Karla Gonzalez
Daphne Hegedus
Kelly Rombough

COMP 551
February 11, 2020

Mini Project 1 - Write Up 1

Abstract

This project had the objective of analyzing four different data sets and implementing Logistic Regression and Naive Bayes models on each of the data sets. The goal was to explore classification and compare different features and models, using 5-fold cross validation in all of the experiments and evaluating the performance using accuracy. We ran the following experiments:

1. Compared the accuracy of Naive Bayes and Logistic Regression on the four datasets.

   We found that the logistic regression approach was achieved worse accuracy than naive Bayes and was significantly slower to train.

2. Tested different learning rates for gradient descent applied to logistic regression.

   We found that there was a "sweet spot" for the learning rate (around 0.01).

3. Compared the accuracy of the two models as a function of the size of dataset (by controlling the training size).

   Naive Bayes seemed to be less affected by the change in test size, whereas Logistic Regression showed a steep decline in accuracy (to that of a complete guess) as the test size increased toward 90%.

4. Results demonstrating that the feature subset used improves performance. Ran on modified Adult Data Set.

   Most importantly, we found that the 'Sex' feature for the Adult data set contributed to the classification greatly.

Introduction

Logistic Regression is a linear discriminative model that computes the posterior probability $P(y|x)$ directly. We then apply a squashing function σ that maps all values to the range of $[0, 1]$. Then we classify based on a threshold, where $\sigma(\omega^T x) >=$ threshold is classified as 1 and $\sigma(\omega^T x) <$ threshold is classified as 0. To update the weights at each iteration, we use a full-batch gradient descent approach. Generally, we used a learning rate of 0.01 in our experiments unless otherwise specified.

Naive Bayes differs from Logistic Regression because it is a generative model, meaning it computes the joint probability $P(x, y)$ assuming conditional independence of the features. Although, this assumption is not always correct, it is a useful tool as it makes this method relatively simple to comprehend, while remaining accurate. In the training phase, we compute the means, standard deviations and prior class probabilities from a subset of the data, and use that to compute the likelihood and posterior probability $P(y|x)$.

In our experiments, we found that the accuracy of Naive Bayes wasn't affected very much by the size of the data set, whereas Logistic regression performs better on larger data sets. For example, the Haberman dataset was small and had only 64% accuracy with Logistic Regression, but 75% accuracy with Naive Bayes. We also found the number of features and number of instances correlated with the accuracy, where larger data sets with more features resulted in a higher accuracy. This can be seen with the fact that the Haberman data had the worst accuracy (due to only 3 features and 306 instances), and the Bank data set had an excellent accuracy (due to 16 features and 45211 instances). Lastly, we found that our models performed better across all data sets when tuned to a learning rate of 0.1, and k-fold cross validation with a k=5-10 (mostly due to the slow performance as k increased).

Referenced Papers and their findings: With Logistic Regression, Haberman's 1976 paper using his survival data reported a 55-59% accuracy with logistic regression, which is comparable to ours. This is probably due to the fact that this dataset only has 3 features that may not be strongly correlated to the probability of survival after 5 years. In regards to a Naive Bayes implementation, Kohavi's paper from 1996 cited a 84% accuracy with the adult dataset, and we averaged around 80% on our test set.

Data Sets

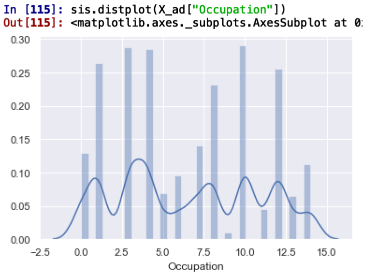
Preliminary Analysis: In order to understand our dataset better, we used the Seaborn library in Python which plots pairs of features against one another. We didn't do this for the ionosphere dataset as it has 30 features, meaning $30 * 30 = 900$ plots. These are helpful in analyzing if there are any clear correlations between two features, or between the classification and one of the features. The most important finding from this supported our idea that the Haberman features were weakly correlated with the survival rate, as there was no significant trend in the features-to-class prediction.

Below, there is a description of each dataset as well as cleaning procedures and basic statistical analysis we performed of them before training our models, in addition to the basic process of labelling the features, creating the X and Y vectors, and splitting the data into train, validation and test sets.

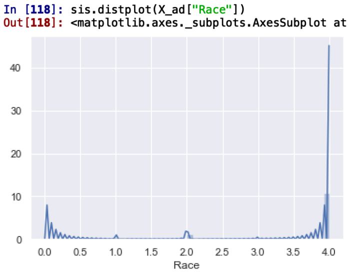
Adult Data Set: Predict whether income exceeds 50K/yr based on census data. Also known as "Census Income" data set.

1. Cleaning of Data: We will also remove certain columns which are not strongly correlated with the class.
   (a) Capital Loss and Capital Gain are too sparse, there are not enough data points to be useful.
   (b) adult[2] is a vague and non-descriptive label. Since we do not know what it represents, it can be removed initially.
   (c) The target column was labelled with $>= 50K$ and $< 50K$. To simplify the output we converted this to a binary feature with 0 and 1 as the values.
2. Missing Values: For simplicity we will remove any data entries with missing values. Notice that the missing values in this data set are denoted with "?" symbol.
3. One-hot encode categorical features.

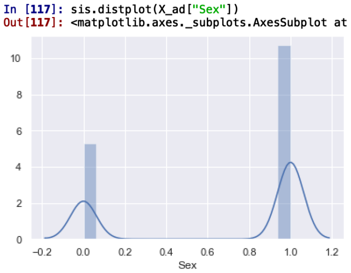
The initial analysis of the data included obtaining the mean of the initial numerical features, and some distributions of encoded variables.



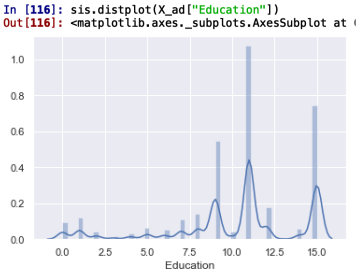
```
In [115]: sis.distplot(X_ad["Occupation"])
Out[115]: <matplotlib.axes._subplots.AxesSubplot at 0:
```



```
In [118]: sis.distplot(X_ad["Race"])
Out[118]: <matplotlib.axes._subplots.AxesSubplot at
```



```
In [117]: sis.distplot(X_ad["Sex"])
Out[117]: <matplotlib.axes._subplots.AxesSubplot at
```



```
In [116]: sis.distplot(X_ad["Education"])
Out[116]: <matplotlib.axes._subplots.AxesSubplot at (
```

An interesting observation is the number of men compared to the number of women sampled was almost double and that the most common race sampled in this data set was 'White'. This bias could influence the model to be worse at classifying minorities. Another note is that there are many 11's in education which corresponds to High School grads.

Ionosphere Data: This data set contains information about radar returns collected in Goose Bay. The aim is to predict whether a given instance is a good radar return or a bad radar return. We did the following to clean the data:

(a) Remove the second column as it was just a zero vector.

(b) Make target variables into binary classifiers

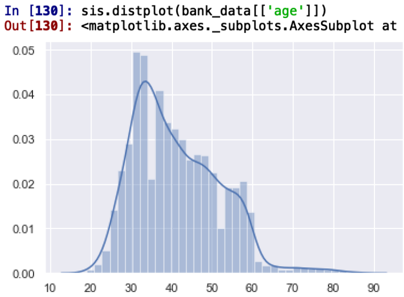
The initial analysis of the data was very limited for this data set as the labels are very non-descriptive and although we can graph distributions for individual labels, they tell us very little.

Bank Marketing Data Set: The data is related to direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. We want to predict whether the marketing campaign was successful for a given account (ie. whether or not they subscribed to the new service).

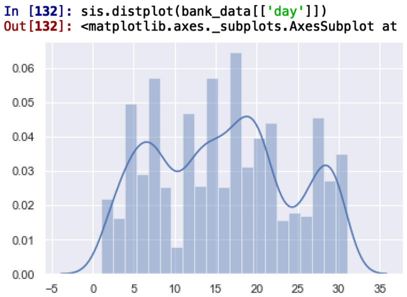
The data was cleaned and analyzed in the following way:

(a) One hot encode the categorical data

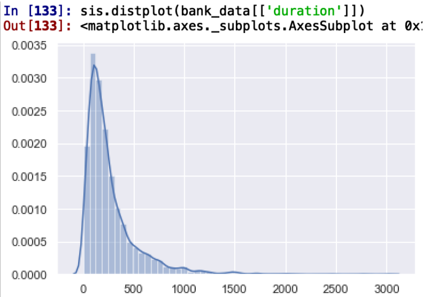
(b) Change y = ”no” to y = 0, and y = ”yes” to y = 1 to make it a binary classification.



```
In [130]: sis.distplot(bank_data[['age']])
Out[130]: <matplotlib.axes._subplots.AxesSubplot at
```



```
In [132]: sis.distplot(bank_data[['day']])
Out[132]: <matplotlib.axes._subplots.AxesSubplot at
```

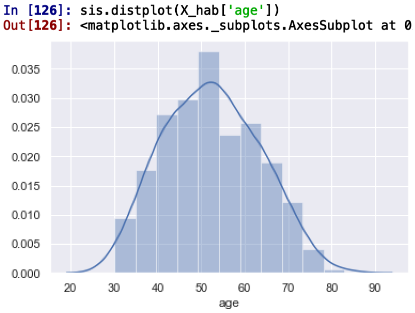


```
In [133]: sis.distplot(bank_data[['duration']])
Out[133]: <matplotlib.axes._subplots.AxesSubplot at 0x
```

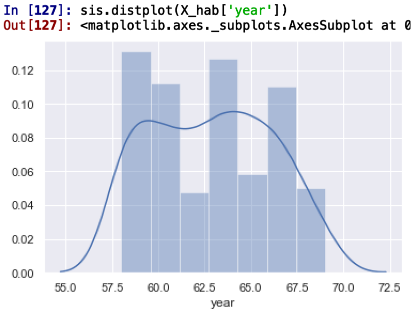
Included above were some of the distributions of the initially numerical variables included in the dataset. No permutation of subsets was found to increase or maintain the accuracy we achieved when we ran the models on the entire dataset.

Haberman’s Survival: The dataset contains cases from a study that was conducted between 1958 and 1970 at the University of Chicago’s Billings Hospital on the survival of patients who had undergone surgery for breast cancer.

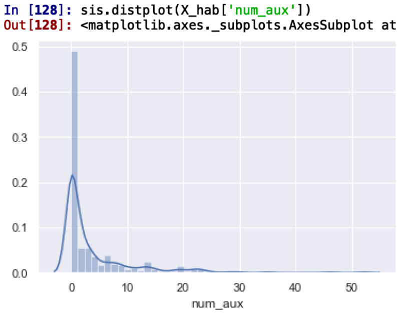
To clean the data, we change y = 1 (didn’t survive) to y = 0, and y = 2 (survived) to y = 1 to make it a binary classification.



```
In [126]: sis.distplot(X_hab['age'])
Out[126]: <matplotlib.axes._subplots.AxesSubplot at 0
```



```
In [127]: sis.distplot(X_hab['year'])
Out[127]: <matplotlib.axes._subplots.AxesSubplot at 0
```



```
In [128]: sis.distplot(X_hab['num_aux'])
Out[128]: <matplotlib.axes._subplots.AxesSubplot at
```

We can see the distributions of the (1) age of patient at time of operation, (2) year of operation and (3) number of positive auxiliary cancer nodes detected. After running all the possible permutations of the labels to see if it would improve the accuracy of our models, we found that no subset improved performance.

Results

We want to mention that upon finishing our Logistic Regression and our Naive Bayes models, we tested them on a toy example using sklearn data sets, to make sure our model implementation was correct before moving on to the complex data sets.With Logistic Regression, we were able to achieve a 96% accuracy on this example. For Naive Bayes we tested it on the Iris data set since it is often used as a benchmark data set. We achieved a 100% on this test. Both of these results told us that our model implementations were correct.

1. A comparison of the accuracy of Naive Bayes and Logistic Regression on both data sets.

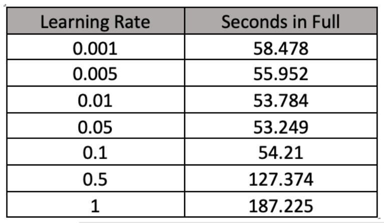
Data sets	**Logistic Regression**	**Naive Bayes**
Adult Data	0.713919	0.799177
Ionosphere	0.812394	0.840402
Haberman	0.646800	0.745002
Bank Data	0.814635	0.869499

Again we notice the slight but ever so present improvement of the accuracies when the Naive Bayes model is implemented vs. the Logistic Regression. This is due to the size of the data sets. As they are not incredibly large, we know that in this case a generative model will beat out a discriminative model.

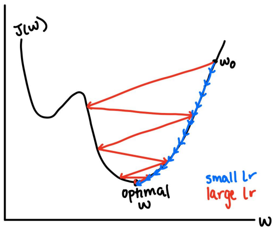
2. A discussion of how the logistic regression performance (e.g., convergence speed) depends on the learning rate.

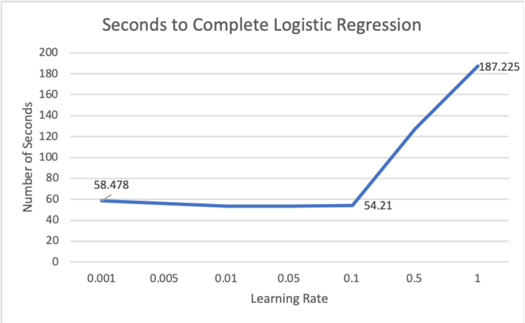
   If the learning rate is too small, then logistic regression may take too long to converge on the optimal weights, even in a convex cost function. On the other hand, if the learning rate is too large, it may overshoot the optima. This would result in either: (1) jumping to the other side of the convex cost function, or (2) missing that optimum if the function is not convex. These ideas are demonstrated in the hand-drawn graph below.

   For ours, the optimal learning rate fell at 0.01 (which seems to be a commonly used value). As a note, the seconds elapsed were to run the entire KFold script with 5-fold cross validation, and also included importing and cleaning the data. If we used the timeit library and only looked at the time elapsed for the 5-fold (without the import and clean of data), the times fell at less than 1 second, but the same trend between the learning rate and the time was seen. The average accuracies remained around the same, only slightly worsening if the learning rate was too large.

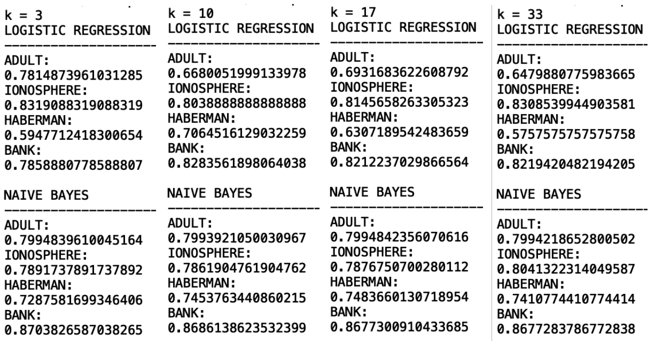


Learning Rate	Seconds in Full
0.001	58.478
0.005	55.952
0.01	53.784
0.05	53.249
0.1	54.21
0.5	127.374
1	187.225





3. Compared the accuracy of the two models as a function of the size of dataset (by controlling the training size)

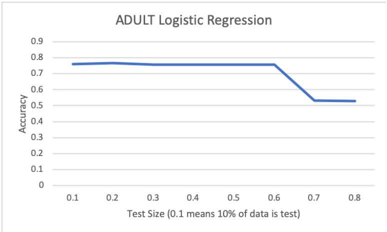


```
k = 3
LOGISTIC REGRESSION
---------------------
ADULT:
0.7814873961031285
IONOSPHERE:
0.8319088319088319
HABERMAN:
0.5947712418300654
BANK:
0.7858880778588807

NAIVE BAYES
---------------------
ADULT:
0.7994839610045164
IONOSPHERE:
0.7891737891737892
HABERMAN:
0.7287581699346406
BANK:
0.8703826587038265
```

```
k = 10
LOGISTIC REGRESSION
---------------------
ADULT:
0.6680051999133978
IONOSPHERE:
0.8038888888888888
HABERMAN:
0.7064516129032259
BANK:
0.8283561898064038

NAIVE BAYES
---------------------
ADULT:
0.7993921050030967
IONOSPHERE:
0.7861904761904762
HABERMAN:
0.7453763440860215
BANK:
0.8686138623532399
```

```
k = 17
LOGISTIC REGRESSION
---------------------
ADULT:
0.6931683622608792
IONOSPHERE:
0.8145658263305323
HABERMAN:
0.6307189542483659
BANK:
0.8212237029866564

NAIVE BAYES
---------------------
ADULT:
0.7994842356070616
IONOSPHERE:
0.7876750700280112
HABERMAN:
0.7483660130718954
BANK:
0.8677300910433685
```

```
k = 33
LOGISTIC REGRESSION
---------------------
ADULT:
0.6479880775983665
IONOSPHERE:
0.8308539944903581
HABERMAN:
0.5757575757575758
BANK:
0.8219420482194205

NAIVE BAYES
---------------------
ADULT:
0.7994218652800502
IONOSPHERE:
0.8041322314049587
HABERMAN:
0.7410774410774414
BANK:
0.8677283786772838
```



From these results, we can see that the Naive Bayes is less affected by the size of the test set/k value. This is because this method is based on the pure probabilities, and isn't affected by the initial weight guess and learning rate like Logistic Regression is. Throughout all of our experiments, Logistic Regression's accuracy varied between trails (usually in a range of 2-4% difference). Also, if we do just one trial (no k-fold cross validation and change the test size) it can be seen in the graph above that there is a steep decrease in accuracy for the adult training set with logistic regression when the test size is much larger than the training size, almost to the point of pure guessing (≈ 50% accuracy for 2 classes).

4. Results demonstrating that the feature subset you used improved performance. Ran on modified Adult Data Set.

Data Permutations of Subsets	**Accuracy (Naive Bayes)**
Original Adult	0.713919
['Age','Race','Sex','Education']	0.756540
['Age','Race','Education']	0.608524

Firstly, there were slight improvements in the accuracies when we reduced the size of the label set. Particularly with the permutation of 'Age', 'Race', 'Sex', 'Education'. Additionally, when the label 'Sex' was removed from the previously mentioned permutation, the accuracy dropped significantly. A immediate conclusion which we can draw from that is that the 'Sex' variable has high significance and is a good variable to include in our models. This was interesting as it points to some evidence of a difference in wages between men and women. Further investigation can be done by analyzing the significance of the p-values which arise when basic (yet more thorough) statistical analysis is performed. We could either confirm or discard the possible hypothesis.

Discussion and Conclusion

The principal goal of this project was to implement a functioning and accurate model of both Logistic Regression and Naive Bayes using k-fold cross validation. Of the four experiments which we ran, we concluded that not only was Naive Bayes the most reliable approach due to the smaller size of some of the data sets. That being said, the accuracy of both methods varied greatly depending on the significance of the features in a data set. This can be seen in the differing accuracies of the Bank data set and the Haberman data set, which can likely be explained by the extensive number of features in the Bank set.

Experiment 2 was particularly insightful as it demonstrated the extent to which the learning rate affects the accuracy. We found that a learning rate of 0.01 was most effective across the board, which is to be expected as it is a commonly used parameter value. While a learning rate of < 0.01 didn't necessarily decrease accuracy, it did greatly slow down the program run time. If the learning rate was > 0.01, the accuracy started to decrease.

Experiment 3 also supported our idea that a test set of more than 60% of the data would significantly decrease accuracy, since the model can only be fit on a small sample and cannot be generalized to the test set. This is an example of overfitting. On the adult data set in particular, a train-test split of $30 : 70\%$ or $20 : 80\%$ resulted in $\approx 52\%$ accuracy (essentially a guess as there is only 2 classes). This experiment also demonstrated the stability of Naive Bayes as we changed our k value in k-fold.

Experiment 4 would be interesting to continue to investigate as maybe there are other mathematical models which support our conclusion about the importance of the 'Sex' variable. It would be particularly compelling to run a similar test on census data in different countries to see if it could be replicated, as we hypothesize that this finding was due to cultural/gender norms in relation to salary.

Lastly, further investigation that could be conducted could be the implementation of leave-on-out cross validation. We chose to not do this as it was too computationally expensive for our machines.

Statement of Contributions

Daphne: Implemented Logistic Regression and Naive Bayes. Ran the second experiment and summarized the findings. Cleaned and one hot encoded the data for Haberman and Bank data sets. Implemented the K-Fold script. Ran experiment 2 on different learning rates. Contributed to the final draft of the write-up.

Kelly: Implemented Logistic Regression and Naive Bayes. Contributed to the final draft of the write-up. Ran experiment 1 to compute accuracies.

Karla: Uploaded, cleaned and one hot encoded Ion and Adult data sets. Computed basic statistics on data sets. Ran experiment 3 by changing k in k-fold. Ran the 4th experiment about removing features and finding pertinent variables. Developed skeleton of the k-fold script. Authored the write-up.

Citations

Haberman, S. J. (1976). Generalized Residuals for Log-Linear Models, Proceedings of the 9th International Biometrics Conference, Boston, pp. 104-122.

Ron Kohavi, "Scaling Up the Accuracy of Naive-Bayes Classifiers: a Decision-Tree Hybrid", Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996