DEVELOPING ADVANCED MACHINE LEARNING BASED MODELS FOR EARLY DETECTION FOR DIABETES AND HEALTH INDICATORS USING EXPLAINABLE MACHINE LEARNING

Table of Contents

1.0 INTRODUCTION

1.1 Background

Diabetes is a disease that is increasingly affecting the world even the most developed countries. Lately, the predominance of diabetes has been growing yearly A study estimated that by 2040, the proportion of adult diabetics worldwide without factoring age group within the range of 0-19 is expected to increase to 10.4%, or about 642 million diabetics worldwide. Diabetes, by virtue of its evolution as a worldwide significant disease, necessitates the highest level of dedication from medical personnel, patients, families, and society. Diabetes has severe social, health, and economic costs (Rodriguez-Sanchez.et.al 2021) Diabetes is a chronic condition defined by an increase in glucose or blood sugar levels due to the body's inability to create insulin, insufficient insulin secretion, or insulin inability to function on the organism's cells. Historically, diabetes treatment has focused on treating the symptoms rather than the underlying cause. The World Health Organization states that, Diabetes affects about 5% of the world's population and the number of patients is constantly increasing (Roglic 2016). In developed countries, diabetes and the largest number of diabetics are found in people over 65 years of age. Whereas in developing countries the largest number of diabetics is found in the age of 45-64 years, but in recent years type 2 diabetes is more commonly encountered also in the age of 30-40 years. The availability of historical data naturally leads to the application of advanced machine learning model for pattern discovery and prediction. The goal is to find rules that help understand diabetes and make it easier to diagnose it sooner. Prevention of diabetes is of great interest in the field of medicine. The use of machine learning algorithm accelerates data analysis (Clustering) prediction, to help analysts examine existing data to identify patterns and trends of diabetes (Bo He, 2019). This paper is structured as follows: Section. 2 would describes the relationship that exists between, machine learning, data acquisition, Data Preprocessing, Feature Engineering and Machine Learning Algorithm. The methodology and description of the dataset are described in Section. 3. Sections. 4 and 5, represent a theoretical description of the result analysis and project evaluation and reflection.

1.2 Aim

The current study examined historical trends of diabetes care in patients with diabetes admitted to a US hospital in order to guide future paths that could lead to improvements in patient safety. Early hospital readmissions are being reduced as a policy priority to improve healthcare quality.

In this case study, we look at how machine learning can assist in resolving challenges created by readmission.

1.3 Objective

In the United States, diabetes is thought to affect (37.3Million)11.3% of the population, (28.5Million)8.7% of the population whom go diagnosed while 2.6% of the population go undiagnosed, it prevalence climbed to 29.2% among persons aged 65 and older. Diabetes patients have a readmission rate of 14.4 to 22.7% after 30 days. Over 26% of diabetes patients are readmitted within 3 months, and 30% within a year, according to estimates of readmission rates after 30 days from hospital discharge (NIDDK 2022). In the USA, diabetes patients spent $124 billion hospitalizing, of which $25 billion was thought to be related to 30-day readmissions assuming a 20% readmission rate. Therefore, lowering the 30-day readmission rate for diabetic patients has the potential to significantly lower healthcare expenses while also enhancing care.

1.4 Ethical, Social, Professional, Legal and Security Consideration

This research paper is established on the UC Irvine Machine Learning Repository Dataset Store which was donated from over 130-US Hospital for years 1999-2008 about diabetes. The dataset contains no personally identifiable information (PII). Due to the anonymized nature of the datasets obtained, this study was not considered human subjects research nor required consent. (Clore,John et al, 2014)

2. DATA

The Dataset gathered to be used for this data contain a span of ten years of clinical care health record data from Different Hospital in the United State which accumulate to 130 Hospital. Each of the data row contains information about a patient who has undergo a medication encounter, diabetic encounter, stayed up to 2 weeks at maximum and laboratory test was done. Despite the high-quality result which have showed improved outcomes from clinical test including therapeutic intervention and preventive intervention in regards to diabetic patient, many patients do not still receive them, this is attributed to casual diabetes management in different hospital environments (Strack, 2014). For Data Quality and privacy protection for the patient each Hospital name for which the data has been gathered have been removed and the Data itself anonymized (Personal Identifiable Information Removed).

2.1 DATA DESCRIPTION

The Data Contained 50 Features where Forty-Nine (49) are the Input Features and One (1) Target column having 101766 distinct (encounter) rows which in total contains 13 Numerical features and 37 categorical features.

S/N	Feature Name	Type	Description	Nunique	Count
1	Id	Integer	Unique identifier	101766	101766
2	Encounter_id	Integer	Unique identifier of an encounter	101766	101766
3	Patient_nbr	Integer	Unique identifier of a patient	71518	101766
4	Race	Object	Values: Caucasian, Asian, African American, Hispanic, and other	5	99493
5	Gender	Object	Values: male, female, and unknown/invalid	3	101766
6	Age	Object	Grouped in 10-year intervals: [0, 10), [10, 20),..., [90, 100)	10	101766
7	weight	Object	Weight in pounds.	9	3197
8	Admission_type	Integer	Integer identifier corresponding to 8 distinct values, for example, emergency, urgent, elective, newborn, and not available	8	101766
9	Discharge_disposition_id	Integer	Integer identifier corresponding to 26 distinct values, for example, discharged to home, expired, and not available	26	101766
10	Admission_source_id	Integer	Integer identifier corresponding to 17 distinct values, for example, physician referral, emergency room, and transfer from a hospital	17	101766
11	Time_in_hospital	Integer	Integer number of days between admission and discharge	14	101766
12	Payer_code	Object	Integer identifier corresponding to 23 distinct values, for example, Blue	17	61510

			Cross/Blue Shield, Medicare, and self-pay		
13	Medical_specialty	Object	Integer identifier of a specialty of the admitting physician, corresponding to 72 distinct values, for example, cardiology, internal medicine, family/general practice, and surgeon	72	51817
14	Num_lab_procedures	Object	Number of lab tests performed during the encounter	118	101766
15	Num_procedures	Object	Number of procedures (other than lab tests) performed during the encounter	7	101766
16	Num_medications	Object	Number of distinct generic names administered during the encounter	75	101766
17	Number_outpatient	Object	Number of outpatient visits of the patient in the year preceding the encounter	39	101766
18	Number_emergency	Object	Number of emergency visits of the patient in the year preceding the encounter	33	101766
19	Number_inpateint	Object	Number of inpatient visits of the patient in the year preceding the encounter	21	101766
20	Diag_1	Object	The primary diagnosis (coded as first three digits of ICD9); 716 distinct values	716	101745
21	Diag_2	Object	Secondary diagnosis (coded as first three digits of ICD9); 748 distinct values	748	101408
22	Diag_3	Object	Additional secondary diagnosis (coded as first three digits of ICD9); 789 distinct values	789	100343
23	Number_diagnoses	Object	Number of diagnoses entered to the system	16	101766
24	Medical drugs	Object	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	4	101766
25	Change	Object	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: change and no change	2	101766

26	diabetesMed	Object	Indicates if there was any diabetic medication prescribed. Values: yes and no	2	101766
27	readmitted	Object	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.	3	101766

Medical Drugs administered: max_glu_serum, A1Cresult, metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide,glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, insulin, glyburide.metformin, glipizide.metformin, glimepiride.pioglitazone, metformin.rosiglitazone, metformin.pioglitazone,

2.2 DATA PREPROCESSING

The data gotten contains noisy information as this was expected since the data is a real-world data, performing data preprocessing is crucial to clean the data, eliminating noise, inconsistent and redundant features. The data check if the data contained duplicate encounter as this would create redundancy and not improve the performance but rather increase the computational complexity of the model, while some features where removed as they contained unique identifier (id, enocunter_id, patient_nbr, payer_code) further preprocessing showed that some features have missing values which was expected, different data imputation technique was used in filling the missing data which include mean and mode(Bai 2015), this features are race (2.23%), medical_specialty(49.08%), diag_1(0.02%), diag_2(0.35%), diag_3(1.39%) and weight which have above 90% was filled using the mean.

Most algorithms used in the area of machine learning are designed previously to with an assumption that the data used in training is balance and implicitly assume that all misclassification errors have the same cost, but this is not the case when dealing with real-world data which is usually imbalanced (Gu, 2008). The imbalance target data issue is widespread and troublesome, affecting a significant portion of the data community in building predictive models.

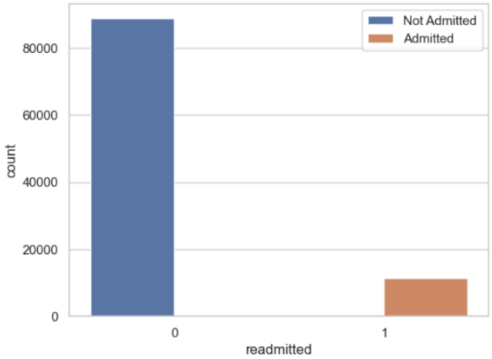


Figure 1: Showing the Target Column Distribution

Data transformation was carried on the data to transform it to a usable form which would be analyzed for Performing predictive analysis notably “medical_specialty” feature, which has too many distinct values,

so when applying one hot encoding, it created lot of features which added noise, hence resulted to use a frequency-based method with addition to domain knowledge for mapping to different category, for diagnois_1,2,3 mapping was done utilizing the ICD Values. New features where created from existing features using feature interaction and engineering new features using various Descriptive Statistics Technique for Both Numerical and Categorical variable (Mean, Max, Count) aggregating one column over another.

2.3 STATISTICAL ANALYSIS

Statistical Analysis was done to identify trends in the data starting with an analysis to use spearman correlation coefficient (de Winter, 2016) to check whether numerical input features and target (output) feature are dependent or independent, features found to be independent on target feature where removed. performing correlation analysis, it was found that the value is always close to zero but spearman which was used doesn’t capture non-linear relationship, hence we used the *pvalue* instead of the coefficient of correlation to create a list of features which would be rejected. Here hypothesis testing is done assuming null hypothesis to be "variables are independent" so assuming significance level = alpha = 0.05 this value has been proven to be standard when selecting significance of a p-value (Di Leo 2020).

if *pvalue* < *alpha* then reject null hypothesis that is we accept variables are dependent.

Rejected Features -- num_procedures, nateglinide, chlorpropamide, acetohexamide, glipizide, tolbutamide, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, glyburide.metformin, glipizide.metformin, glimepiride.pioglitazone, metformin.rosiglitazone, metformin.pioglitazone, average_lab_procedure_cost

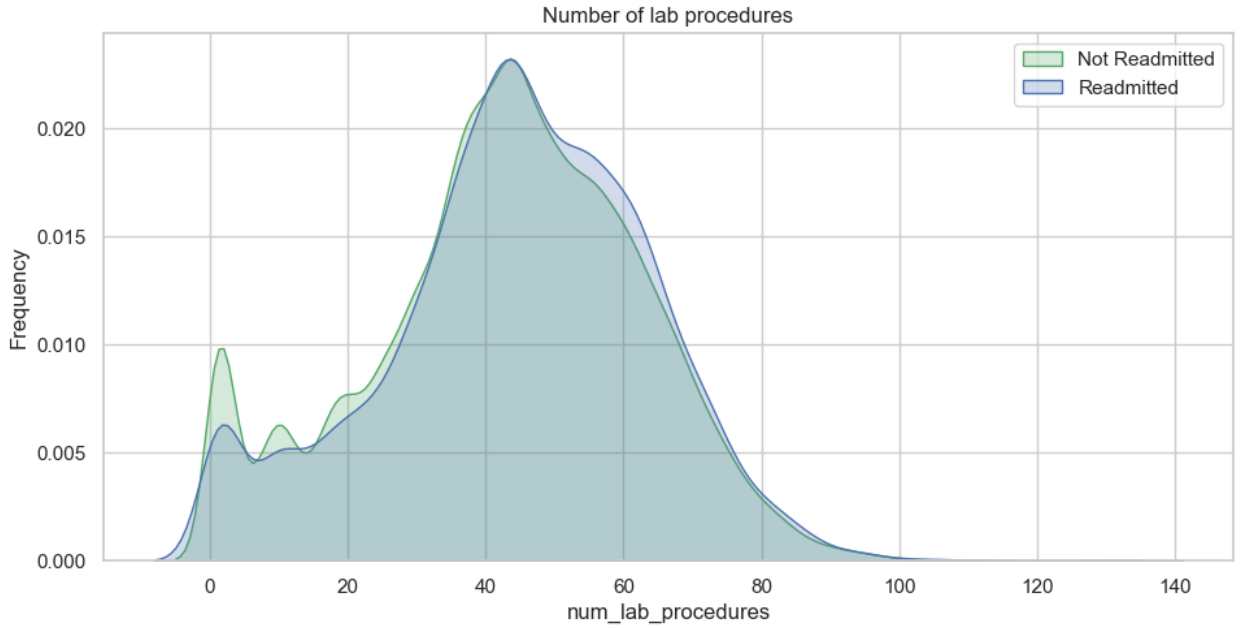
Table below shows the Mean, Standard Deviation, Skew, Kurtosis and T-Stat of the data

FEATURE NAME	MEAN	STD DEV	SKEW	KURTOSIS	T-STAT
age	65.830953	15.947425	-0.626691	0.274194	2.932362e-05
admission_type_id	1.783667	1.330994	1.441207	0.703495	2.781926e-04
discharge_disposition_id	2.088359	3.725944	4.127874	15.939251	9.792589e-04
admission_source_id	5.119314	2.880833	-0.570573	-1.390028	4.177205e-04
time_in_hospital	4.389646	2.974531	1.137864	0.870927	6.358810e-05
num_lab_procedures	42.943305	19.620940	-0.241456	-0.253291	2.155715e-04
num_procedures	1.330723	1.700286	1.326109	0.891054	8.314389e-05
num_medications	15.981821	8.092511	1.333098	3.523620	1.224501e-04
number_outpatient	0.369429	1.264006	8.817999	148.544463	6.898910e-05

FEATURE NAME	MEAN	STD DEV	SKEW	KURTOSIS	T-STAT
number_emergency	0.198334	0.935537	22.841802	1185.155821	2.841709e-05
number_inpatient	0.632829	1.261833	3.626420	20.833069	3.114327e-05
number_diagnoses	7.409164	1.938288	-0.867576	-0.109544	3.350850e-04
max_glu_serum	9.075654	42.879610	5.283608	28.806573	1.484103e-04
A1Cresult	1.162685	2.635304	1.913712	1.852590	6.911284e-05
metformin	-15.975788	8.180076	1.606534	0.785317	5.147915e-08
repaglinide	-19.688555	2.532517	8.221573	67.758646	-6.452732e-06
chlorpropamide	-19.982520	0.605211	35.375575	1284.681756	-2.636540e-06
glimepiride	-18.958687	4.529928	4.239244	16.626671	3.144879e-05
glyburide	-17.864534	6.324902	2.735429	5.906418	2.868863e-05
tolbutamide	-19.995805	0.289634	69.024152	4762.333543	-6.630901e-07
pioglitazone	-18.534171	5.266930	3.368455	9.613615	1.751111e-05
rosiglitazone	-18.726951	4.926771	3.660947	11.666777	3.140684e-05
acarbose	-19.937971	1.124153	18.291449	338.020819	3.695772e-07
insulin	-9.417364	10.995387	0.384750	-1.359391	6.439308e-05
change	-0.072198	0.997395	0.144773	-1.979041	6.116113e-05
diabetesMed	0.771840	0.419648	-1.295568	-0.321504	3.437836e-05
readmitted	0.113441	0.317132	2.437855	3.943136	2.084673e-06
health_index	0.832090	0.296320	-1.330674	0.056182	8.968118e-05
severity_of_disease	0.000010	0.000004	0.184378	0.127745	1.393558e-04
number_of_changes	0.287542	0.487859	1.425330	1.433659	8.282556e-05
total_procedures	44.274028	19.781814	-0.217825	-0.208645	2.022932e-04
total_medical_interactions	1.200591	2.292775	5.334641	67.774970	6.664321e-05
medication_ratio	5.061163	3.806133	2.210864	6.984680	2.198539e-04

FEATURE NAME	MEAN	STD DEV	SKEW	KURTOSIS	T-STAT
avg_procedures_per_visit	inf	NaN	NaN	NaN	NaN
diagnoses_per_procedure	0.379956	0.972591	6.516748	47.928591	1.468025e-04
time_in_hospital_per_procedure	inf	NaN	NaN	NaN	NaN
number_medications_per_diagnosis	2.282230	1.322322	2.459574	12.722960	1.818657e-04
emergency_room_visit_rate	0.000002	0.000009	22.841802	1185.155821	2.841709e-05
inpatient_admission_rate	0.000006	0.000013	3.626420	20.833069	3.114327e-05

Distribution Number of lab procedures for readmitted and not readmitted patient is exactly same. But it has high variance. High variance features are considered information rich features.



2.4 PACKAGES USED FOR ANALYSIS

- **Pandas**: is an open-source Python Data Analysis Library, free to use (under a BSD licens e); it is one of the most popular and widely used data munging/wrangling tools. It gathers data (from a CSV or TSV file or a SQL database) and turns it into a Python data frame wi th rows and columns like a table in statistical applications such as Excel or SPSS (Bronsht ein Adi).

- **NumPy**: It is a Python library that includes a multi-dimensional array adding support for multi-dimensional arrays and matrices as well as several derivative objects (such as mask

ed arrays and matrices, contains a variety of routines for performing quick array operation s, such as mathematical, logical, basic statistical operations, shape manipulation, selecting , sorting, I/O, random simulation, discrete Fourier transforms, and more (GeeksforGeeks, 2021)

- **Scikit-Learn**: Scikit-learn is a free machine learning library that focuses on data modellin g data by providing a consistent Python interface for supervised and unsupervised learnin g algorithms. It features many classifications, Regression, and clustering algorithms and i s designed to work with other scientific and numerical libraries, including NumPy (J. Bro wnlee 2021)

- **Scipy:** SciPy is a library of numerical routines for the Python programming language that provides fundamental building blocks for modeling and solving scientific problems. SciP y includes algorithms for optimization, integration, interpolation, eigenvalue problems, al gebraic equations, differential equations and many other classes of problems; it also provi des specialized data structures, such as sparse matrices and k-dimensional trees (Virtanen, P., 2020)

3.0 COMPARATIVE ANALYSIS

In order to determine whether a patient has diabetes or not based on the analysis and preprocessing done on the data as stated in chapter 2, our goal in this research is to use an improved yet robust classification scheme from various machine learning classifiers that have been implemented/developed by various research teams, ranging from from Lightgbm(Machado, M.R, 2019), Catboost (Dorogush, A.V, 2018), RandomForest (Breiman, L., 2001, Hackeling, G., 2017), Logistic Regression(Karp, A.H, 1998),Gradient Boosting Classifier(Bentéjac, C, 2021) etc. Based on the review of the literature, we conclude that current machine learning classifiers perform adequately well on related modeling and necessitate significant expert consultation for pre-processing and performance tuning. Optuna (Akiba, T.2019) was used in this process because it has been demonstrated to provide the best parameter search result based on data and algorithm search parameter.

That was required because of variations in domains and so in data. We wish to create a robust classifier that is also compact and portable for our use case, which focuses on comprehending diabetic patterns. In order to do this, we first conduct an experimental study that functioned as the baseline model using the default search parameters. After that, we develop a cooperative setup to create an ensemble learning approach. The comparative comparison of current algorithms is the primary driver behind this research's experimental investigation, which aims to identify the best performing algorithm for subsequent parameter training. Our experiment begins by training a subset of classifiers on a provided dataset across multiple iterations, providing a general perspective. Moreover, cycles of parameter adjustment will be applied to the top classifier after that. We have the best classifiers at the end of this stage. From a great perspective, the software development life cycle has largely impacted our study technique when assessing the quality of our model. (O. Adepoju, Romi S. 2014), and (Moura, A. F. D. 2014) both use the same methods. But the approach we discovered was really similar to what we had done (Shah, S. Gala and N. Patil, 2014). They chose to examine three classifiers in order to rank their performance on the heart disease dataset. Since the goal of the research is to find the best performing classifier that understood the problem better using the specified evaluation metric rather than to choose a single best classifier, it can be used for final work at a later time; their methodology was extended as follows:

▪ The evaluation phase is the name given to the training phase. Every candidate classifier will undergo individual training and performance tweaking.

▪ We choose our algorithms based on a variety of criteria, including regression, tree-based models, and neighboring models, therefore the number of classification methods is not defined in our process. For instance, the eight classification techniques we use in this work can differ based on the size of the dataset and the feature groups that are related to it.

▪ Due to the target data's unequal class size, we employ a stratified k-fold split in place of a percentage split (Khushi, M., 2021).

▪ Every potential classifier will be assessed using the F1 Score weighted average, which is determined by combining Precision and Recall. Both the ensemble phase and the evaluation phase have used the same dataset.

Retraining doesn't need to be done because the classifiers are already qualified for their optimal shapes and the dataset doesn't change. The final grouping is made up of the top contributing variables, or classifiers. Candidate classifiers with this mindset function as independent variables since they are flexibly adjustable, optimized, and chosen based on predetermined standards. This increases the flexibility to try out different machine learning strategies. The ultimate choice of the optimal phase model is the dependent element of this process. Therefore, the better construction of the top model components is influenced by the inclusion and selection of suitable candidate classifiers.

Evaluation metrics and machine learning classifiers are briefly reviewed in this section. The classification algorithms that are part of this study have been chosen based on their generality, base approach, and wide applicability.

3.1 MACHINE LEARNING ALGORITHMS

3.1.1 Logistic Regression

According to (Armengol E. and De Mantaras, R.L.1998), logistic regression is a generalization of linear regression that is used to estimate binary or multi-class dependant variables when the response variable is discrete. A statistical model called logistic regression models binary dependent variables using sigmoid functions. Because sigmoid (logistic) functions always place any real number in the interval between 0 and 1, they are appropriate for statistical analyses of binary classification problems. Similar in nature to linear regression, which determines the output for a given input by utilizing predetermined weights or coefficients, is logistic regression. The sole distinction is that binary outputs of 0 or 1 are consistently produced by logistic regression. When classifying low-dimensional data with nonlinear bounds, logistic regression is utilized. For the problem of binary classification, as described in (Cao, C., 2012),

3.1.2 Decision Tree

A tree structure model with leaves for class labels and branches for feature conjunctions is used to model decision trees, which can be used as a classification technique. The decision tree technique is one of several supervised learning algorithms that are used to address classification and regression issues. Developing a training model that can forecast the target variables using the decision rules that have been learned is the primary objective. Depending on the kind of target variables, there are two sorts of decision tree algorithms: continuous variable decision trees and categorical variable decision trees (Kögel, B. a. 2013).The learning process produces a classification tree, with each leaf representing a single value of the target variable and the split at each node of the tree representing an if-then decision rule...

3.1.3Gradient Boosting

Gradient Boosting is a machine learning method for classification and regression problems that produces an ensemble of weak prediction models from which a prediction model is constructed. This is due to the fact that decision trees frequently Gradient boosted trees are produced by the poor learning process of decision trees, which typically outperforms random forest. It also employs wise step techniques, similar to other boosting models, and optimizes a random differentiable loss function.

3.1.4 Random Forest

For classification, regression, and other tasks, Random Forest is an ensemble learning technique that builds a large number of decision trees during training and outputs the class that is the mean/average prediction (regression) or the mode of the classes (classification) of the individual trees (Tin Kam Ho 1995, Breiman, L., 2001). The mean or average prediction made by each individual tree is returned for regression tasks. The tendency of decision trees to overfit to their training set is compensated for by random decision forests. Although they are less accurate than gradient-boosted trees, random forests still perform better than choice trees in most cases. Their performance, however, may be impacted by the peculiarities of the data.

3.1.5 Light Gradient Boosting Machine

Another algorithm created by Microsoft Research Asia (Xie Y,2018) made use of the GBDT architecture was called LightGBM. It attempts to increase computing efficiency in order to address large data prediction issues more successfully. A tree-structured learning algorithm is employed. Unlike previous tree-based learning algorithms, LightGBM expands the tree in a vertical manner. It can also be said that it grows additional algorithms at the level of the tree and at the leaf level. This algorithm's benefits include fast processing speed, GPU learning, and low memory requirements. (Ke, G 2017)

3.1.6 Categorical Boosting

Every new tree constructed by CatBoost and other common gradient boosting implementations is meant to approximate the gradients of the existing model (Dorogush, A.V 2018).However, the issue of biased pointwise gradient estimations leads to overfitting in all classical boosting techniques. The same data points that were used to build the present model are used to estimate the gradients used at each stage. This causes overfitting because it shifts the distribution of predicted gradients in any feature space domain relative to the true distribution of gradients in that domain. compared to other gradient boosting algorithm that, on a number of well-known publicly accessible datasets, Catboost performs better in terms of quality than current publicly available gradient boosting implementations while handling categorical features with ease(A. Gulin, G. Gusev, 2021). When applied to ensembles of comparable sizes, they outperform existing gradient boosting libraries by a large margin.

3.2 DATA PREPARATION

The training dataset and testing dataset are prepared based on the use of stratified K-Fold validation method as they are used and have shown success in the area of Imbalance Data Classification on medical data (Khushi, M, 2021).

In this study, we separated the dataset into training and testing sets, the former being used to train the model and the latter to test it, in order to assess the f1 score of our model using an error metric. Additionally, the f1 score obtained for a particular test set may differ significantly from acquired for a different person. Furthermore, as the accuracy attained for one test set may differ from that of another, it could result in inaccurate accuracy for a set (Khushi, M, 2021). Consequently, we used the stratified K-fold cross-validation approach, which takes the data and divides it into folds, with each fold serving as a testing set at some time.

After using the cross-validation technique to reduce biases, train/test splits are used to evaluate the performance of classifiers and machine learning models. For this study, a 10-k fold was employed since a larger K indicates less bias toward overestimating the true anticipated error but also a higher variance and longer running time. A version of K-Fold (here, K=10) that yields stratified folds is called a stratified K-fold cross-validation object. By keeping the percentage of samples for each class constant, the folds are created. To further divide data into train and test sets, it offers train/test indices.

The above-mentioned data preparation variants are used for all used machine learning algorithms. The feature vector contains 41 basic features related to data of a particular patient. As a patient visitation would be different from that of another patient. A classification feature that defines if a person has diabetes or not.

3.3 EVALUATION METRICS

PRECISION

Out of all the samples that are expected to be positive, this is the actual number of positive samples. It is also known as the ratio of the number of positive samples that really belong to Class 1 to the number of samples that were anticipated for Class 1 (Rohit Kundu, 2022). It solely calculates the false positive rate. A false positive is regarded as a more serious error in certain fields, such spam detection (missing a crucial email is usually more destructive than inadvertently deleting junk that passed the filter). In an unbalanced classification problem with two classes, precision is calculated by dividing the total number of genuine positive values by the total number of false positives. Thus, it tries to respond to the query of what percentage of identifications were accurate.

$$Precsion = \frac{True\ Positive}{True\ Postive + False\ Positive}$$

RECALL

Recall is primarily concerned with the percentage of correctly detected positives. In (Rohit Kundu, 2022) Recall compares false negatives to real positives, and is the antithesis of precision. False negative results are important, especially to avoid illness diagnosis and other safety forecasts. In a two-class unbalanced classification problem, it is expressed as the number of true positives divided by the total number of true positives and false negatives. Thus, it provides an answer to the following query: What percentage of real positives were accurately identified?

$$Recall = \frac{True\ Positive}{True\ Postive + False\ Negative}$$

F1 SCORE

The weighted average of Precision and Recall is known as the F1 Score. It is employed to strike a compromise between the two goals of high recall and high precision (Rohit Kundu, 2022). The F1 Score is a test's accuracy measured by the harmonic mean between recall and precision. [0, 1] is the F1 score range. It demonstrates both the robustness and accuracy of our classification (the number of cases in which it is correctly classified). Therefore, the precision and recall weighted average can be represented as

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall\ + Precision}$$

F1 Score with "Weighted" Average: it calculates the F1 score for each class individually and then takes a weighted average, where the weight is based on the support (number of instances) of each class. This means that classes with more instances have a greater impact on the average than classes with fewer instances. It addresses the issue of class imbalance by giving more weight to the classes with more data.

3.4 HYPERPARAMETER SEARCH

Optuna is being used in this research with the aim of introducing new design criteria for the next generation hyperparameter optimization library (Akiba, T., 2019). Optuna is superior to other hyperparameter search methods due to the following criteria:

(1) define-by-run API that allows users to construct the parameter search space dynamically,

(2) efficient implementation of both searching and pruning strategies

(3) easy-to-setup, versatile architecture that can be deployed for various purposes, ranging from scalable distributed computing to lightweight experiment conducted via interactive interface

3.5 MODEL INTERPRETABILITY USING SHAP ANALYSIS

The capacity of tree-based algorithms to offer details on the choices taken in relation to forecasts is a significant benefit. Weights that are allocated to the features as a result of the learning process are used to offer this information. This help to look at how each characteristic was rated by the prediction models since the weight value assigned to a particular feature indicates the feature's importance as judged by the prediction model (Danso et.al 2021).

In order to investigate the relationships between the features, we also used the SHapley Additive exPlanation (SHAP) algorithm (Vavilala, et al., 2018). In a nutshell, the algorithm draws inspiration from game theory, wherein the relationship between features is viewed as a "team" of features, each of which is a member of the team that determines the overall risk. A collection of anticipated values generated by the prediction model is recorded by an instance of the feature interaction. The SHAP algorithm uses these values as input to produce a different set of values known as "impact values." In order to ascertain the likelihood of danger and the function of each component individually, the SHAP values offer a dynamic perspective of the interactions between the features. Additionally, the SHAP method provides the ability to contrast a predicted risk probability for each individual with a baseline prediction, which is the average predicted probability referred to as the "base value."(Parsa, et.al 2020)

4.0 RESULT

4.1 Baseline Model Result

Baseline Model Default Parameter

S/N	MODEL NAME	DEFAULT PARAMETER
1	Logistic Regression ('LR'):	Regularization: L2 (Ridge), Regularization strength (C): 1.0 Solver: 'lbfgs'
2	Random Forest Classifier ('RF')	Number of Estimators (Trees): 100, Criterion: 'gini' Max Depth: None, Min Samples Split: 2, Min Samples Leaf: 1, Max Features: 'auto' (sqrt(n_features))
3	Gradient Boosting Classifier ('GBM'):	Number of Estimators (Trees): 100, Learning Rate: 0.1 Max Depth: 3, Min Samples Split: 2, Min Samples Leaf: 1 Max Features: None (use all features)
4	Extra Trees Classifier ('ET'):	Number of Estimators (Trees): 100, Criterion: 'gini', Max Depth: None, Min Samples Split: 2, Min Samples Leaf: 1 Max Features: 'auto' (sqrt(n_features))
5	XGBClassifier	Number of Estimators (Trees): 100, Learning Rate: 0.3 Max Depth: 6,Min Child Weight: 1, Gamma: 0, Subsample: 1 Colsample by Tree: 1, Colsample by Level: 1, Lambda (L2 Regularization): 1, Alpha (L1 Regularization): 0, Scale Pos Weight: 1, Objective: 'binary:logistic' (for classification)
6	CatBoostClassifier	Number of Estimators (Trees): 1000, Learning Rate: 0.03 Depth: 6, Loss Function: 'Logloss' (cross-entropy) Regularization: L2 , Border Count: 254 Minimum Child Weight: 1.0, Feature Preprocessing: '0.0' , Auto-Scale: 'true' Random Seed: None , Custom Metric: None
7	LGBMClassifier ('LG')	Number of Estimators (Trees): 100, Learning Rate: 0.1, Max Depth: -1 (unlimited), Min Child Samples: 20. Min Child Weight: 0.001, Subsample: 1.0, Colsample by Tree: 1.0 Colsample by Level: 1.0, Num Leaves: 31

BASELINE MODEL RESULT

S/N	MODEL	MODEL F1 SCORE	WEIGHTED F1
1	LOGISTIC REGRESSION	0.0255	0.938
2	RANDOM FOREST	0.0185	0.939
3	GRADIENT BOOSTING MACHINE	0.0245	0.938
4	EXTRA TREES	0.0190	0.939
5	XGBOOST	0.0584	0.933

6	LIGHT GBM	0.0277	0.939
7	CATEGORICAL BOOSTING	0.0473	0.936

4.2 Feature Engineering Result

FEATURE ENGINEERING MODEL RESULT

S/N	MODEL	MODEL F1 SCORE	WEIGHTED F1
1	LOGISTIC REGRESSION	0.0000	0.939
2	RANDOM FOREST	0.0257	0.938
3	GRADIENT BOOSTING MACHINE	0.0239	0.938
4	EXTRA TREES	0.0255	0.938
5	XGBOOST	0.0521	0.933
6	LIGHT GBM	0.0227	0.938
7	CATEGORICAL BOOSTING	0.0346	0.936

4.3 Imputation Result

IMPUTATION MODEL RESULT

The missing values of all the columns where filled using various imputation methods i.e weight was filled with the mean

S/N	MODEL	MODEL F1 SCORE	WEIGHTED F1
1	LOGISTIC REGRESSION	0.0000	0.940
2	RANDOM FOREST	0.0120	0.939
3	GRADIENT BOOSTING MACHINE	0.0226	0.938
4	EXTRA TREES	0.0108	0.938
5	XGBOOST	0.0461	0.927
6	LIGHT GBM	0.0224	0.937
7	CATEGORICAL BOOSTING	0.0162	0.937

4.4 Smote Analysis Result

SMOTE ANALYSIS MODEL RESULT

The Top 3 Model from the imputation model result was selected to carry out Smote analysis using Imblearn for handling imbalance target class, two different method was used to generate two tables which include

Smote Analysis was applied on the train data after splitting the data into it Fold of train and test using Stratified KFold Technique

S/N	MODEL	MODEL F1 SCORE	WEIGHTED F1
1	LOGISTIC REGRESSION	0.185	0.499
2	RANDOM FOREST	0.050	0.932
3	EXTRA TREE	0.058	0.930

Smote Analysis was applied on the whole data before performing Stratified KFold Technique for Splitting into train and test

S/N	MODEL	MODEL F1 SCORE	WEIGHTED F1
1	LOGISTIC REGRESSION	0.185	0.514
2	RANDOM FOREST	0.050	0.936
3	EXTRA TREE	0.058	0.939

4.5 Hyperparameter Tuning Result

OPTIMISED MODEL PARAMETERS

A Study of 50 trials was conducted using optuna for tuning each of the parameters while an n_estimator/iterations of 1500 was used for both model, two model was selected from the two different smote analysis

S/N	MODEL	OPTIMISED PARAMETER
1	RANDOM FOREST	{'max_depth': 21, 'min_samples_split': 0.3571908227294197 , 'min_samples_leaf': 0.4248541076699189, 'max_features': ' log2'}
2	EXTRA TREES	{'max_depth': 30, 'min_samples_split': 0.35757621217023305, 'min_samples_leaf': 0.49783107457055603, 'max_features': 'log2'}

S/N	MODEL	MODEL F1 SCORE	WEIGHTED F1
1	RANDOM FOREST	0.022	0.940
2	EXTRA TREES	0.000	0.940

5.0 CONCLUSION

By using various machine learning approaches to such data, my goal is to aid in the early detection of diabetes in patients. Starting with a basic baseline model that was run on raw data, the project flow was established. The same classifier used in the baseline was utilized to evaluate every classifier before doing feature engineering on the data. This is followed by an exploratory data analysis utilizing statistical approaches and data transformation. All the candidate classifiers were trained and tested to manage target class imbalance when dividing the data into train and test folds using stratified k-folding (k=10) cross-validation.

The next step involved training a queue of classifiers, similar to the one used for the baseline model, and using various imputation techniques to fill in the missing data for every column. The top three models, Random Forest, Logistic Regression, and Extra Tree Classifier, were chosen based on the F1 Weighted Score. These models will be used to further analyze the data, which will first undergo a smote analysis to oversample the data and obtain a balanced distribution. After the best three models have been trained and the oversampling process is complete, the top model (Random Forest) will be chosen, and its parameter will be optimized using Optuna before training on the data.

The results indicate that there was a 0.001% increase in the optimized result from the baseline model, and the statistical analysis performed on the data resulted in a lower result, suggesting that the model is not appropriate for the given data.

6.0 REFERENCE

1. Akiba, T., Sano, S., Yanase, T., Ohta, T. and Koyama, M., 2019, July. Optuna: A next-generation hyperparameter optimization framework. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 2623-2631).

2. Armengol, E. and De Mantaras, R.L., 1998. Machine learning from examples: Inductive and Lazy methods. Data & Knowledge Engineering, 25(1-2), pp.99-123.

3. A. Gulin, G. Gusev, L. Ostroumova Prokhorenkova, A. V. Dorogush, and A. Vorobev. Catboost:unbiased boosting with categorical features. arXiv preprint arXiv:1706.09516, 2017.

4. Bai, B. M., N.Mangathayaru and Rani, B. P. (2015) 'An Approach to Find Missing Values in Medical Datasets', Proceedings of the The International Conference on Engineering & MIS 2015. Available at: https://api.semanticscholar.org/CorpusID:9775851

5. Bentéjac, C., Csörgő, A. and Martínez-Muñoz, G., 2021. A comparative analysis of gradient boosting algorithms. Artificial Intelligence Review, 54, pp.1937-1967.

6. Breiman, L., 2001. Random forests. Machine learning, 45, pp.5-32.

7. Bo He, Kuang-i Shu and Heng Zhang, Machine Learning and Data Mining in Diabetes Diagnosis and Treatment, IOP Conference Series: Materials Science and Engineering, Volume 490, Issue 4, IOP Conf. Series: Materials Science and Engineering 490 (2019) 042049 IOP doi:10.1088/1757899X/490/4/042049

8. Bronshtein Adi, "Pandas" Python Library'., (2019), Towards Data Science: [Online] [Accessed https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library- f1b678f34673.

9. Cao, C., 2012. Sports data mining technology used in basketball outcome prediction.

10. Clore,John, Cios,Krzysztof, DeShazo,Jon, and Strack,Beata. (2014). Diabetes 130-US hospitals for years 1999-2008. UCI Machine Learning Repository. https://doi.org/10.24432/C5230J.

11. Danso, S.O., Zeng, Z., Muniz-Terrera, G. and Ritchie, C.W., 2021. Developing an explainable machine learning-based personalised dementia risk prediction model: A

transfer learning approach with ensemble learning algorithms. Frontiers in big Data, 4, p.613047.

12. de Winter, J. C. F., Gosling, S. D., & Potter, J. (2016). Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. Psychological Methods, 21(3), 273–290. https://doi.org/10.1037/met0000079

13. Di Leo, G., & Sardanelli, F. (2020). Statistical significance: p value, 0.05 threshold, and applications to radiomics—reasons for a conservative approach. Eur Radiol Exp, 4, 18. https://doi.org/10.1186/s41747-020-0145-y

14. Dorogush, A.V., Ershov, V. and Gulin, A., 2018. CatBoost: gradient boosting with categorical features support. arXiv preprint arXiv:1810.11363.

15. GeeksforGeeks, 2021, NumPy in Python.. [Online] [Accessed https://www.geeksforgeeks.org/numpy-in-python-set-1-introduction.

16. Gu, Q., Cai, Z., Zhu, L., & Huang, B. (2008). Data Mining on Imbalanced Data Sets. In 2008 International Conference on Advanced Computer Theory and Engineering (pp. 1020-1024). doi:10.1109/ICACTE.2008.26

17. Hackeling, G., 2017. Mastering Machine Learning with scikit-learn. Packt Publishing Ltd

18. Hancock, J., & Khoshgoftaar, T. M. (2020). Performance of CatBoost and XGBoost in Medicare Fraud Detection. 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA). doi:10.1109/icmla51294.2020.00095

19. J. Brownlee (2021), “Scikit-Learn: A Python Machine Learning Library.”, Machine Learning Mastery.[Online],[Accessed https://machinelearningmastery.com/a-gentle-introduction-to- scikit-learn-a-python-machine-learning-library.

20. Karp, A.H., 1998. Using logistic regression to predict customer retention. In Proceedings of the Eleventh Northeast SAS Users Group Conference. http://www. lexjansen. com/nesug/nesug98/solu/p095. pdf (Vol. 15).

21. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Advances in Neural Information Processing Systems, pp. 3146–3154 (2017)

22. Ќegl, B. a. (2013). The return of AdaBoost.MH: multi-class Hamming trees. CoRR, abs/1312.6086.https://www.semanticscholar.org/paper/The-return-of-

AdaBoost.MH%3A-multi-class-HammingK%C3%A9gl/a37c1df39575fd59d8b3b4697da2de486c71ab3.

23. Khushi, M., Shaukat, K., Alam, T.M., Hameed, I.A., Uddin, S., Luo, S., Yang, X. and Reyes, M.C., 2021. A comparative performance analysis of data resampling methods on imbalance medical data. IEEE Access, 9, pp.109960-109975.

24. Machado, M.R., Karray, S. and de Sousa, I.T., 2019, August. LightGBM: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry. In 2019 14th International Conference on Computer Science & Education (ICCSE) (pp. 1111-1116). IEEE.

25. Moura, A. F. D., Pinho, C. M. D. A., Napolitano, D. M. R., Martins, F. S. and Fornari Junior, J. C. F. D. B. (2020) 'Optimization of operational costs of Call centers employing classification techniques.' Research, Society and Development, 9(11) p. e86691110491.

26. National Institute of Diabetes and Digestive and Kidney Disease About Diabetes Statistics Accessed at https://www.niddk.nih.gov/health-information/health-statistics/diabetes-statistics on Sept 17, 2023

27. O. Adepoju, J. Wosowei, S. lawte and H. Jaiman, "Comparative Evaluation of Credit Card Fraud Detection Using Machine LearningTechniques," 2019 Global Conference for Advancement in Technology (GCAT), Bangalore, India, 2019, pp. 1-6, doi:10.1109/GCAT47503.2019.8978372.

28. Parsa, A.B., Movahedi, A., Taghipour, H., Derrible, S. and Mohammadian, A.K., 2020. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. *Accident Analysis & Prevention*, *136*, p.105405.

29. Python package training parameters, 2020. [Online]. Available: https://catboost.ai/docs/concepts/python-reference parameters-list. html#python-reference parameters-list.

30. Rodriguez-Sanchez B, Aranda-Reneo I, Oliva-Moreno J, Lopez-Bastida J. Assessing the Effect of Including Social Costs in Economic Evaluations of Diabetes-Related Interventions: A Systematic Review. Clinicoecon Outcomes Res. 2021;13:307-334 https://doi.org/10.2147/CEOR.S301589

31. Roglic, Gojka. WHO Global report on diabetes: A summary. International Journal of Noncommunicable Diseases 1(1):p 3-8, Apr–Jun 2016. DOI: 10.4103/2468-8827.184853

32. Rohit Kundu (2022), F1 Score in Machine Learning: Intro & Calculation. V7: [Online] [Accessed] https://www.v7labs.com/blog/f1-score-guide.

33. Romi S. Wahono, N. Suryana, Sabrina Ahmad, A Comparison Framework of Classification Models for Software Defect Prediction, October 2014, Journal of Computational and Theoretical Nanoscience 20 (10-12):1945-1950, DOI: 10.1166/asl.2014.5640.

34. Shah, S. Gala and N. Patil, "ModBoost for unbiased classification," 2014 International Conference on Data Mining and Intelligent Computing (ICDMIC), Delhi, India, 2014, pp. 1-5, doi: 10.1109/ICDMIC.2014.6954252.

35. Strack, B., DeShazo, J. P., Gennings, C., Olmo, J. L., Ventura, S., Cios, K. J., & Clore, J. N. (2014). Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records. BioMed Research International, 2014, Article ID 781670

36. Tin Kam Ho, "Random decision forests," Proceedings of 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 1995, pp. 278-282 vol.1, doi: 10.1109/ICDAR.1995.598994.

37. Vavilala, M. S., Lundberg, S. M., Nair, B., Horibe, M., Eisses, M. J., Adams, T., et al. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. Nat. Biomed. Eng. 2, 749–760. doi: 10.1038/s41551-018-0304-0

38. Virtanen, P., Gommers, R., Oliphant, T.E. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat Methods* **17**, 261–272 (2020). https://doi.org/10.1038/s41592-019-0686-2

39. Xie, Y., et al.: Evaluation of machine learning methods for formation lithology identification: a comparison of tuning processes and model performances. J. Pet. Sci. Eng. 160, 182–193 (2018)