time health data sets. Therefore, our aim is to devise the most accurate prediction model for an early diagnosis of diseases based on previously recorded patient data without performing any extensive laboratory tests which is the most needed application today. This project applied different algorithms to build the model to predict whether the person have diabetes or not based data on the sample of urine.

***Keywords—Diabetes, NHANES, CatBoost, KNN, LightGBM, SVM***

# INTRODUCTION

Bioinformatic is one of the most emerging field in today's time and Machine Learning has substantiated itself an extremely valuable equipment in this field. It has been utilized to extraordinary impact in early forecast of Diseases, for example, cancer {cite:insurgence}, and work is being done in anticipating the beginning of Alzheimer's and Parkinson's diseases {cite:insurgence}. These analysis depend on information from sequencing of gene and biomarkers, among different kinds of organic estimations. Endeavors have additionally been made to predict the initial stage of diseases, for example, diabetes dependent on overview of data and required information. On account of innovation that makes it a lot simpler to gather data and to study it, later on a greater amount of this sort of information may wind up accessible on a large number of the population. This bigger volume of survey data shows another chance to improve overall predictions of diseases, particularly in condition where lifestyle is profoundly related to diseases oncoming {cite:Incidence}.

The objective of the study is to discover connection, and design a model for diabetes discovery on a patients dependent on their urine test. The inspiration of this work is to locate a simple, convenient, inexpensive and non-painful (blood test) approach to screen diabetes.

This paper is structured as follows:

Chapter 2 presents the Data Description. Chapter 3 provides description about different related work. Chapter 4 explains about the different models used to perform the classification on this project. Chapter 5 explores about the result of the model and compares them. Chapter 6 discussed about the results of the modeling Chapter 7 pointing forward to conclusion and several possibilities in this project aim which can be performed in near future.

# DATA DESCRIPTION

This project utilizes the extensive National Health and Nutrition Examination Survey (“NHANES”) dataset, which is maintained and overseen by the Centers for Disease Control and Prevention headquartered in Atlanta, Georgia. The 2013-14 NHANES data set was obtained from the Kaggle web site {cite: National Health and Nutrition Examination Survey} and consists of information, measurements, and questionnaire and interview responses from thousands of patients randomly chosen from across the country every year.

The NHANES data set consists of six separate data tables, each in a comma-separated values file. Each data table has thousands of patient observations, with each observation containing information on between 13 and 952 measurements or responses from the patient. Significantly, each patient observation includes a unique patient identifier, called “SEQN” for “sequence number”, that allows a patient’s observations from across multiple data tables to be combined for analysis. As such, the NHANES dataset offers a valuable look at a wide variety of properties, measurements, and questionnaire responses for the same patient. Table 1 below presents the number of patients and the number of predictors (excluding the patient identifier) in each data table.

|  |  |  |
| --- | --- | --- |
| Data Table | Number of Predictors | Number of Patients |
| Demographics | 46 | 10175 |
| Dietary | 167 | 9813 |
| Examinations | 223 | 9813 |
| Laboratory | 423 | 9813 |
| Medications | 13 | 20194 |
| Questionnaire | 952 | 10175 |
| Total | 1824 | 69983 |

Table 1: The data tables of the NHANES data set.

The combined data set contains a total of 10175 distinct patients and 1824 distinct predictors, excluding the patient identifier column found in each data table. This study required information from only four of the data tables: Demographics, Examination, Laboratory and Questionnaire. All together this study have to initially deal with 1644 distinct predictors.

# RELATED WORK

In research done by Wei Yu et al., used the dataset from 1999-2004 National Health and Nutrition Examination Survey (NHANES) to develop and validate SVM model for two Classification scheme I : diagnosed or undiagnosed diabetes vs. pre-diabetes or no diabetes and Classification Scheme II :undiagnosed diabetes or pre-diabetes vs. no diabetes. They found the result that Support vector machine modeling is a promising classification approach for detecting persons with common diseases such as diabetes and pre-diabetes in the population.

Krati Saxena et al., K- Nearest neighbor algorithm for the diagnosis of diabetes mellitus for the dataset from Stanford University. And found the accuracy and error rate for K=3, 5. Their result shows that as the value of k increases, accuracy rate and error rate will also increase. {Diagnosis DiagnosisDiagnosis Diagnosis of of Diabetes Me Diabetes Me llitus using K Nearest llitus using K Nearest llitus using K Nearest llitus using K Nearest llitus using K Nearest llitus using K Nearest llitus using K Nearest llitus using K Nearest Neighbor Neighbor Neighbor Algorithm Algorithm Algorithm Algorithm Algorithm

Krati Saxena1, Dr. Zubair Khan2, Shefali Singh3

}

In experiment done by Anna Veronika Dorogush et al., to find which are significantly faster than other gradient boosting libraries on similar sizes of data found that CatBoost can be scored around 25 times faster than XGBoost and around 60 times faster than LightGBM.

Huang, G. et al., used CatBoost along with random forest and SVM for the estimation of Evapotranspiration in humid region. They used the meteorological data from five different station in southern china The data used were recorded in duration from 2001 to 2015. And they found that the CatBoost performed best with complete combination of parameter tuning where as without combination of parameter SVM is better.

Arkaprabha Saua, and Ishita Bhaktab used different machine learning algorithms for screening of anxiety and depression among the seafarers. They used algorithm like CatBoost, Random Forest, Logistic Regression, Naïve Bayes and SVM and found that found that Catboost to be best one for that purpose with based on accuracy and precision.

# METHODOLOGY

In this project, the data preprocessing was quite challenging (feature's name unintelligible, missing data, cross-checking between the different .csv). Past work has concentrated on the utilization of same survey data to predict the initial stage of diabetes in a huge sample of the population utilizing Support Vector Machines (SVM) {cite:Application of support vector}. The outcome of this study was created on the basis of the Receiver Operating Characteristics (ROC) curve. We will likely utilize this equivalent dataset and endeavor to improve overall precision. And also to more clearly describe the outcomes, and demonstrate how such a model might be utilized in reality.

The objective of this project is to have an estimation of the diabetes based on the urine sample and simple test and to improve the performance of model using various approaches. In order to obtain more accurate and precise model to predict desired output, we will performed several Machine Learning techniques and model . We perform the following steps in our study:

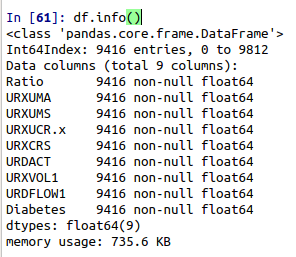
1. Data Preperation and Data Pre-processing
2. Data Modeling

All modeling was done using scikit-learn, a python-based toolkit for simple and efficient machine learning, data mining and data analysis {cite:Scikit-learn: Machine Learning in Python}.

## DATA PREPARATION AND DATA PREPROCESSING

The first step is to load the dataset into a data frame for easy manipulation and exploration.

Then, we selected urine data, demographic data and used the questionnaire answer to create label "Diabetes". Then we clean the data, and get rid of the feature with too many missing value. Finally, we oversample the diabetes label. After this step, we got the following data.

Fig.1. *Summary of the Dataset to be used in this project*

Following steps have been performed for the treatment of huge amount of data from different datasets of NHANES and to extract only those data which can be favrouble to utilizing in our study for predicting of diabetes.

### Feature Selection

The variables were picked to be like the variables which were utilized in {cite:Application of support vector}, since so much work had been already done in this past study to choose variables. Extra highlights were inspected, for example, diet, anyway upon further investigation these most were avoided because of their insignificant impact on execution or a remarkable segment of their information was missing (> 60% missing). Of the huge number of potential variables, we concentrated on just 3 attributes 'SEQN', 'BMXWT', 'BMXHT' of examination dataset, only 1 attribute 'DIQ010' of questionnaire dataset, 2 attributes 'RIAGENDR', 'RIDAGEYR'.of demographic dataset and the complete lab dataset. This made translation more straightforward and preparing time shorter.

### Dataset Creation

After feature selection from different data sets we created our own four datasets based on those selected attributes. Renaming of attributes were also done while creating these dataset so that it will be easy to understand. Feature cleaning and Feature processing were performed on these dataset and those attributes which shows high risk were dropped. Also missing value of each dataset were analyzed and only those attributes that have at least half the data and handled by filling mean data in the remaining features. The examination datasets which we have created contains only contains two attributes weight and height so we created new attributes ‘Ratio’ by calculating

Ratio = Weight/Height

and replacing them with it.

Then we merge all the four datasets to create single dataset using concate object of pandas with inner join.

Now the newly created dataset has 9416 data with 9 attributes of which 8 are Predictors and 1 is outcome.

### Managing Imbalance Data

A dataset is imbalancedif the distribution of output classes is not uniform. Here we have imbalanced data. By using this type of imbalance dataset problem may occurs such as:

1. The algorithms will be biased towards majority class. So it will predict the majority class irrespective of the predictors.
2. The test Accuracy of the model will be more than 90% but it won’t serve the our objective because the test data will also have more than 90% of the data pertaining to majority class. So, here, the accuracy won’t be a good representation of the model performance.
3. The algorithm assumes that errors obtained from both classes have same cost. But in this case, the Type II error (False Negative) error is very serious than the Type I error (False Positive). Because we should not miss the chance or the opportunity of identifying a potential customer. But we can afford to get a few false positive, because comparatively it won’t do much harm.

So to avoid these type of issues we need to deal with imbalance data. For this project, Synthetic Data Generation method is used. In regards to synthetic data generation, Synthetic Minority Oversampling Technique” (SMOTE) - a powerful and widely used method is applied. SMOTE algorithm creates artificial data based on feature space (rather than data space) similarities from minority samples. It generates a random set of minority class observations to shift the classifier learning bias towards minority class. To generate artificial data, it uses bootstrapping and k-nearest neighbors.

### Data Splitting

The dataset is divided into training data and test data with the intention of using the training data to find the parameters of the particular model being used (fitting the model on the training data) and then applying this to the test data to determine the model’s performance and to draw conclusions about its predictive capability. Before model training, a 20% test set was removed from the entire dataset which left 80% data for training.

## DATA MODELING

With all the data analysis and prepatration of new dataset for the study, only few features are left but we still could use various Machine Learning techniques.

In order to model the data, we used four different machine learning algorithms in this project to predict the Risk for Diabetes patients.

1. CatBoost Modeling
2. K-Nearest Neighbors Algorithm (k-NN)
3. LightGBM Modeling
4. Support-Vector Machine (SVM)

Each modeling process can be broken into following steps:

1. Import classifiers
2. Set multiples Parameters
3. GridSearchCV to find the best parameters
4. Load Classifiers with the best parameters
5. Predict with the modeling
6. Calculate Logloss , F1 Score, Precision, Accuracy and ROC AUC
7. Plot Confusion Matrix
8. Plot ROC Curve

At the end of each tests the Logloss , F1 Score, Precision, Accuracy and ROC AUC of latest model is shown with the previous all of the loaded model so that it will be easy to make comparison.

### Catboost Modeling

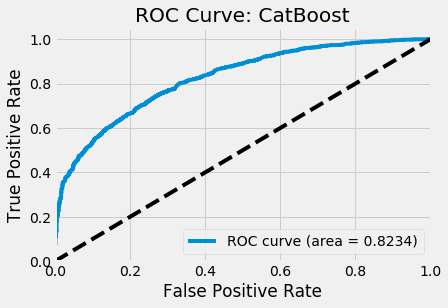
CatBoost is a open-source machine learning algorithm from Yandex {<https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/>}. CatBoost is a fast, scalable, high performance open-source gradient boosting on decision trees library. {<https://catboost.ai/>} It provides state-of-the-art results without extensive data training typically required by other machine learning methods, and provides powerful out-of-the-box support for the more descriptive data formats that accompany many business problems. The major advantages of CatBoost are it provides efficient and result and is competitive in performance basis, handling categorical features automatically as Cat in Catboost means categories of data, it is robust as it reduces the need for extensive hyper-parameter tuning and lower the chances of over fitting.

We calculate best parameter for CatBoost through GridSearchCV by using cross validation of 3 fold and found

{'depth': 10, 'iterations': 200, 'l2\_leaf\_reg': 4, 'learning\_rate': 0.15}

which we used during modeling.

The ROC curve of CatBoost is displayed in Figure



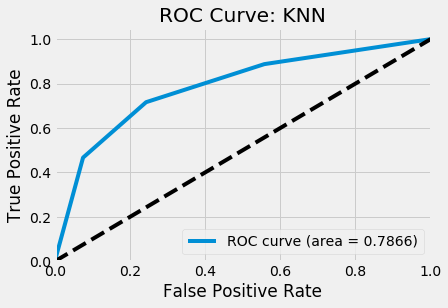
*Fig.5. ROC curve of Catboost*

### K-Nearest Neighbors Algorithm (KNN)

K-Nearest Neighbors algorithm (KNN) is a non parametric method used in machine learning for classification and also for regression. The k nearest-neighbor classification rule assigns an input sample vector, which is of unknown classification, to the class of its nearest neighbor. The vector is assigned to the class that is represented by a majority amongst the K-nearest neighbors. There is the possibility of being more than one class so the value of K is restricted. {<https://ieeexplore.ieee.org/abstract/document/376493>}

We use n\_neighbors=3 while modeling KNN and calculate the result based on that.

The ROC curve of KNN is displayed in Figure



*Fig.5. ROC curve of KNN Model*

### LightGBM Modeling

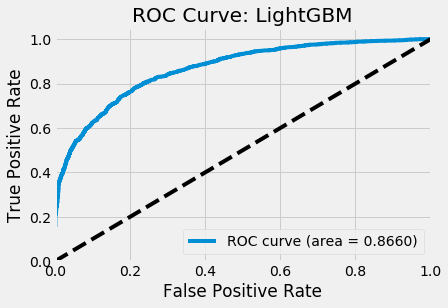
LightGBM is a gradient boosting Algorithm that utilize tree based learning algorithms. {<https://lightgbm.readthedocs.io/en/latest/>} Light GBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise. It will pick the leaf with max delta loss to grow. When developing a similar leaf, Leaf-wise algorithm can decrease more loss than a level-wise algorithm. {<https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc>}. The advantage of LightGBM is that it gives Faster training speed and higher efficiency with lower memory usage and better accuracy. It also supports parallel and GPU learning and is capable of handling large scale data.

We calculate best parameter for LightGBM through GridSearchCV by using cross validation of 3 fold and found

(boosting\_type='gbdt', class\_weight=None, colsample\_bytree=1.0, importance\_type='split', learning\_rate=0.1, max\_depth=75, min\_child\_samples=20, min\_child\_weight=0.001, min\_split\_gain=0.0, n\_estimators=100, n\_jobs=-1, num\_leaves=300, objective=None, random\_state=None, reg\_alpha=0.0, reg\_lambda=0.0, silent=False, subsample=1.0, subsample\_for\_bin=200000, subsample\_freq=0)

which we used during modeling.

The ROC curve of LightGBM is displayed in Figure



*Fig.5. ROC curve of LightGBM Model*

### Support-Vector Machine (SVM) Modeling

SVM is a supervised machine learning algorithm which is mostly used for classification but also can be used for regression. In this algorithm, we plot each data item as a point in n-dimensional space is plotted (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then the classification is done by finding the hyper-plane that differentiate the two classes very well.{<https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>} The advantages of support vector machines is it is effective in high dimension spaces where number of dimensions is greater than the number of samples, it is also memory efficient as it uses a subset of training points in the decision function called support vectors.{<https://scikit-learn.org/stable/modules/svm.html>}

We calculate best parameter for SVM through GridSearchCV by using cross validation of 5 fold and found

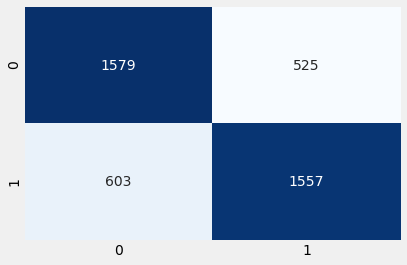
which we used during modeling.

The ROC curve of LightGBM is displayed in Figure

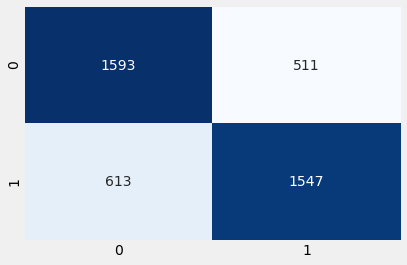
*Fig.5. ROC curve of SVM Model*

# RESULTS

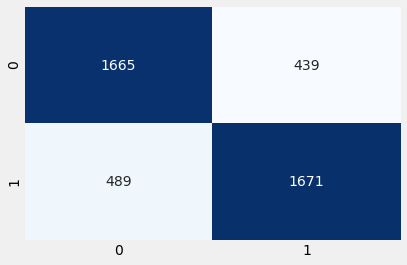
We performed four different Machine learning models to classify whether a patient have risk of diabetes or not based on the data sample of urine. We evaluate different test results of our models like Logloss, F1 Score, Precision, Accuracy, ROC AUC. We also consider Confusion Matrix metrics for evaluation which is a breakdown of predictions into a table showing correct predictions (the diagonal) and the types of incorrect predictions made (what classes incorrect predictions were assigned). Figure 11, Figure 12, Figure 13, and Figure 14 shows the details of confusion matrix of different models used in this project.



*Fig.11. Confusion Matrix of CatBoost model*



*Fig.12. Confusion Matrix of KNN model*



*Fig.13. Confusion Matrix of LightGBM model*

*Fig.14. Confusion Matrix and Statistics of XGBoost model*

The undermentioned table demonstrates the *Logloss, F1 Score, Precision, Accuracy and ROC AUC* of the four machine learning models we have used in our project using sklearn.metrics.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Logloss | F1 Score | Precision | Accuracy | ROC AUC |
| CatBoost | 9.136998 | 0.734088 | 0.747839 | 0.735460 | 0.823425 |
| KNN | 9.104595 | 0.733523 | 0.751701 | 0.736398 | 0.786645 |
| LightGBM | 7.516964 | 0.782670 | 0.791943 | 0.782364 | 0.865960 |
| SVM |  |  |  |  |  |

*Table.1. Logloss, F1 Score, Precision, Accuracy and ROC AUC of different models*

From the table, it can be seen that Random Forest and XGBoost are the top performing algorithms. Even though the differences in all the metric values are infinitesimal, we will choose XGBoost in terms of predicted accuracy and Random Forest as the best model for our prediction because it ranks first in 2 out of the 3 considered metrics .

# DISCUSSION

Four models were utilized to characterize diabetics and non-diabetics dependent on review information gathered from NHANES. These five models were given equivalent weighting in a troupe model that utilized the probabilistic yield of each model. From the ROC curve of the considerable number of models (found in Figure 2) it was discovered that the best performing model (the most astounding AUC, found in Table 2) was the Gradient Boosting Classifier, which really beat out the group strategy.

We were astonished that the solitary Gradient Boosting Classifier performed best, as we imagined that the shrewdness of each model would consolidate into the best classifier through an outfit technique. This would have been steady with the hypothesis of the 'intelligence of groups.'

The more unfortunate execution of the group strategy could be because of a couple of components. The first is that there was deficient hyperparameter tuning. To anticipate the need to tune 10 hyperparameters at the same time, which would have taken an over the top measure of time, we rather tuned each model independently. In doing this we counteracted the group model from truly utilizing the best highlights of each model, as we didn't take into account the tuning of hyperparameters dependent on gathering execution. An extra hyperparameter that ought to have been enhanced was the weighting of each model. Our gathering model weighted each model similarly, which may have given a lot of weight to more unfortunate performing models, for example, the K-Nearest Neighbors Classifier (the AUC for this model was much lower than the AUC of the other 4 models).

Later on, if computational time were not an issue, synchronous tuning of hyperparameters would be completed. Extra execution improvement could likewise be accomplish by actualize ing a 'stacked model,' which is basically a two-layered model that would utilize the yield of every individual model as a component to prepare a lower dimensional model.

One of our preprocessing steps was to ascribe values for missing information. This is especially significant for where 25% of cells are missing qualities. For numerical highlights (for example BMI and tallness) we determined the mean from the preparation information and alloted it to missing fields in both preparing and test sets. For straight out factors (for example level of pay and liquor use) we doled out missing qualities the most widely recognized name from the preparation set. Further execution upgrade could likewise be acquired ascribing missing information by structure a model from the accessible to foresee missing qualities for a given element and through grid factorization.

The top AUCs from the models displayed in this paper were predictable with crafted by [6], who accomplished and AUC of 0.83 for a well-tuned bolster vector machine. Our consistency with this past investigation shows that the 0.83 AUC might be a hard maximum utmost on the discriminative intensity of models prepared on the NHANES information with the end goal of the forecast of diabetes utilizing straightforward order procedures, for example, those talked about in the present work and in [6]. Be that as it may, considering minor exertion was made to attribute information in either works, astute information ascription may introduce the best way for expanding execution.

# CONCLUSION

The results explained in the previous chapter concludes the study. We managed to get results from the given data set and compare algorithms, which was our primary goal for this project. While programming, interpreting the results and studying the results, whole research group got a hands-on experience with modern tools for computational data analytic with a real business case.

The next step for this project would be the optimization of each algorithm and implementing neural network. In this project, default settings of each algorithm was used. Adjusting settings to get better results would require a more in-depth study of each algorithm to know how the performance could be improved.

##### References

[1] “UCI Machine Learning Repository: Bank Marketing Data Set.” [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Bank+Marketing. [Accessed: 14-Mar-2019].

[2] S. Moro, P. Cortez, and P. Rita, “A data-driven approach to predict the success of bank telemarketing,” *Decis. Support Syst.*, vol. 62, pp. 22–31, Jun. 2014.

[3] C. Vajiramedhin and A. Suebsing, “Feature selection with data balancing for prediction of bank telemarketing,” *Appl. Math. Sci.*, vol. 8, pp. 5667–5672, 2014.

[4] H. A. Elsalamony and A. M. Elsayad, “Bank Direct Marketing Based on Neural Network and C5.0 Models,” vol. 2, no. 6, p. 10.

[5] A. Criminisi, E. Konukoglu, and J. Shotton, “Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning,” Oct. 2011.

[6] “C5.0 Node,” 24-Oct-2014. [Online]. Available: undefined. [Accessed: 19-Mar-2019].