Comparison of Logistic Regression and Naïve Bayes on Diabetes Prediction

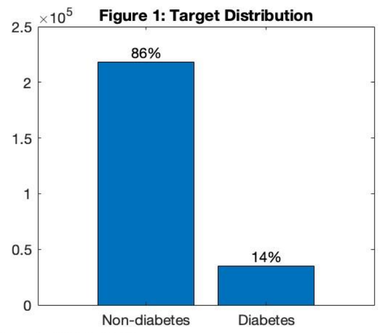
Ling-Yun, Huang

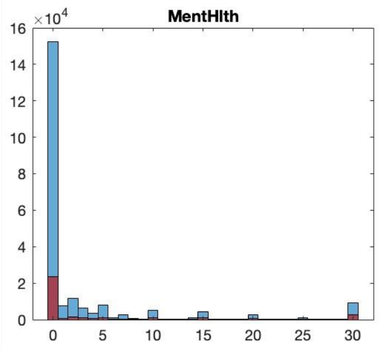
Description and Motivation of the Problem

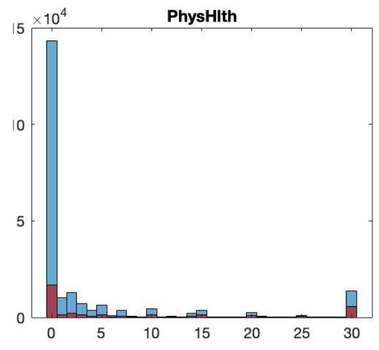
- The main goal is to build two machine learning models, Logistic Regression and Naïve Bayes, and compare the performance of these two models.
- Two models would be built for a binary target dataset, to predict whether an individual has diabetes.
- Similar models were built by Zidian Xie et al. in 2019 with the previous year's data [1].

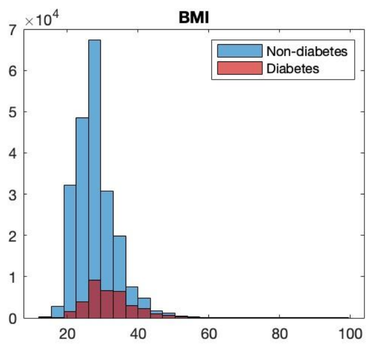
Initial Analysis of the Data Set

- The dataset is the Diabetes Health Indicators Dataset from Kaggle [2], originally from the BRFSS2015 survey conducted by CDC annually.
- The dataset I chose to build the models has already been cleaned on Kaggle. It contains 253,680 rows and 22 columns; with one binary target, 21 features, and no missing value. Among the features, 14 are binary, 3 are numerical, and 4 are ordinal.
- Figure 1 shows that the binary target is imbalanced, with 86% in the non-diabetes class and 14% in the diabetes class.
- The percentages of 14 categorical features within two target groups are shown in Table 1.
- The histograms show the distribution of 3 numerical features within two groups. "MentHlth" and "PhysHlth" would be transformed into two groups based on their distributions, and BMI would be standardised into z-score for the later model building.
- The heatmap shows the Spearman coefficient between all ordinal and numerical features.









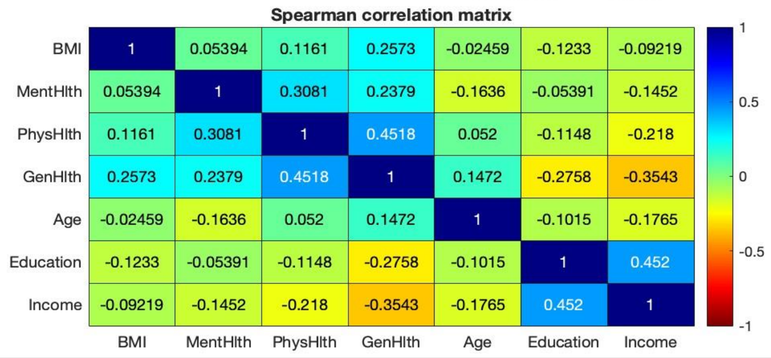


Table 1: Categorical features distribution

	Non-Diabetes		Diabetes	
	N = 218334		N = 35346	
	n	%	n	%
HighBP	82225	37.66%	26604	75.27%
HighChol	83905	38.43%	23686	67.01%
CholCheck	209105	95.77%	35105	99.32%
Smoker	94106	43.10%	18317	51.82%
Stroke	7024	3.22%	3268	9.25%
HeartDiseaseorAttack	16015	7.34%	7878	22.29%
PhysActivity	169633	77.69%	22287	63.05%
Fruits	140205	64.22%	20693	58.54%
Veggies	179105	82.03%	26736	75.64%
HvyAlcoholConsump	13424	6.15%	832	2.35%
AnyHealthcare	207339	94.96%	33924	95.98%
NoDocbcCost	17612	8.07%	3742	10.59%
DiffWalk	29554	13.54%	13121	37.12%
Sex	94771	43.41%	16935	47.91%

Two Machine Learning Methods

Logistic Regression [3]

- Logistic Regression is a widely used classification algorithm that predicts the probability of a binary target dataset. It assumes that the log-odds of the dependent variable are a linear combination of the independent variables.
- It calculates probabilities using a logistic function and optimizing parameters to fit the training data.

Pros

- It is simple, interpretable, and efficient, making it suitable for real-world data.
- It provides ragularisation methods (L1 and L2), offering flexibility to different types of data.

Cons

- Should be careful with the multicollinearity among independent variables.
- It can be sensitive to outliers and impacts the model's performance.
- If the sample size is less than the number of features, it would cause overfitting.

Naïve Bayes [3,4]

- Naïve Bayes is a probabilistic machine learning algorithm commonly used for classification problems. The key assumption is that all features used in the model are independent of each other.
- For the model training, it computes the prior probability for each class and calculates the likelihood with the given features in each class with the training dataset. And computes the product of these likelihoods with given features to calculate the posterior probability, then compares the results within classes.

Pros

- This algorithm is fast, efficient, and easy to perform because of its independent assumption.
- Compared to other classifier methods, it performs better with less training data when the independent assumption holds.

Cons

- Due to its independence assumption, it is hardly found this kind of data in the real world [3]. Need to be careful with the correlations within features.
- If a feature and class combination does not appear in the training data, it would be assigned a zero probability, which might cause issues with unseen data.
- May perform badly with imbalanced datasets, since there might be more misclassed in smaller group [5].

Choice of Parameters and Experimental Results

- In Logistic Regression, I used a cross-validated lasso regularization with 3-fold and chose two sets of parameters with MSE rule and 1SE rule [6].
- In Naïve Bayes, I used the Chi-squared test and ANOVA, with their p-values to determine the importance of features.

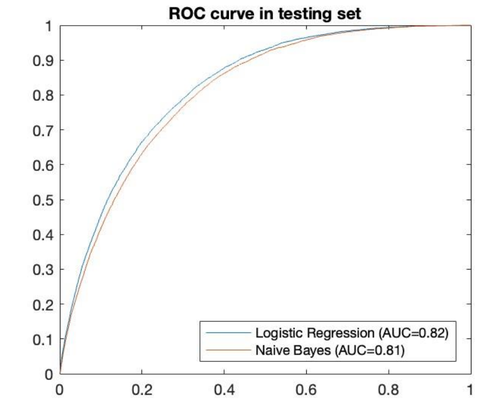
Training and Testing results:

Training results	Logistic Regression		Naïve Bayes	
	Model 1	Model 2	Model 1	Model 2
Number of features	21	15	21	15
Mean Training errors	0.2701	0.2702	0.1843	0.1842
Mean Validated F1-score	0.4412	0.4416	0.4309	0.4302
Mean Validated Precision	0.3096	0.3099	0.3776	0.3777
Mean Validated Recall	0.7674	0.7679	0.5020	0.4997
Mean Validated Accuracy	0.7296	0.7298	0.8156	0.8159
Mean Validated AUC	0.8218	0.8218	0.8086	0.8075
Training Time (s)	3.4819	2.1081	2.6534	1.9106

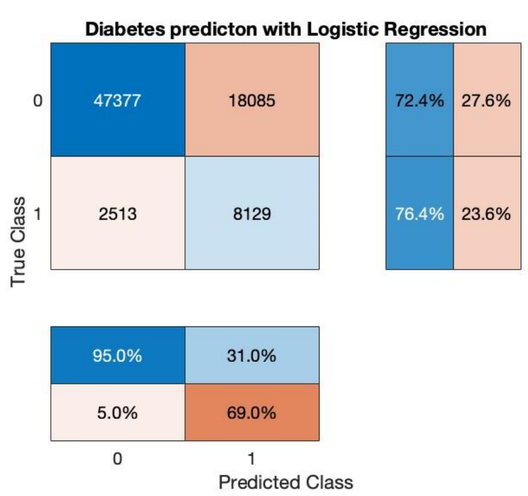
Testing results	Logistic Regression	Naïve Bayes
F1-score	0.4411	0.4326
Precision	0.3101	0.3772
Recall	0.7639	0.5071
Accuracy	0.7293	0.8140
AUC	0.8221	0.8083
Testing time (s)	0.0028	0.0911

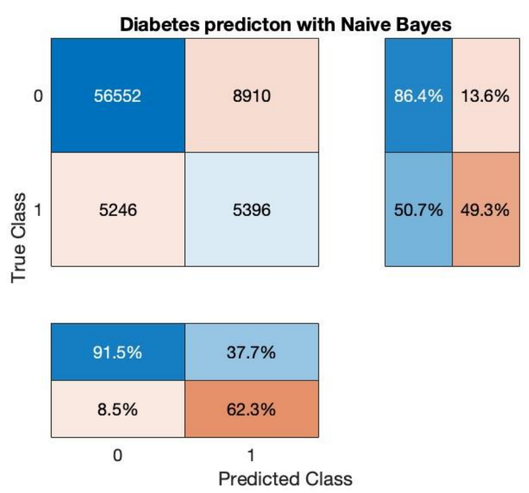
Choice of final models:

- Compared to models in both algorithms with mean validated F1-score. Then choose Model 2 in Logistic Regression and choose Model 1 in Naïve Bayes.
- In Logistic Regression, use the mean of coefficients within 10-fold as a final model. In Naïve Bayes, use the whole training set to calculate the probability of features as a final model.



*The cut-off point in Logistic regression predicted probability was set as 0.14 due to the imbalance of the target.





Hypothesis statement

- Both models are expected to perform similarly to Zidian Xie et al. in 2019 [1]. Accuracy in Logistic Regression is 0.8068 and in Naïve Bayes is 0.7756, and AUC in Logistic Regression is 0.7932 and in Naïve Bayes is 0.7598.
- The Precision for both models may be low due to the target imbalance [5].
- Naïve Bayes is expected to have a lower training and testing time.

Choice of Training and Evaluation Methodology

- Splitting dataset into training and testing sets with 70/30 percentage randomly.
- Applying feature selection approaches:
  - In Logistic Regression, I use Lasso regularisation with MSE and 1SE rule [6] to choose two different sets of features.
  - In Naïve Bayes, I use the chi-squared test and ANOVA with its p-value to choose two different sets of features.
- Using 10-fold cross-validation to train the model with selected features.
- The mean F1-score of validation sets would be used as an evaluation metric.
- Choose a model with a better F1-score in the same algorithm to test the testing dataset and see its performance.
- Compare the two algorithms' best model with test metrics, including F1-score, accuracy, precision, recall, and AUC.

Analysis and Critical Evaluation of Results

Compare the results with Zidian Xie et al. in 2019 [1]:

- The AUCs in both models are higher than the AUCs in the paper. But shows a similar pattern in which Logistic Regression is higher than Naïve Bayes.
- Accuracy shows the other way around. In Naïve Bayes, the accuracy for the model still performs better than the paper's result (0.8083 vs. 0.7756). However, the model's accuracy in Logistic Regression is only 0.7293, compared to 0.8068 in the paper. This may be due to the choice of cutoff point, as I chose 0.14 instead of 0.5.

Compare the two final models with their performance:

- Logistic Regression is performing better than Naïve Bayes in general. Whether it is in F1-score, recall, AUC, or testing time.
- However, the performance of accuracy and precision are better in Naïve Bayes. Again, this might be caused by the choice of cutoff point. Since the lower number in the cutoff would result in true positive, and false positive increased while true negative and false negative decreased.

Training and Testing time:

- In both methods, the training time of Model 2 is lower than Model 1. Due to the fewer features.
- Overall, Naïve Bayes has lower training time compared to Logistic Regression.
- However, Naïve Bayes has a higher testing time than Logistic Regression. This might be because of the Naïve Bayes has more features in the final or the library I use to predict the outcome in Matlab.

The discussion of feature selection:

- Although applying the regulations, both models 1 still have all the features in the model. Two reasons might cause this result. The sample size of the training dataset is big enough, and the features were already the chosen ones in the Kaggle dataset [2].
- There is no clear rule of p-value in Naïve Bayes for feature selection in Model 2. Using the same number of features in Logistic Regression Model 2 is for a fair comparison of Training time and its performance.
- Using the chi-squared test and ANOVA in Naïve Bayes might not be ideal since it only considers the importance of the target. But never consider the correlation between features with independent assumptions in the algorithm, it might influence the model performance.

Models in training dataset:

- In both algorithms, the training results in both models are similar, only slightly different and might be caused randomly. However, this might mean that the extra features do not contribute much to the model. On the other hand, fewer features can also avoid the possibility of overfitting.
- Due to the target imbalance, accuracy performs well in models while other metrics values remain low [5].

Lessons Learned, and Future Work

Lessons learned:

- There are lots of ways to choose the cutoff point in Logistic Regression, choosing different points may dramatically influence some matrices, such as precision or recall.
- Target imbalance will impact the training process and the model's performance. Causing the high misclassification in the smaller group [5].

Future work:

- Deal with the target imbalance by introducing undersampling or SMOTE.
- Might have more insight to include more features from the sources.
- Including more data from different years may reduce the possible bias.
- Trying different feature selection techniques to compare different models.

References:

1. Xie Z, Nikolayeva O, Luo J, Li D. Building Risk Prediction Models for Type 2 Diabetes Using Machine Learning Techniques. *Prev Chronic Dis*. 2019;16:E130. doi:10.5888/pcd16.190109
2. TEBOUL A. Diabetes Health Indicators Dataset. Kaggle. Accessed December 9, 2023. https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset/data
3. Prakash KB. Basic Algorithmic Learning. In: *Data Science Handbook*. Wiley; 2022:173-196. doi:10.1002/9781119858010.ch8
4. Verdhan V. Supervised Learning for Classification Problems. In: Verdhan V, ed. *Supervised Learning with Python: Concepts and Practical Implementation Using Python*. Apress; 2020:117-190. doi:10.1007/978-1-4842-6156-9_3
5. Chawla N V, Japkowicz N, Kotcz A. Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explor Newsl*. 2004;6(1):1-6. doi:10.1145/1007730.1007733
6. Zhou H, Seker V, Downar T. Enhanced Lasso Regularization-Based Self-Adaptive Feature Selection Algorithm for the High-Dimensional Uncertainty Quantification of TREAT Transient Test Modeling. *Nucl Technol*. 2020;206(6):839-861. doi:10.1080/00295450.2020.1746620