IML (67577) Hackathon - Who Tweeted What?

Or Mizrahi, Daniel Levin, Shahar Nachum, Alon Emanuel

June 7, 2019

Our Goal

Our task was to implement a supervised learning algorithm for classifying tweets into their appropriate authors.

The Setting

As input we got ten data sets (.csv), each consisting of ~ 3200 tweets posted by a given public figure, e.g. Donald Trump, Kim Kardashian etc.

Tweets are given as strings, and the labels are integers from 0 to 9.

As output we returned a decision rule (=a hypothesis) that receives a tweet and outputs a label potentially determining it's author.

Our Work Process

1. Put Your Test-Set in a Vault

To avoid data-snooping resulting in biased results, we start of by setting our test-set aside. It is untouched until the very last stage of the learning process, to be used as an honest estimation for our generalized error.

As a rule of thumb, the test-set was chosen to be 0.15% of the data given.

2. Set the Bar

In order to improve ourselves and have some point of reference for our performance, we set up a naive baseline learner.

It had poor results, but it had **some** results, which was what we needed.

The baseline learner used the Bag of Words method to vectorize each tweet, and suffered from large variance, resulting in a generalization accuracy of ~ 0.2.

3. Extract Features

Since the data is raw text, we had to manually extract features out of it. To help us decide on what features we want to include, we observed the data and understood its behavior from the birds eye.

We found that a specific implementation of Bag of Words, with specific values given to its parameters.

4. Select Your Models

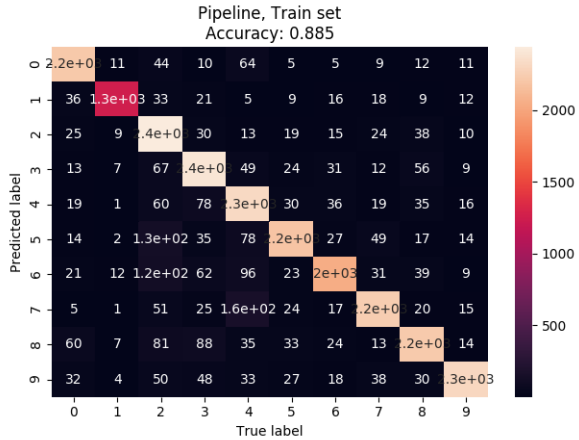
We tried a lot of models.

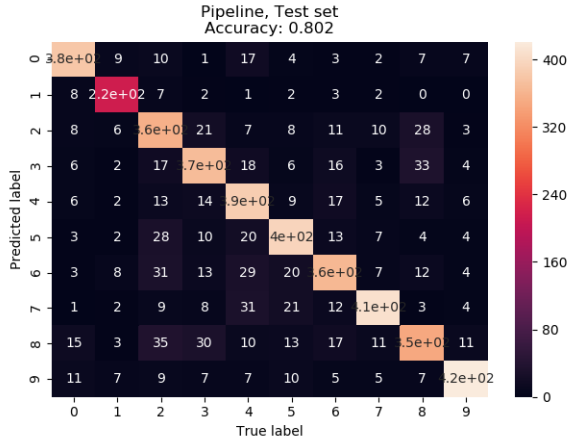
We played with each model's knob to find a sweet spot between the time complexity and the model performance. Our final contestants were Logistic Regression, Random Forest, Multinomial Naive Bayes and

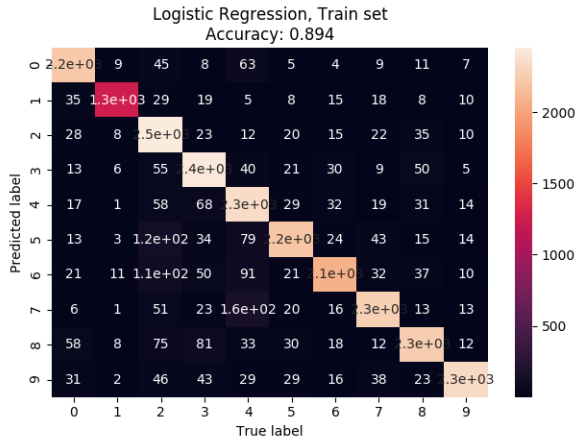
On the other hand, we noticed that using a pipeline over these models made them even more robust and accurate. So eventually we went with Linear SVC pipelined.

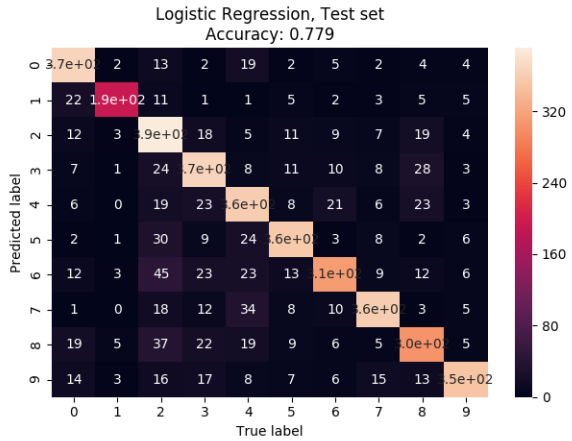
The Results

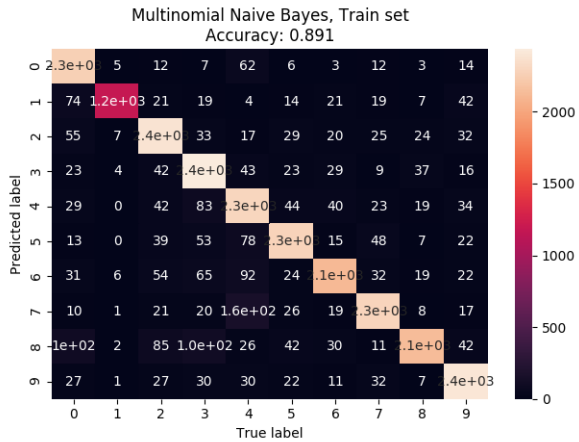
These are the results per classifier (train + test error), in descending order:

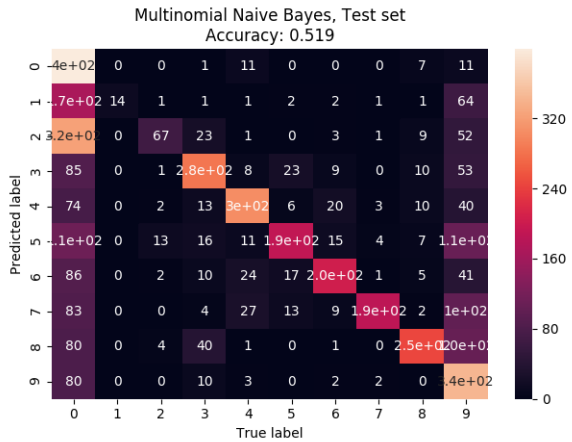

Pipeline, Train set
Accuracy: 0.885
Predicted label
True label


Pipeline, Test set
Accuracy: 0.802
Predicted label
True label


Logistic Regression, Train set
Accuracy: 0.894
Predicted label
True label


Logistic Regression, Test set
Accuracy: 0.779
Predicted label
True label


Multinomial Naive Bayes, Train set
Accuracy: 0.891
Predicted label
True label


Multinomial Naive Bayes, Test set
Accuracy: 0.519
Predicted label
True label