CS 412 Final Project Report (Team H)

Adam Beigel, Grace Pnacek, Hari Seelam, Ish Soni

Introduction

Many modern problems nowadays can be solved using machine learning, given the right data. Throughout this project, our team explore machine learning techniques such as Naive Bayes and Logistic Regression to solve problems such as labeling news articles based on various news categories, predicting a rating on a scale of 1-5 of an automatically generated translation, classifying whether or not people are likely for heart disease, and analyzing somebody's risk for diabetes.

Our project consists of four different datasets; two with multinomial classifiers and two with binary classifiers. After pre-processing each dataset, we implemented Naive Bayes and Logistic Regression models from the sklearn library. For the multinomial datasets we also used the SGDClassifier with the logistic loss function as a secondary test.

One of the problems that we attempted to solve was that of automatically labeling news articles in order to make it easier for readers to be served articles that are relevant to their interests without those articles needing to be manually classified. In order to do this, a news category dataset consisting of 202,372 samples of news articles from HuffPost was used. Each sample of this dataset has a category, headline, authors, link, short description, and date published. In terms of categories, the samples are pretty heavily weighted. There are 38 categories, but 16% of the samples have the category “POLITICS”, 8.5% have the category “WELLNESS”, and 8% have the category “ENTERTAINMENT”. The goal of this experiment is to predict the category for each sample based on the words in the headline and short description for the article.

Nowadays, more and more social media sites are using machine translation algorithms such as Google Translate to automatically translate posts for the user. However, the user has no way of knowing how accurate that translation is and therefore is susceptible to potentially misleading translations. One way that our team sought to solve this problem was to classify potentially bad translations in order to help users assess how much they should trust these auto generated translations. This solution uses a translations dataset containing 1,000 samples of English translations generated by Google Translate from German sentences. Each sample translation was rated by 16 respondents on a scale of 1-5 on how "good" or "bad" the translation was, with 1 being the lowest and 5 being the highest.

The diabetes indicators dataset contains 3 different files. The first file, diabetes indicators contains 253,680 survey responses to the CDC’s BRFSS in 2015. It has 3 classes, 0 for no diabetes or only during pregnancy, 1 for prediabetes, and 2 for diabetes. Moreover this file has 21 feature variables and is unbalanced. The second file contains the same details but instead is broken up into only 2 classes (0 for no diabetes and 1 for prediabetes or diabetes). The third file contains a balanced dataset with 70,692 survey responses and has a 50-50 split of respondents with no diabetes (class 1) or with either prediabetes or diabetes (class 2). This dataset has 21 feature variables. For all 3 files, some examples of the features included are high blood pressure, bmi, smoker, stroke, etc. These features help assess what risk factors are most predictive of diabetes and make other predictions. Since early diagnosis can lead to lifestyle changes and better treatment, making predictive models for diabetes risk important tools for public and public health officials and as a result, the goal of this task was to accurately identify if an individual has diabetes or not based on their health factors.

The heart disease dataset contained 319795 samples of Personal Key Indicators of Heart Disease from the Behavioral Risk Factor Surveillance System (BRFSS), which is run by the CDC. It includes 17 different factors (which are the columns) that could affect a person’s chances of developing heart disease, including whether or not the person smokes or drinks, has diabetes, if they have had a stroke before, etc. However, the classes are not balanced in the dataset. Using this dataset, we want to be able to provide a way for anyone to be able to answer the questions that were provided in the BRFSS and be able to know they could be at risk for heart disease.

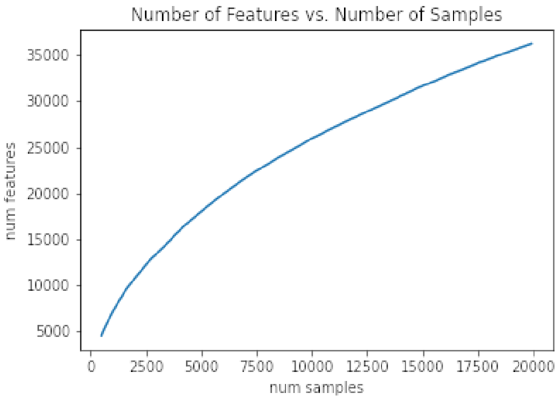
Pre-processing

News Categories

For the news categories dataset, all of the columns except headline and short description were dropped. These two were then combined into one column. Each sample had the numbers and symbols removed and was converted to all lowercase.

The dataset started out with 41 labels. However, some of these were basically the same but phrased differently, like “ARTS”, “ARTS & CULTURE”, and “CULTURE & ARTS”. In these cases, the labels were unified into one descriptor. Lastly, the 19 most distinctive labels were chosen from the dataset based on the most mainstream and consistent news categories and only the samples with those labels were kept.

One of the main issues that needed to be addressed with pre-processing was the number of features that were being created for each sample. The news headlines often refer to people, places, and events that are part of popular culture. This results in many proper nouns being used, so almost every sample in the training set contributed at least 1 unique word to the bag of words. To the right is a graph of the number of features created as the number of samples used to train the model increased. This also contributed heavily to the amount of time it took to train the model and predict labels.

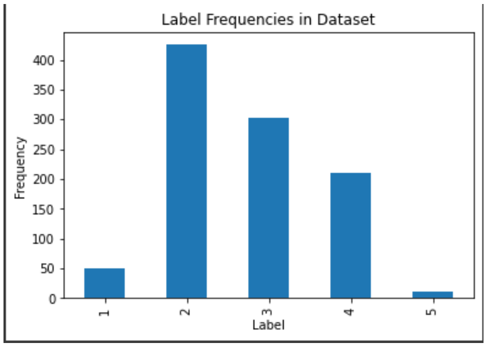


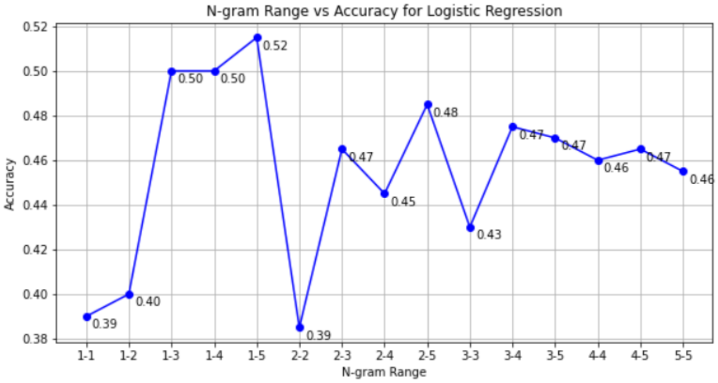
To avoid this problem, we decided to “fix” the bag of words. Instead of creating the bag from the samples themselves, we used the most frequently occurring words in the English language. A range of values for the size of the bag were tested, and a bag size of around 800 was the best compromise between runtime and accuracy in the model. we then created a sklearn CountVectorizer object fitted to the bag of words. Each sample was transformed to represent the frequency of each of those words in the sample’s headline and description.

Translations

Since the goal of classifying "bad" translations vs "good" translations only relies on the translation and not the original text, the original German text was removed during pre-processing in order to prevent convoluted data. With only the English translations left, they were changed to all lowercase and the punctuation was removed. Since the dataset is so small, it would be hard to find patterns in the data from just taking individual words and their frequencies, which is the usual Bag-of-Words representation. To solve this, each sample in the dataset was converted to a sentence of just the Part-of-Speech tags. For example, a sentence such as "The cat is riding the bike" is converted to "DT NN VBZ VBG DT NN". To find a single individual label for each sample, the mode of all 16 respondents was taken and added to its own column. After finding a single label for all samples, the first graph below shows the distribution of labels. Label 2 has the most number of occurrences so this dataset is unbalanced.

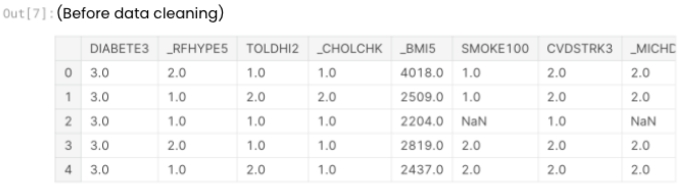
Finally, CountVectorizer was used to find the frequencies of various ranges of n-grams to then send to the off-the-shelf algorithms. The optimal range of n-grams was found by running both algorithms with different ranges and plotting the accuracies. As seen in the second graph below, the range of 1-5 n-grams yielded the highest accuracy and is the one that was used going forward.





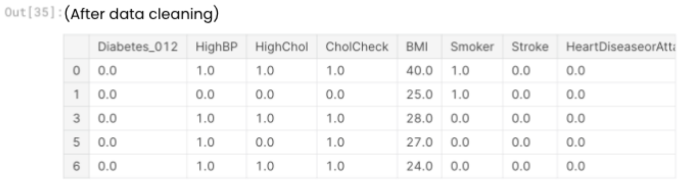
Diabetes Indicators

The diabetes indicator dataset is a cleaned and consolidated version of the BRFSS 2015 dataset on Kaggle. As a result, most of the dataset was already preprocessed. Some other additions made in order to further improve preprocessing were to remove categories such as BMI, General Health, Mental Health, Physical Health, Age, Education, and Income. This was because these categories had a range of values and couldn't be converted into binary classifications (0 and 1).



Out[7]: (Before data cleaning)

	DIABETE3	_RFHYPE5	TOLDHI2	_CHOLCHK	_BMI5	SMOKE100	CVDSTRK3	_MICHD
0	3.0	2.0	1.0	1.0	4018.0	1.0	2.0	2.0
1	3.0	1.0	2.0	2.0	2509.0	1.0	2.0	2.0
2	3.0	1.0	1.0	1.0	2204.0	NaN	1.0	NaN
3	3.0	2.0	1.0	1.0	2819.0	2.0	2.0	2.0
4	3.0	1.0	2.0	1.0	2437.0	2.0	2.0	2.0

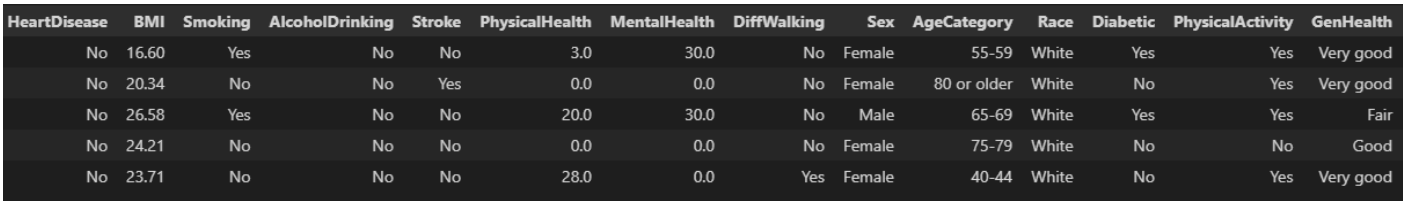


Out[35]: (After data cleaning)

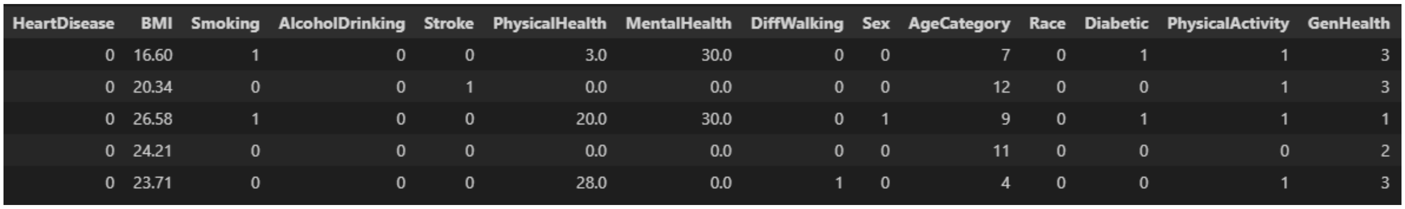
	Diabetes_012	HighBP	HighChol	CholCheck	BMI	Smoker	Stroke	HeartDiseaseorAtt
0	0.0	1.0	1.0	1.0	40.0	1.0	0.0	0.0
1	0.0	0.0	0.0	0.0	25.0	1.0	0.0	0.0
3	0.0	1.0	1.0	1.0	28.0	0.0	0.0	0.0
5	0.0	1.0	0.0	1.0	27.0	0.0	0.0	0.0
6	0.0	1.0	1.0	1.0	24.0	0.0	0.0	0.0

Heart Disease

To preprocess the data, all the non numeric data had to be converted to numeric data, for example the No's to 0's and Yes's to 1's. The Age category and race columns in the dataset were the most affected by this as they had more than two values they could be so they couldn't simply be converted into binary 0 and 1 values as we did with the rest of data. Each category had to be replaced with an increasing numerical number to represent that category. So in the end, the categories in the Age category became numbered from 0-12 and the race category became numbered from 0-5 along with other columns that had contained more than two values.



HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth
No	16.60	Yes	No	No	3.0	30.0	No	Female	55-59	White	Yes	Yes	Very good
No	20.34	No	No	Yes	0.0	0.0	No	Female	80 or older	White	No	Yes	Very good
No	26.58	Yes	No	No	20.0	30.0	No	Male	65-69	White	Yes	Yes	Fair
No	24.21	No	No	No	0.0	0.0	No	Female	75-79	White	No	No	Good
No	23.71	No	No	No	28.0	0.0	Yes	Female	40-44	White	No	Yes	Very good



HeartDisease	BMI	Smoking	AlcoholDrinking	Stroke	PhysicalHealth	MentalHealth	DiffWalking	Sex	AgeCategory	Race	Diabetic	PhysicalActivity	GenHealth
0	16.60	1	0	0	3.0	30.0	0	0	7	0	1	1	3
0	20.34	0	0	1	0.0	0.0	0	0	12	0	0	1	3
0	26.58	1	0	0	20.0	30.0	0	1	9	0	1	1	1
0	24.21	0	0	0	0.0	0.0	0	0	11	0	0	0	2
0	23.71	0	0	0	28.0	0.0	1	0	4	0	0	1	3

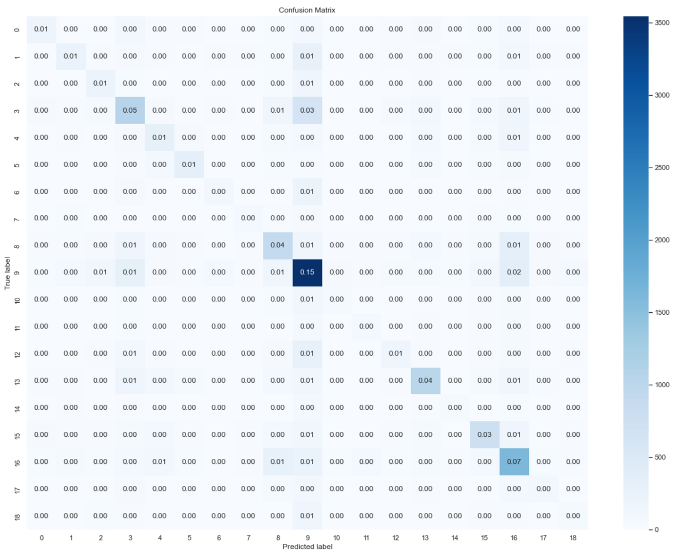
Naive Bayes

We decided as a group to try a Naive Bayes probabilistic model first implemented by using sklearn algorithms. This proved to be a challenge with multinomial classification. The length of time it took to train the model increased exponentially with the number of features for the dataset. This was one of the main reasons for switching from a learned bag of words to a fixed bag with the News Category dataset. We used the Multinomial Naive Bayes function from the sklearn library and calculated the accuracy score, balanced accuracy score, and F1 scores. From the predicted labels we created a confusion matrix.

News Category dataset with 774 features and a train size of 0.8:

- Accuracy Score: 0.466
- Balanced Accuracy Score: 0.346
- Weighted F1 Score: 0.456

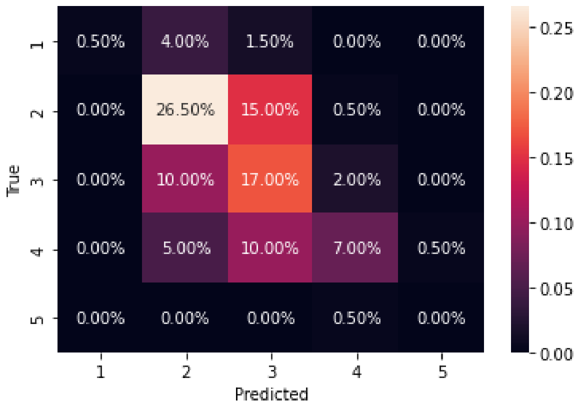
Confusion Matrix:

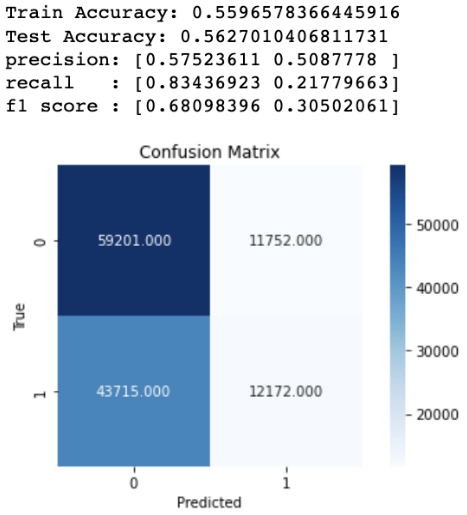


The dark squares don't indicate better accuracy, those are just the labels with the most samples. Those labels are:

- class 9: 'POLITICS'
- class 16: 'WELLNESS'
- class 3: 'ENTERTAINMENT'

For the translations dataset, Multinomial Naive Bayes from sklearn was used. The dataset was split 80-20 using train_test_split. The accuracy of the Multinomial Naive Bayes algorithm was 0.49 with the largest percentage of labels predicted correctly being in labels 2 and 3, which is unsurprising given the high frequency of those labels in the dataset. Much like the news categories dataset, the confusion matrix on the right shows that the labels with the highest frequencies were the ones that were classified correctly the most amount of times.

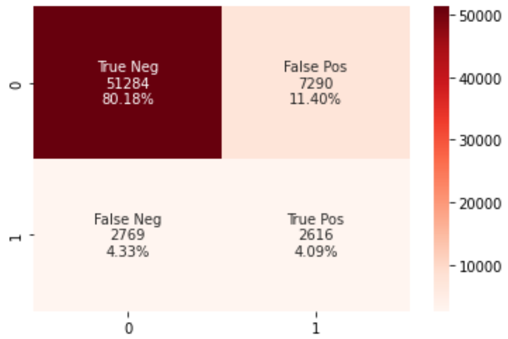


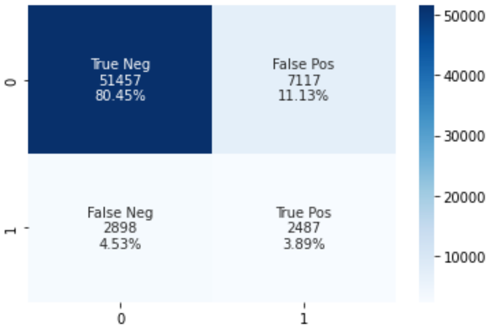


```
Train Accuracy: 0.5596578366445916
Test Accuracy: 0.5627010406811731
precision: [0.57523611 0.5087778 ]
recall    : [0.83436923 0.21779663]
f1 score : [0.68098396 0.30502061]
```

For the Diabetes Indicators dataset, the Naive Bayes algorithm (Gaussian from sklearn) was used on the balanced dataset (file #3). The dataset was split into 80-20 like the Translation dataset. Initially the first run of this data yielded a test accuracy of 0.292 but after dropping some additional features that were redundant, such as General Health and Physical Health, the accuracy was 0.563. The model was analyzed with precision, recall, F1 score, and confusion matrix as well. Overall, according to the confusion matrix there were more true positive classifications on the whole.

To use Naive Bayes on the Heart Disease dataset, we imported the built in version of Gaussian Naive Bayes from sklearn. The data was run through the model twice, once with all the data (model A) and once excluding the race and age columns (model B) to see if it would make any significant difference on the accuracy since those were the two categories that were the most affected by the preprocessing. The model was analyzed with the accuracy, F1, and AUC scores as well as a confusion matrix (A has the red one and B has the blue one). The accuracies were very similar, with model A getting 84.27% and model B getting 84.34%. However, the F1 scores and AUC scores went down from model A to B, from 0.342 and 0.68 to 0.3318 and 0.67 respectively. This could be because model A had a higher percentage of positive classifications (both false and true) and model B had a higher percentage of negative classifications (both false and true). Since there are more negative classifications in the dataset as a whole than positive classifications, having a higher true negative percentage will also give you a higher accuracy in total than having a higher true positive percentage.





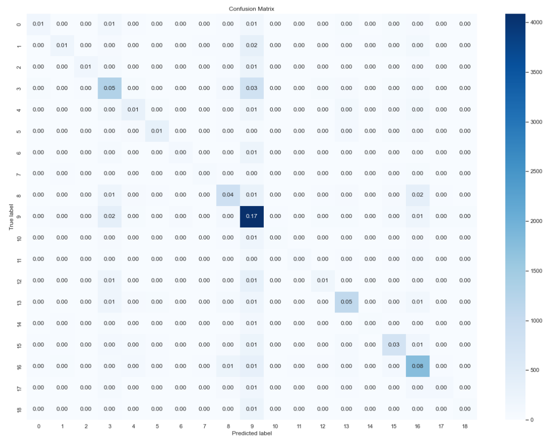
Logistic Regression

For a true machine learning model we chose logistic regression with gradient descent. For the multinomial classification problems we also used the sklearn SGDClassifier model with the log-loss function.

Multinomial: Logistic Regression / SGDClassifier

To create a sklearn LogisticRegression model for the News Category dataset, we used the ‘lbfgs’ solver and specified a ‘multiclass’ problem. The other options for the solver, ‘newton-cg’, ‘lbfgs’, ‘liblinear’, ‘sag’, ‘saga’, either took too long and never finished running or were incompatible with a multiclass problem.

Confusion Matrix:



Accuracy Statistics:

- Accuracy Score: 0.495
- Balanced Accuracy Score: 0.330
- Weighted F1 Score: 0.472

The SGDClassifier with log-loss performed slightly worse and took much longer:

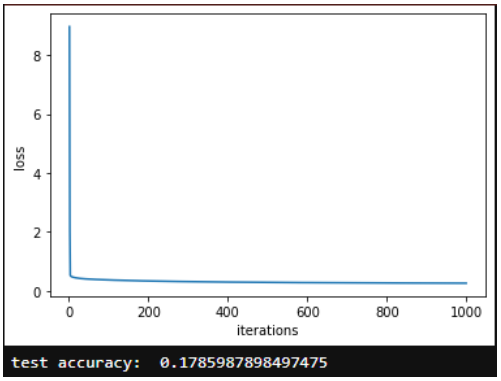
Accuracy Statistics:

- Accuracy Score: 0.467
- Balanced Accuracy Score: 0.291
- Weighted F1 Score: 0.441

And here again we see the same labels that were predicted more accurately; these are just the more frequent labels in the dataset.

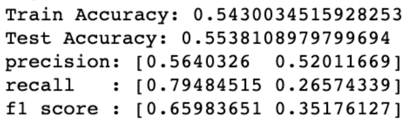
Despite this downturn in performance for the News Categories dataset, SGDClassifier performed better for the translations dataset. The highest accuracy for this dataset was 0.52 with a maximum number of iterations equal to 600. This difference in performance for the two datasets could be attributed to size of the dataset since the translations dataset is much smaller than the news categories dataset and has significantly less labels.

Binary: Logistic Regression

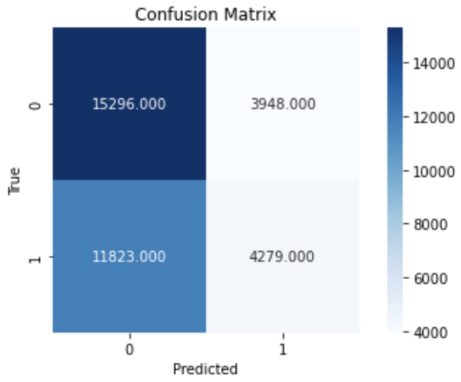


```
test accuracy:  0.1785987898497475
```

To use Logistic Regression with the binary classification problems, the sklearn version of logistic regression was imported. There are different algorithms that can be used with logistic regression, but saga was the algorithm that was the best for the Heart Disease Database since saga is faster for large databases. We analyzed the model with the mean absolute error, which was 0.157, and the log loss, which was 5.432. We also used the logistic regression algorithm that was created for Homework #3. However, after running it on the database, it gave a poor accuracy of 17.86% which was also reflected in the log loss graph.



```
Train Accuracy: 0.5430034515928253
Test Accuracy: 0.5538108979799694
precision: [0.5640326  0.52011669]
recall    : [0.79484515 0.26574339]
f1 score : [0.65983651 0.35176127]
```


Confusion Matrix

True \ Predicted	0	1
0	15296.000	3948.000
1	11823.000	4279.000

Similarly, the saga algorithm that was imported from sklearn was used with the Diabetes Indicators dataset since this dataset was also large. When we analyzed the model for this dataset, the mean absolute error was 0.446 and the log loss was 15.41. The accuracy for logistic regression was 0.553 which is slightly worse than the Naive Bayes algorithm.

Conclusion

In conclusion, there are a lot of factors that go into successfully solving a problem with machine learning. As seen throughout this paper, not only can the choice of algorithm itself affect accuracy, but so can different methods of implementing the chosen algorithm. Since different implementations of Naive Bayes are dependent on the type of classification (i.e. multinomial or binary), the type of dataset can also affect accuracy. For example, when Naive Bayes was run on our different datasets, it was shown that it performed better with binary classification than with multi-class datasets. The same effect was shown with logistic regression. However, other factors, such as the size of the dataset or the number of features, could also have affected the outcomes. Too little data, like in the case of the translations dataset, results in not enough data to train the algorithms. However, too much data can result in slow algorithms that take too many resources to run. Overall, our team found that Naive Bayes and Logistic Regression performed similarly to each other on all of the datasets, however they both performed significantly better for binary classification. This leads us to believe that these algorithms are better suited for solving problems of binary classification.

References

- News Category dataset:
  https://www.kaggle.com/datasets/rmisra/news-category-dataset

- English Word Frequency
  https://www.kaggle.com/datasets/rtatman/english-word-frequency

- Diabetes Health Indicators dataset:
  https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset

- Heart Disease Indicators dataset
  https://www.kaggle.com/datasets/kamilpytlak/personal-key-indicators-of-heart-disease

- Translations dataset:
  https://www.cl.uni-heidelberg.de/statnlpgroup/humanmt/

- scikit-learn:
  https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

- Confusion matrix visualization:
  https://medium.com/@dtuk81/confusion-matrix-visualization-fc31e3f30fea