**Data Analytics II:**

**Assignment #3**

**Readings:** http://ai.stanford.edu/~ang/papers/nips01-discriminativegenerative.pdf

1. Fit a logistic regression model to the mushrooms dataset from the UCI data repository using 10-fold cross-validation. Next, fit a (Bernoulli) Naïve Bayes model to the same dataset using the same criteria. First, report the average accuracy scores for each type of

Next, generate aggregate contingency tables for each type of classifier

Which type of classifier would you use when trying to decide if you want to eat a mushroom? Why?

I would care more about the precision in both models, because I wouldn’t want to eat a mushroom that is predicted ediblie but is actually poisonous.

The precision ratio using logistic regression is 60/(60+131)=0.31

The precision ratio using naïve bayes is 78/(78+109)=0.42

In this case, I would be choosing Naïve Bayes instead of Logistic Regression even though the accuracy scores for Logistic Regression is higher than that for Naïve Bayes

1. Do the example assignment in Chapter 6 of the NLTK book: <http://www.nltk.org/book/ch06.html>
   1. Develop a Naïve Bayes classifier that can determine the likely gender of a given name, including at least four features using a training set consisting of 500 randomly-selected samples. List the top 10 features for this classifier along with their corresponding odds ratios. Also indicate the accuracy of this classifier on the test set:

Top 10 features for this classifier along with their corresponding odds ratios:

last\_letter = 'a' female : male = 22.9 : 1.0

last\_letter = 'o' male : female = 16.6 : 1.0

last\_letter = 'd' male : female = 12.2 : 1.0

count(o) = 2 male : female = 7.9 : 1.0

last\_letter = 'r' male : female = 5.5 : 1.0

last\_letter = 'x' male : female = 4.3 : 1.0

count(a) = 3 female : male = 4.1 : 1.0

first\_letter = 'o' male : female = 4.0 : 1.0

first\_letter = 'w' male : female = 3.9 : 1.0

count(d) = 2 male : female = 3.5 : 1.0

The accuracy for this classifier: 0.768

* 1. Determine the accuracy of this classifier for all of the students in this course.

0.5238095238095238

* 1. Fit a logistic regression model to the same dataset using the same features. Which one performs better? Why?

Naïve Bayes performs better. Logistic Regression needs more data and Naïve Bayes performs better when you only have small amount of data

* 1. Vary the size of the training set from 100 to 2000 in increments of 100. For each increment, fit a Naïve Bayes classifier and a Logistic Regression classifier. Plot or tabulate the accuracy of each classifier. What can you conclude about the relative performance of NB vs. MaxEnt classifiers as the size of the training set increases?

Logistic regression is doing better when the size of training dataset is relatively small, but as the size of training dataset increases, NB model starts to perform better than logistic regression model.

1. Fit several logistic regression models and (Gaussian) naïve Bayes models to the Pima diabetes dataset from the UCI machine learning repository, varying training set size from 10 to 500 in units of 10. Create a plot comparing how average accuracy from each type model varies with training set size. When does each model perform better and why?

Seems like when the data set is very small (smaller than 50), Gaussian naïve bayes models perform better, but when the data set is larger, logistic regression is doing better.

I think it’s probably because that naïve bayes models assume that there’s no relationships between features while logistic regression doesn’t care. However, when the dataset is small, one data/outlier could significantly shift the logistic regression model but not so much to naïve bayes models.

1. Using data from the US Social Security Administration (<https://www.ssa.gov/oact/babynames/limits.html>), develop a gender classifier using only full names as features. Compare the classifier from #2 to this classifier. What does these new classifiers predict for the list of students in this course (for simplicity, assume that undergraduate students in this course were born in 1995, and graduate students were born in 1985)?

I wrote my own algorithm for this question. However, running it is very slow, so I stored the results I got from running it in the target list directly for later use.

The classifier predicts undergraduate students to be 6 males and 1 female

The classifier predicts graduate students to be 10 males and 4 females

What do the classifiers predict for the gender of the names “Dylan”, “Madison”, “Tyler”, and “Dana”, assuming that they are 21 and 30 years old, and assuming that they were born in 2015?

Assume LaPlacian smoothing for this problem. (HINT: coding your own algorithm may be faster!)

We can see from the Jupyter Notebook results that assuming they are 21, the classifier predicts them to be [‘M’, ‘F’, ‘M’, ‘F’].

Assuming they are 30, the classifier predicts them to be [‘M’, ‘F’, ‘M’, ‘F’].

Assuming they were born in 2015, the classifier predicts them to be [‘M’, ‘F’, ‘M’, ‘F’]

1. Do the tutorial at <http://scikit-learn.org/stable/tutorial/statistical_inference/supervised_learning.html>
   1. Explain the “curse of dimensionality” in your own words and describe how this problem is addressed by
      1. k-Nearest-Neighbor

When you are using kNN classifier curse of dimensionality is a big issue because when the dimensionality/number of features increases, the amount of data needed to develop/train/fit a good classifier increases exponentially. Therefore, when the dimensionality/number of features increases to some point, even big data is not enough.

* + 1. MaxEnt

MaxEnt classifiers are log-linear models, therefore they are robust when it comes to curse of dimensionality, which means that they wouldn’t be affected as much as kNN classifiers

* + 1. Naïve Bayes, and

Naïve Bayes works well when the assumption, which is, all the features are conditionally independent, which means that it assumes that each feature is unidimensional, therefore, curse of dimensionality doesn’t do as much hard to Naïve Bayes as long as the features are independent, if they are not, the dimensionality is going to be lower

* + 1. Support Vector classifiers

The results of SVM don’t depend on dimensionality of space, rather, they depend on a dot product(inner product) of data vectors, therefore SVM will not be affected much by curse of dimensionality

1. Do the example assignment at <http://scikit-learn.org/stable/auto_examples/svm/plot_separating_hyperplane.html#example-svm-plot-separating-hyperplane-py>
   1. Comment each line of code, explaining its function
   2. Explain the concept of a maximum-margin hyperplane in your own words.

The distance between the boundary and the nearest point in each class is called margin, and therefore, in order to well classify each class (which means make sure that the right class will be correctly classified), you’ll have to find the maximum distance between the closet points between two classes, and make sure that the distance between the boundary and the closet point of each class is the same, or at least very close to each other so that the points wouldn’t be misclassified.

1. Use a linear SVM to construct a sentiment classifier using the Breast Cancer Wisconsin (Diagnostic) Data Set from the UCI Repository
2. Include all features and use them to construct a soft-margin linear SVM classifier using a standard hinge loss function. Using 10-fold cross-validation, what is this classifier’s accuracy?

The average accuracy is 0.57

1. Explain a hinge loss function in your own words

Hinge loss is a loss function used to train maximum-margin classifiers. When the y and t have the same sign, l(y)=0, otherwise l(y) increases linearly with y.

1. Use a non-linear kernel to generate classification results. What is this classifier’s accuracy? Is there another metric, besides accuracy, that would lead you to favor the classifier with the lower accuracy?

The average accuracy for 10-fold validation with the non-linear kernel is 0.63.

I would be more concerned about the false negative ratio (or Precision) because those are the ones that are predicted not to have cancer but actually have cancer, if they miss the opportunity to get treated in time it might be dangerous

For the linear SVM, the ratio is 36/(130+36)=0.2169

For the non-linear SVM, the ratio is 39/(130+39)=0.2308

Therefore I would prefer the non-linear SVM which has a higher accuracy and also lower false negative/higher precision ratio

1. Do the example classifications at <http://scikit-learn.org/stable/auto_examples/svm/plot_iris.html>
2. Comment each line of code, explaining its function
3. Explain the results of each of these outputs. How do they differ and why?

SVC with linear kernel basically is to try to separate the classes with linear kernel, it uses libsvm estimators that don’t penalize/scale the intercept. It minimizes the regular hinge loss.

LinearSVC minimizes the squared hinge loss and uses liblinear estimator and do infact penalize the intercept. It also tries to separate the classes linearly.

We can see that the plot generated with SVC with linear kernel and LinearSVC are pretty similar, with some minor differences of how the linear line was drawn between the red dots and the white dots part

SVC with RBF kernel is using a rbf kernel instead of a linear kernel and maps a single vector to a vector of higher dimensionality, so that when classes cannot be separated well with linear kernel, they are projected to high dimensionality so that they could be separated better in higher dimensionality. It’s one of the kernel tricks.

We can see that with SVC with RBF kernel, there’s a small ellipse around the concentration area of blue class and an ellipse around the concentration area of white class and the rest as a whole for the red class (might just because the red dots have a larger sporadic distribution than the other two classes, therefore in higher dimensionality, it’s taking the rest of the space.

SVC with polynomial (degree 3) kernel: looks not only at the given features of input samples to determine their similarity but also combinations of the features.

The SVC with polynomial kernel doesn’t separate the three classes harshly, instead it uses a combination of curves and ellipses

1. Explain the “kernel trick” in your own words

Sometimes when the linear SVC cannot classify/separate classes very well, with kernel tricks, the dots/classes are projected to higher dimensionality, where they can be better separated/classified.

1. Write out the loss functions of SVM, Naïve Bayes, and Logistic Regression. Under what circumstances should each of these be used?

SVCs usually use a “hinge loss function” which is:

𝑙 𝑦 =max(0,1−𝑡:𝑦)

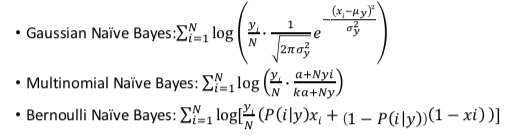
•𝑦=𝒘:𝒙+𝑏

• t= 1 if class is correctly predicted; -1 otherwise.

SVCs can also use zero-one loss function in which ly=1 if class is corrected predicted and 0 if otherwise. Zero-one loss function cares only about if you make the right prediction or not, which is accuracy. However, if you care about more than accuracy you should use hinge loss function for SVCs

Naïve Bayes :

𝑙 𝑦 =log[P(x,y)]=P(x|y)P(y)



Gaussian/Multinomial/Bernoulli Naïve Bayes classifiers usually use its own loss function

Foe Logistic Regression:



Logistic Regression classifier usually use its own loss function

1. EXTRA CREDIT: Using all of the classifier types that you’ve learned so far (MaxEnt, Naïve Bayes, SVM), create sentiment classifiers using the UCI Sentiment Labeled Sentences dataset. Report their respective accuracy scores. Which classifier would you use for this task and why? (HINT: Performance of the Naïve Bayes classifier can be boosted using TF-IDF weighting)
2. Fit a MaxEnt (ME) model, a Naïve Bayes (NB) model, and a Support Vector classifier model to the Thoracic Surgery Data dataset, which aims to predict 1-year survival rate of thoracic surgery patients.
   1. Calculate the baseline proportion of the population that survives for greater than one year. If necessary, undersample the population such that there are equal numbers of survivors and non-survivors.

The baseline proportion of the population that survives for greater than one year is around 15%

* 1. Select a model scoring criterion and justify your choice in the context of the dataset.

In this case a false positive is really bad (predicted that the patient is going to survive while the patient actually is not going to survive, which means that the patient will require more attention/treatment/care) so I choose precision as the scoring criterion

* 1. Use grid search cross validation to select the best-fitting classifier for each type of model. What is the best fitting model in each category, and what is its score?

The best fitting model for Logistic Regression is when C=1 and using l2 penalty

Its score is 0.6116

The best fitting model for Naïve Bayes is when alpha= 1e-06

Its score is 0.65

The best fitting model for RBF SVC is when C=10, gamma=0.001

Its score is 0.5

The best fitting model for Poly SVC is when C=100, degree=3

Its score is 0.567

* 1. After applying 10-fold cross-validation, generate an averaged contingency table for each best-fitting classifier in the ME, NB, and SVC categories.

In Jupyter Notebook

* 1. For each best-fitting classifier, indicate the following:
     1. Accuracy
     2. Precision
     3. Recall
     4. F1

The accuracy score for the best-fitting Logistic Regression classifier is 0.525,

The Precision score for the best-fitting Logistic Regression classifier is 0.5179

The Recall score for the best-fitting Logistic Regression classifier is 0.5177

The F1 score for the best-fitting Logistic Regression classifier is 0.5175

The accuracy score for the best-fitting Multinomial Naïve Bayes classifier is 0.55

The Precision score for the best-fitting Multinomial Naïve Bayes classifier is 0.5625

The Recall score for the best-fitting Logistic Regression classifier is 0.5606

The F1 score for the best-fitting Logistic Regression classifier is 0.5489

The accuracy score for the best-fitting RBF SVC classifier is 0.4,

The Precision score for the best-fitting RBF SVC classifier is 0.394

The Recall score for the best-fitting RBF SVC classifier is 0.394

The F1 score for the best-fitting RBF SVC classifier is 0.394

The accuracy score for the best-fitting Poly SVC classifier is 0.575,

The Precision score for the best-fitting Poly SVC classifier is 0.5751

The Recall score for the best-fitting Poly SVC classifier is 0.575

The F1 score for the best-fitting Poly SVC classifier is 0.5747

* 1. For each best-fitting classifier generate a plot of the ROC curve and indicate the AUC

The AUC for the best-fitting Logistic Regression classifier is 0.54

The AUC for the best-fitting Multinomial Naïve Bayes classifier is 0.56

The AUC for the best-fitting RBF SVC classifier is 0.41

The AUC for the best-fitting Poly SVC classifier is 0.56

* 1. Make an argument for the selection of a classifier based on the metrics listed above; however, do not base your argument solely on the metric selected for part a, and justify your choice.

the best fitting Poly SVC outperformed all other best classifiers in accuracy, precision, recall and F1 score. I care more about precision criterion. But it is very time consuming to fit and use grid search on to find the best fitting model. Besides, the AUC of RBF outperformed the AUC of Poly, and the AUC of Multinomial Naïve Bayes are not bad either. And the dataset is small in which case Naïve Bayes outperforms Logistic Regression. In this case I would choose Naïve Bayes

1. FINAL PROJECT Module 3: Fit Naïve Bayes, MaxEnt, and Support Vector classifiers to your dataset identified in Module #1. Using your knowledge of this dataset, choose a metric to score the quality of your classifiers and justify your choice of metric in the context of a specific problem that your data would address. Indicate which classifier achieves the highest scores and speculate why this might be the case given your knowledge of the dataset. Finally, provide contingency tables and plots of the ROC curve for each classifier and indicate the strengths and weaknesses of each classifier for your specific dataset.

I choose accuracy for this question since the correct prediction matters more in this question.

The accuracy of Logistic Regression Model is 0.88

The accuracy of Multinomial Naïve Bayes Model is 0.75,

The accuracy of Linear SVC is 0.86

The AUC of Logistic Regression is 0.93

The AUC of Naïve Bayes is 0.78

The AUC of Linear SVC is 0.93

Overall Logistic Regression model performs the best among the three models.