Data Mining Project Proposal: Problem

Identifying health risk factors and improving popular machine learning algorithms to predict diabetes in US citizens.

SUE-ELLEN BERNADINA *s1035357*
NATALIA KARDAMI *s1034194*

Diabetes is a severe chronic degenerative disease that affects approximately 400 million people worldwide (Diabetes: Advances in Diagnosis and Treatment). Machine Learning has been a great tool in helping not only diagnose individuals but also identify common risk factors to help disease prevention. In this paper we test performance on diabetes diagnosis using three common classifiers, Naive Bayes, Random Forest and Support Vector Machines, tune them for a diagnostic dataset. We attempt to improve each one by interpreting our knowledge on the nature of the dataset or by boosting pre-existing methods. An ensemble method is employed to combine all the classifiers into one. We found that the best performance was by the Majority Voting Ensemble (accuracy ≈ 0.85) and the weakest were the Naive Bayes classifiers with an average performance of accuracy ≈ 0.69.

Contents

1 Introduction

Diabetes is a severe chronic degenerative disease that affects approximately 400 million people worldwide (David M. Nathan, 2015). Diabetes is a metabolic disease that causes abnormally high levels of blood sugar due to either the inability to produce, or use insulin. Insulin is a hormone that regulates the amount of glucose (sugar) in the body. There are many types of diabetes, with the most prevalent being Type 1, Type 2 and gestational diabetes. Those with Type 1 diabetes have an immune system that breaks down insulin, Type 2 diabetics either cannot produce or cannot react to insulin and gestational diabetes which occurs during pregnancy(Khan et al., 2019). There are two combined factors that can cause diabetes. One is a genetic predisposition, and two is environmental changes that trigger it (American Diabetes Association, 2023). There exists a condition that is called prediabetes, which is diagnosed when blood sugar levels are abnormal, but not enough to be classified as diabetes Type 2. It is crucial to note that prediabetes is reversible (CDC, 2022).

1.1 Implications

Living with diabetes is a challenge from multiple angles. Treatment includes necessary insulin injections, strict dieting Diabetes reduces life expectancy and causes several health problems including cardiovascular, vision impairment and organ failure, amongst others. Diabetes also has a strong impact on mental health, since it can be a life-changing diagnosis. Lastly, insulin is a notoriously expensive and hard-to-make resource (Dall et al., 2019). In certain regions, like the US, diabetic patients are financially crippled in order to pay for necessary medication in order to survive. That is why early diagnosis and prevention are crucial to manage this worldwide epidemic.
As supported by (Saydah et al., 2004), identifying cases of diagnosed diabetes and data analysis is valuable for trend tracking in order to direct the efforts to alleviate the personal and socioeconomic issues surrounding this disease. This leads us to our motivation for this project.

1.2 Project Goal and Motivations

Our purpose was to implement a smart agent that could play the game The process of analysing health data has undergone major advancements in the last few years. From manually extracting conclusions, using simple statistical methods that can be incredibly computationally expensive for big datasets, the latest trend in predictive diagnostics is Data Mining algorithms. The premise is to train and fit a model on known data, such as old patient records, and use it to label novel data. This significantly increases the amount of both man- and computational power needed to go through massive databases. In this paper we attempt to:

1. Identify risk factors that contribute to diabetes
2. Implement 3 different models to classify a diabetes diagnosis based on health data
3. Improve on each algorithm to enhance performance and accuracy
4. Find a multi-algorithm approach

We hope to answer the following questions:

1. What indicators are stronger when it comes to a diabetic patient?
2. Which algorithm is the best for diagnosing diabetes specifically?
3. Can there be any improvements to the algorithms that are already used?

2 Methods

2.1 Software

We used Jupyter Lab notebooks with Python v3 for processing and visualising our data, and LaTeX editor Overleaf to produce our report. Our main tool was python package sklearn, one of the most popular resources for machine learning algorithms.

2.2 Dataset

The dataset we chose was retrieved from Kaggle (2022), named "Diabetes Health Indicators Dataset". The original data was retrieved from CDC's (2015) Behavioural Risk Factor Surveillance System (BRFSS). Kaggle author provides 3 versions of this dataset. A three-class unbalanced dataset of 253,680 survey responses where 0= no diabetes, 1 = prediabetes, 2= diabetes, a binary balanced dataset of 70,692 survey responses where groups are evenly split into 0= no diabetes or 1= prediabetes of diabetes. Lastly, the dataset we plan to use, a binary unbalanced dataset of 253,680 survey responses where 0= no diabetes or 1= prediabetes of diabetes. We chose the binary unbalanced dataset because we believe it is crucial to select a database that is closer to the real world statistics, and we want to treat the unbalanced data with our own methods.

2.2.1 Attributes

1. 'Diabetes_binary', 0= no diabetes or 1= prediabetes of diabetes.
2. 'HighBP', 0 = no high BP 1 = high BP
3. 'HighChol', 0 = no high cholesterol 1 = high cholesterol
4. 'CholCheck', 0 = no cholesterol check in 5 years 1 = yes cholesterol check in 5 years
5. 'BMI', Body Mass Index, float.
6. 'Smoker', Have you smoked at least 100 cigarettes in your entire life? [Note: 5 packs = 100 cigarettes] 0 = no 1 = yes
7. 'Stroke', (Ever told) you had a stroke. 0 = no 1 = yes
8. 'HeartDiseaseorAttack', coronary heart disease (CHD) or myocardial infarction (MI) 0 = no 1 = yes
9. 'PhysActivity', physical activity in past 30 days - not including job 0 = no 1 = yes
10. 'Fruits', Consume Fruit 1 or more times per day 0 = no 1 = yes
11. 'Veggies', Consume Vegetables 1 or more times per day 0 = no 1 = yes
12. 'HvyAlcoholConsump', (adult men ¿=14 drinks per week and adult women¿=7 drinks per week) 0 = no 1 = yes
13. 'AnyHealthcare', Have any kind of health care coverage, including health insurance, prepaid plans such as HMO, etc. 0 = no 1 = yes
14. 'NoDocbcCost', Was there a time in the past 12 months when you needed to see a doctor but could not because of cost? 0 = no 1 = yes

15. 'GenHlth', Would you say that in general your health is: scale 1-5 1 = excellent 2 = very good 3 = good 4 = fair 5 = poor
16. 'MentHlth', days of poor mental health scale 1-30 days
17. 'PhysHlth', physical illness or injury days in past 30 days scale 1-30
18. 'DiffWalk', Do you have serious difficulty walking or climbing stairs? 0 = no 1 = yes
19. 'Sex', 0 = female 1 = male
20. 'Age', 13-level age category, 1 = 18-24 9 = 60-64 13 = 80 or older
21. 'Education', Education level scale 1-6 1 = Never attended school or only kindergarten 2= elementary etc.
22. 'Income' Income scale 1-8 1 = less than $10,000 5 = less than $35,000 8 = $75,000 or more

2.2.2 Reliability

The Center of Disease Control's online database, BRFSS, is a system of health telephone surveys conducted in the United States. CDC claims it is the "largest continuously conducted health survey system", as they conduct approximately 400,000 adult interviews yearly. (CDC, 2022). It is generally considered reliable, especially when it comes to more specific measurements of "current smoker, blood pressure screening, height, weight, and BMI, and several demographic characteristics" (Holtzman, 2003). Measurements of moderate reliability include: "sedentary lifestyle, intense leisure-time physical activity, and fruit and vegetable consumption". This information will guide us when we select our variables.

2.3 Algorithms

Disease prediction with data mining techniques is a widely employed method in the field nowadays. There is vast bibliography concerning which algorithms are more commonly used, and more importantly which are more effective. Upon research we have decided to use SVM, Random Forest and Naive Bayes (Uddin et al., 2019). First, it is crucial that we pre-process the data accordingly for each algorithm in order to get reliable results. Second we will use them separately and assess their accuracy, filter them through K-fold cross validation and lastly, use a weighted function based on accuracy to yield a prediction based on all three algorithms. Lastly, we will assess the overall performances of the algorithms using suitable techniques such as ROC curves and f1-scores to generate our results and form our conclusions.
Below you will find more details on why we chose each algorithm and how we plan to improve each implementation either by educated preprocessing, or better visualisation techniques.

2.3.1 Naive Bayes

Naive Bayes (NB) is a classification process that in its core uses Bayes' theorem to calculate the probability of a certain event, given prior knowledge. Bayes' theorem works great with disease prediction, because it accounts for the probability of having a disease in the general population. Furthermore, this reflects the shape of our data, as we have a heavily imbalanced dataset, where the positive cases (1= diabetes) are generally

rare in the population. Additionally, according to (Fatima Pasha, 2017), Naive Bayes performs better specifically for diabetes prediction, compared to other diseases. We aim to improve NB performance by smoothing the data, taking the log of probabilities and visualising with heat maps (Tokuc, 2022).

2.3.2 Support Vector Machines

Support Vector Machines (SVM) works by finding a (hyper)plane that can accurately separate the data points. For 2 dimensional data, the hyperplane is a line, for 3 dimensional data the hyperplane is a slice, and so forth. There are multiple planes that can be a solution, that is why the optimal hyperplane is the one that has maximum distance from the classes (Gandhi, 2018). SVMs are versatile, can be applied to multidimensional and non-linear data and are memory efficient (sklearn). We plan to improve SVM by training with K-fold validation, and avoid overfitting by selecting a proper kernel (Han Jiang, 2014)

2.3.3 Random Forest

Random Forest (RF) is an explorative classifier algorithm. RF consists of multiple decision trees, which are randomly generated. The output of an RF is the class that is "voted" by the majority of the decision trees. It scales well in large databases such as ours, and highlights the most important variables of the set (Advantages and limitations of different supervised machine learning algorithms). We will decide between Balanced Random Forest and Weighted Random Forest, both of which are used to combat imbalance in data (Uddin et al., 2019). We aim to improve performance by performing educated feature selection and parameter tuning. (Koehrsen, 2018)(Jalal et al., 2022)

3 Implementation

3.1 Preproccesing

Firstly, we imported the database as a Pandas Dataframe. This made it easy to perform many useful functions on our data. We previewed our data in a table form and observe the column names and rows. We observed some important statistical measures for each attribute such as the mean, the standard deviation, the IQRs and the edge values of each attribute. We then proceeded to clean our data by checking for NaN/missing values and removing duplicate entries. Then we separated the independent attributes, stored them as 'X', and stored our target class column 'Diabetes_Binary' as 'y'.

3.1.1 Exploratory Data Analysis

We then plotted histograms of all of our attributes to visualise their distributions. See the attached .ipynb file for a deeper dive on the distribution of our data. Most importantly we noted a moderate imbalance in our target variable 'y'. We observed a positive (1=(Pre)Diabetes) to negative (0 = No diabetes) ratio of approximately 18%. This means that we are dealing with a moderately imbalanced dataset. This means that we have to either over-sample or under-sample our data to make the cases occur equally, or use proper algorithms that account for such imbalance. In our report we try a combination of those methods.

3.1.2 Correlation measures; Dropping Attributes

After having a clear idea of what our data looks like, we used correlation measures to remove unnecessary attributes. This is because our dataset is very big and computing for all attributes is very computationally intensive, and because models perform better when redundant variables are removed.
Firstly, we checked for attributes that do not contribute significantly towards the class output. We calculated the Pearson Correlation Coefficient (PCC) of each of the attributes with respect to y. A value of -1 implies negative correlation, a value of 0 implies no correlation, and 1, a positive correlation. We set a threshold of a=—0.05—, where any attribute with correlation ¡ a is deemed weak and insignificant to the result. Our analysis revealed that [('Smoker', 0.04550398911142438),('Sex', 0.03272416229206401),('AnyHealthcare', 0.02533133630772611),('NoDocbcCost', 0.020048275618898107), ('Fruits', -0.024805336132517033) and('Veggies', -0.04173376417288313)] can be dropped from the dataframe. The low connection of 'Fruits' and 'Veggies' was a pleasant result, since in our research (see section 2.2.2) these two attributes were not quality measures.
Next, we checked for co-correlated attributes. We did that using a heatmap and setting a threshold of a=—0.4—. Because of the size of our dataset, computation of the heatmap took impossibly long, which is why we chose a stratified sample of our dataset (n=2295). We observed the below correlated feature pairs from the heatmap:
Group 1:
PhysHlth - GenHlth: 0.52
DiffWalk - GenHlth: 0.45
PhysHlth - DiffWalk: 0.47
Group 2:
Income - Education: 0.42
The variables in the groups seem to be co-correlated, which means we can keep only one from each group. We break the ties by selecting the variable with the highest absolute value of PCC as calculated earlier. From Group 1 we kept GenHlth, and from Group 2 we kept Income.

3.2 Models

In this section we discuss the implementation of our models and the decisions we made. We used two versions of the data set. The regular one and an over-sampled one using the SMOTE method (add source). Re-sampling the data with SMOTE allows for the creation of a balanced dataset where 0 and 1 have approximately a 50% chance of occuring, while keeping the same number of observations. In this section we will merely discuss the methods we used, whereas the numerical results will be discussed in section 4. Note that for each of our models we used the `MinMaxScaler()` from sklearn to scale our data, since Naive Bayes does not accept negative values.

3.2.1 Naive Bayes

For Naive Bayes, we imported the `ComplementNB()` function from sklearn. We chose this classifier because it is well suited to imbalanced datasets. We evenly split into test and train sets using stratification, keeping the y class ratios of the samples as representatives of the whole dataset. We used all the default parameters.
Next, we experimented with techniques to test the performance of Naive Bayes on our dataset. We propose a combination of K-Fold validation along with incremental learning. Note that we are aware that incremental learning and validating is a dangerous technique since there are two separate processes running at the same time, however we were curious to see if performance increased. In this case, we performed 10-Fold validation while partially fitting our dataset. Using the `partial_fit()` method, we fitted our dataset in batches, which is more appropriate for datasets of such size.(https://machinelearningmastery.com/cross-validation-for-imbalanced-classification/). Lastly, we also experimented using the balanced (resampled data) with our Complement Naive Bayes model.

3.2.2 Random Forest

A random forest classifier is a meta-estimator which uses a form of incremental learning called bootstrapping.
For the simple implementation, we chose sklearn's `RandomForestClassifier()` with the following parameters: `criterion = "log_loss",max_depth=2,`. We chose `'log_loss'` as a measure because we wanted classifiers which also calculated the probabilities of their predictions. The significance of the adjacent function `model.predict_proba()` will become apparent in section 4. Increasing `'max_depth'` did not improve performance before the algorithm became too slow, therefore we chose the value of 2 as a trade-off. Likewise, we chose n_estimators = 100.
We attempted to improve performance by introducing class significance weighting. We want our algorithm to penalise miss-classifying positive (=1) cases over negative cases. This was achieved through the `class_weight` hyperparameter. We then observed and visualised the feature importances. *(https://mljar.com/blog/feature-importance-in-random-forest*

3.2.3 Support Vector Classification (SVC)

Support Vector Machines are highly effective and memory efficient classifier. We first attempted classifying with a linear kernel. The best method for that is sklearn's `LinearSVC(dual=False)`. We tested perfomance with both balanced and original data. However, we wanted an estimator that could provide probabilities for each prediction and implement incremental learning. This is why we moved to a more powerful classifier. sklearn provides the `SDGClassifier()`, which implements stochastic gradient descent learning on linear classifiers.(sklearn SVM: Separating hyperplane for unbalanced classes). This method implements `partial_fit`, is able to support probabilistic

predictions and provides multiple loss functions. We trained this classifier incrementally, using 10-fold validation, like the complementary Naive Bayes. We slowly added improvements such as class importance weights, inputting the balanced data and experimented with several loss functions.

3.3 Voting classifier

After implementing and improving our 3 main models, we thought, what if we combined the predictive power of all 3 into one powerful meta-estimator? We chose the best version of each of the models and inputted them into a `VotingClassifier()`. sklearn's Voting Classifier is a way to combine similarly performing yet different models, using a voting principle to combine performance. There are two voting principles. 'Hard' (majority) voting works by counting the class output instances of the classifiers and choosing the label most classifiers "elected". On the other hand, "soft" voting calculates the argmax of the sum of the probabilities that the classifiers predicted. We tested both and observed their performances. Our study showed

4 Results

First thing we noted is that using overfitted data for our models was not optimal, since we already used methods that accounted for imbalance. While testing our models, we discovered that using our resampled data only worsened performance in every case.

4.1 Comparison

The best classifier, with an accuracy of 0.8497, is Hard Ensemble. Followed by SVC Linear with an accuracy of 0.8498. The next best classifiers, both with an accuracy of 0.8493, is the linear classifier with SGD training and the Soft Ensemble. The linear classifier does have a higher precision and recall than the Soft Ensemble.
With an equal precision, recall and an equal accuracy of 0.8471, Random Forest and Weighted Random Forest are ranked next. The weakest performers are the Naïve Bayes and Improved Naïve Bayes classifiers, with accuracy of 0.7060 and 0.6817, respectively. The first criteria used when ranking these curves was to compare the precision and recall of all clasifiers. Based on that we could easily rule out Naïve Bayes and Improved Naïve Bayes. The second criteria was to compare the accuracy. Lastly we compared the ROC curves, as not all classifiers have an ROC curve.
For every model, there is a no significant difference between the accuracy of the correct predictions of the test data and the accuracy of the correct predictions of the train data. This is important as this means we successfully prevented overfitting the data on the train set.

4.2 Identifying Health Risks

We chose the Random Forest Classifier to identify the most important predictive features for diabetes. Due to the fact that RF classifiers work by bootstrapping, we believe the best attribute importance can be derived through there. We note that the top 3 features that signify a case of diabetes are High BP, General Health and High Cholesterol.

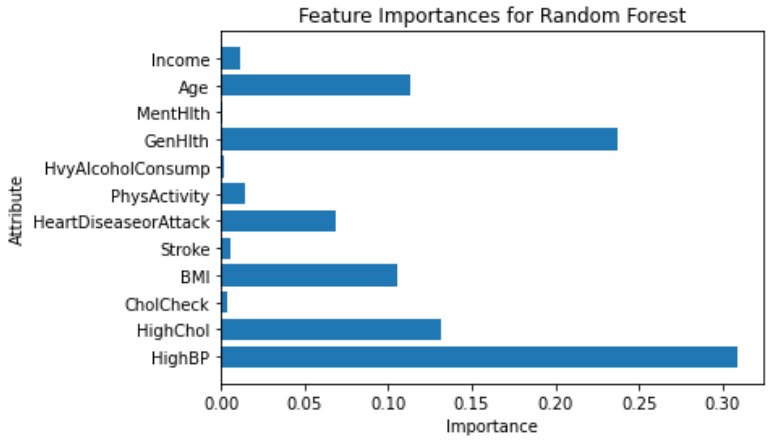

Figure 1: Bar Plot of Feature Importances using Random Forest Classifier

4.3 Jupyter Notebook

For a conclusive look on our methods and implementations, please consult final_diabetes.ipynb, attached to this file. The code is compile-able and flexible so any user can understand and test the different algorithms and parameters we used. Visit our github page here to download the full project and data.

5 Discussion

5.1 Possible Improvements

We believe we missed the mark by using so many techniques and trying to improve in so many ways. This made it difficult to truly compare the algorithms and identify the components that made them better. One example is our use of `partial_fit()` and K-fold cross validation together. In retrospect we believe that investigating whether this combination works is a project by itself and needs further investigation and more detailed analysis. Another mistake we made was that we used the proper accuracy measures (ROC, precision and recall), only after we had fitted our models and we tuned only by accuracy which was a big mistake. Some algorithms gave us a very satisfactory score of 0.85 but in reality we only realistically predicted the 0 cases.

5.2 Future Research

In the future we are interested in comparing different methods for dealing with unbalanced data. Is over/undersampling with regular models better than using natural data with suited methods? Imbalanced data is a very common occurence in the real world and while most ML methods are tailored towards more manageable, even data, we urge scientists to turn their focus on more difficult, but realistic data. Disease prediction and machine learning go hand-in-hand, and we believe that in the future diagnostic algorithms will be the main predictive party in the medical world.

6 Bibliography

- American Diabetes Association. (n.d.). Genetics of diabetes. Genetics of Diabetes — ADA. Retrieved January 9, 2023, from https://diabetes.org/diabetes/genetics-diabetes

- CDC. (2022, July 7). The surprising truth about prediabetes. Centers for Disease Control and Prevention. Retrieved January 9, 2023, from https://www.cdc.gov/diabetes/library/features/truth-about-prediabetes.html

- Dall, T. M., Yang, W., Gillespie, K., Mocarski, M., Byrne, E., Cintina, I., Beronja, K., Semilla, A. P., Iacobucci, W., Hogan, P. F. (2019, April 2). Economic burden of elevated blood glucose levels in 2017: Diagnosed and undiagnosed diabetes, gestational diabetes mellitus, and prediabetes. American Diabetes Association. Retrieved January 9, 2023, from https://diabetesjournals.org/care/article/42/9/1661/36300/The-Economic-Burden-of-Elevated-Blood-Glucose

- David M. Nathan, M. D. (2015, September 8). Advances in diabetes. JAMA. Retrieved January 9, 2023, from https://jamanetwork.com/journals/jama/article-abstract/2434688

- Fatima, M., Pasha, M. (2017, January 24). Survey of machine learning algorithms for disease diagnostic. Journal of Intelligent Learning Systems and Applications. Retrieved January 9, 2023, from https://www.scirp.org/journal/paperinformation.aspx?paperid=73781

- Gandhi, R. (2018, July 5). Support Vector Machine - introduction to machine learning algorithms. Medium. Retrieved January 9, 2023, from https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

- Han, H., Jiang, X. (2014, December 9). Overcome support vector machine diagnosis overfitting. Cancer informatics. Retrieved January 9, 2023, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4264614/

- Holtzman, D. (2003). Analysis and interpretation of data from the U.S. Behavioral Risk Factor Surveillance System (BRFSS). Global Behavioral Risk Factor Surveillance, 35–46. https://doi.org/10.1007/978-1-4615-0071-1_5

- Jalal, N., Mehmood, A., Choi, G. S., Ashraf, I. (2022). A novel improved random forest for text classification using feature ranking and optimal number of trees. Journal of King Saud University - Computer and Information Sciences, 34(6), 2733–2742. https://doi.org/10.1016/j.jksuci.2022.03.012

- Khan, R. M. M., Chua, Z. J. Y., Tan, J. C., Yang, Y., Liao, Z., Zhao, Y. (2019, August 29). From pre-diabetes to diabetes: Diagnosis, treatments and Translational Research. MDPI. Retrieved January 9, 2023, from https://www.mdpi.com/1648-9144/55/9/546

- Koehrsen, W. (2018, January 8). Improving the random forest in python part 1. Medium. Retrieved January 9, 2023, from https://towardsdatascience.com/improving-random-forest-in-python-part-1-893916666cd

- Saydah, S. H., Geiss, L. S., Tierney, E., Benjamin, S. M., Engelgau, M., Brancati, F. (2004). Review of the performance of methods to identify diabetes cases among vital statistics, administrative, and Survey Data. Annals of Epidemiology, 14(7), 507–516. https://doi.org/10.1016/j.annepidem.2003.09.016

- Tokuç, A. A. (2022, November 11). How to improve naive Bayes classification performance? Baeldung on Computer Science. Retrieved January 9, 2023, from https://www.baeldung.com/cs/naive-bayes-classification-performance

- Uddin, S., Khan, A., Hossain, M. E., Moni, M. A. (2019). Comparing different supervised machine learning algorithms for disease prediction. BMC Medical Informatics and Decision Making, 19(1). https://doi.org/10.1186/s12911-019-1004-8