Twitter Sentiment Analysis using NLP

Capstone project I

Priya Theru

Springboard April 2019
Data Science Career Track

Background

- Labels represent the sentiment of a text. These labels helps in analyzing the travellers' experience in using the US airlines - the actual feedback is from the text itself. The primary goal is to build a classifier that understands the text and assign an appropriate label to it.
- Each sentiment label is categorized as positive,negative and neutral. Therefore we will have a supervised, multi-class classifier with actual text as input.
- The dataset contains the travellers' reviews on US Airlines in February 2015. There are a total of 14640 records of different airline reviews.

Steps

- Data Wrangling
- Inferential Statistics
- Data Visualization
- Machine Learning model

Data Wrangling

- The data is stored in a .csv file. It is downloaded and is stored in the form of pandas dataframe.

- The columns tweet_id and retweet_count are integers,airline_sentiment_confidence and negative_reason_confidence are floats while the rest of them are strings(objects). Also, user_timezone column has only 9820 records therefore the missing values is filled by 'forward fill' method of 'fillna' function.

- I created a map of some common contractions and have written a customized function 'expand' to expand the words if it come across any of them listed in the map.

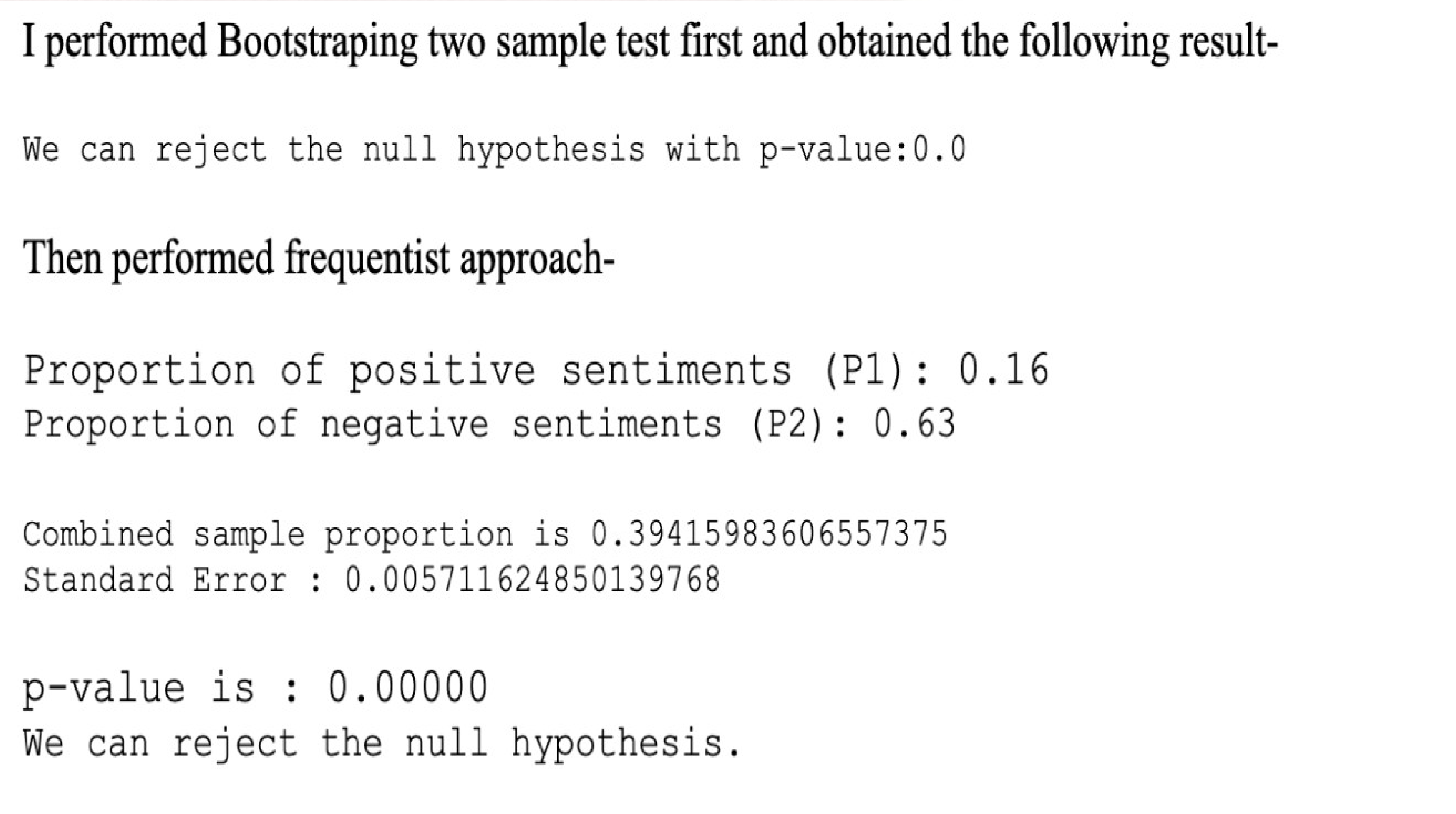
Data Wrangling contd.

- The function 'tweet_to_words' first expands the contractions, gets rid of words that have multiple repeating characters in them such as the word 'loooove' basically is 'love' and the user is trying to convey the degree of intensity by putting in multiple repeating characters. Then it gets rid of symbols, punctuations and numbers, stop words, words that have 'http' or 'www' in them, words whose lenghths are greater than 2. Finally the words are lemmatized which means that it converts all of them to their root words, for example, the word 'caring' has the lemmatized(root) word 'care'.

Inferential Statistics

- Bootstrap and frequentist methods were used to test if the proportion of negative sentiments are equal to the positive sentiments.
- After simulating the conditions for 10,000 trials, we got the same results following both the bootstrap and frequentist approaches, hereby concluding that the proportion of negative sentiments is NOT equal to positive sentiments.

Inferential Statistics contd.



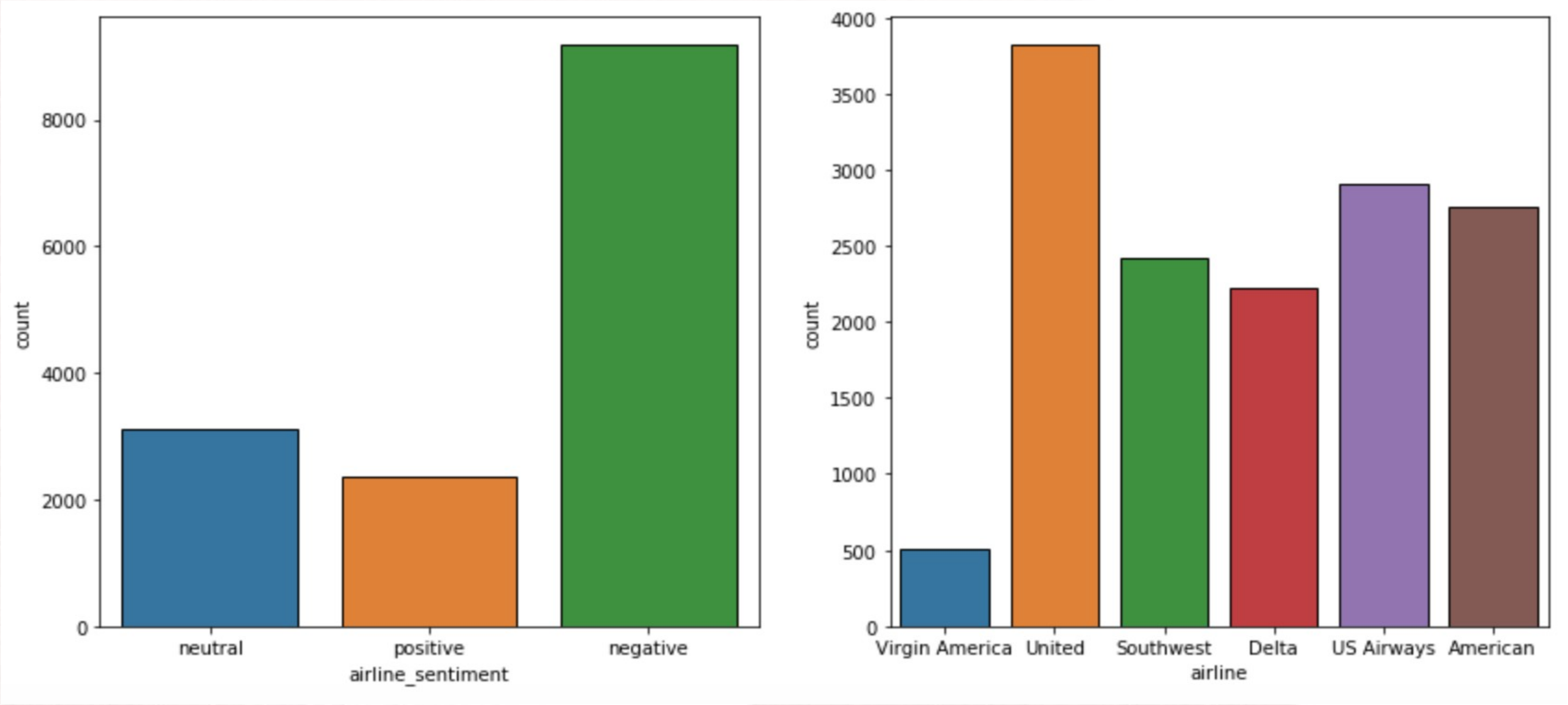
I performed Bootstraping two sample test first and obtained the following result-

```
We can reject the null hypothesis with p-value:0.0
```

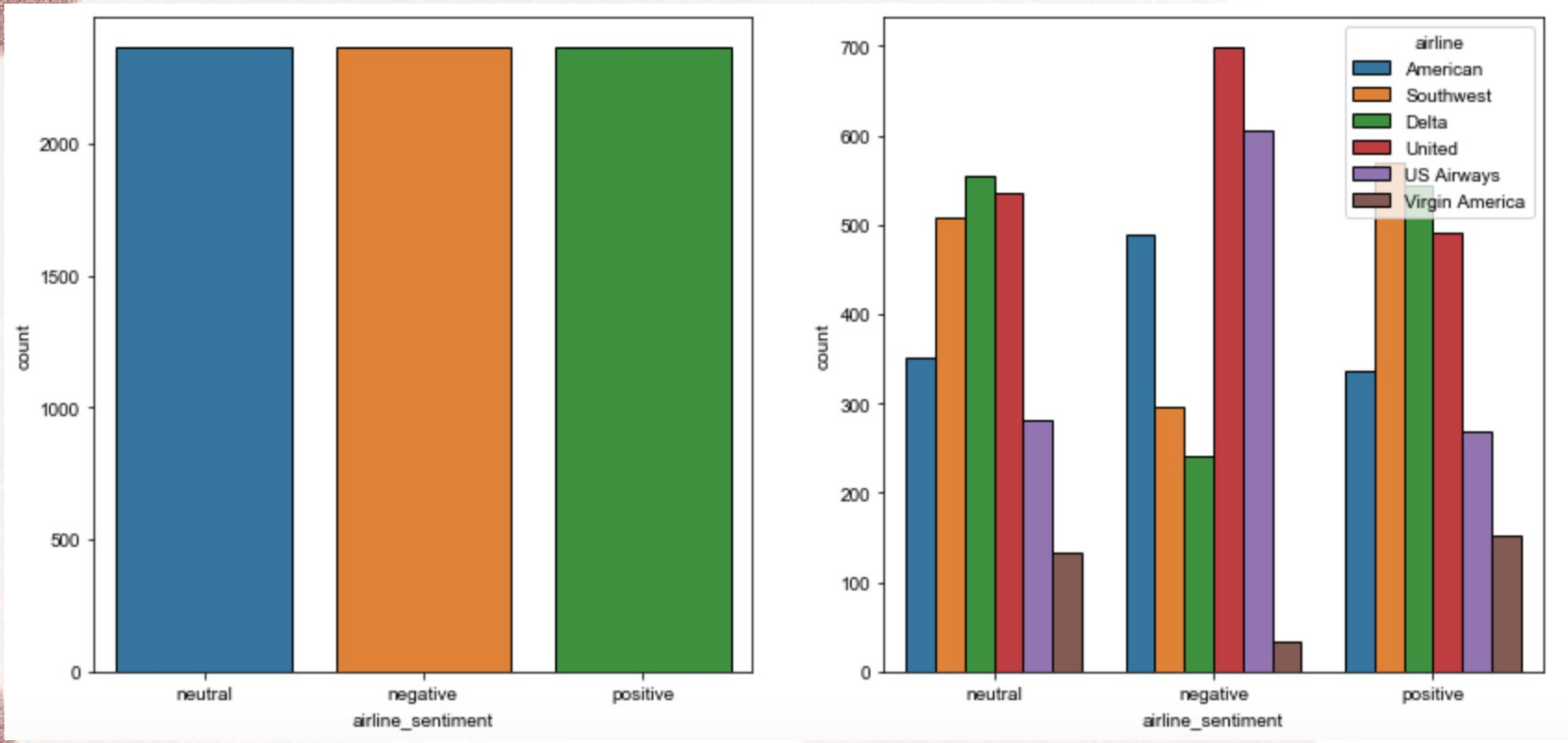
Then performed frequentist approach-

```
Proportion of positive sentiments (P1): 0.16
Proportion of negative sentiments (P2): 0.63

Combined sample proportion is 0.39415983606557375
Standard Error : 0.005711624850139768

p-value is : 0.00000
We can reject the null hypothesis.
```

Data Visualization



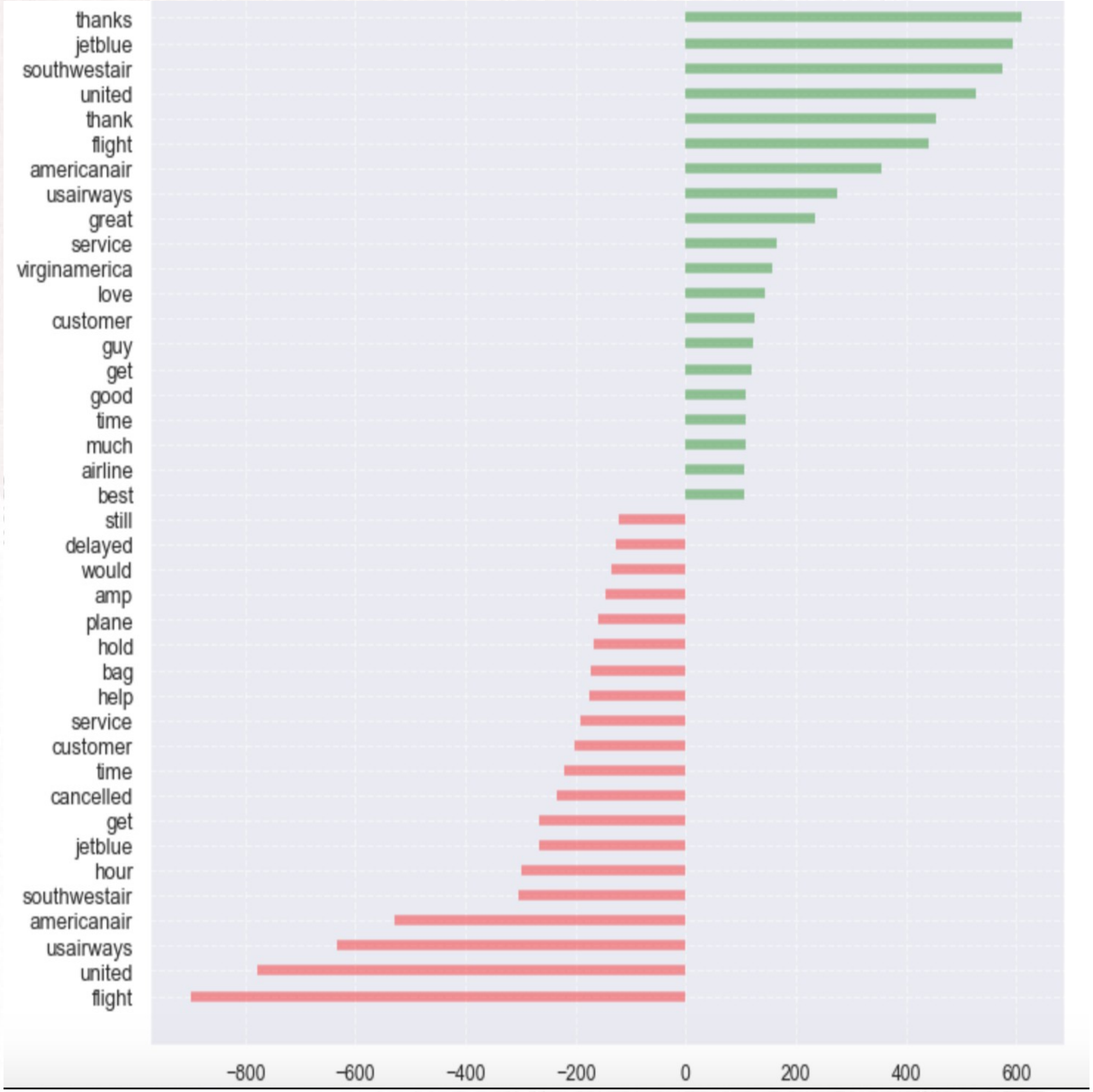
Data Visualization contd.



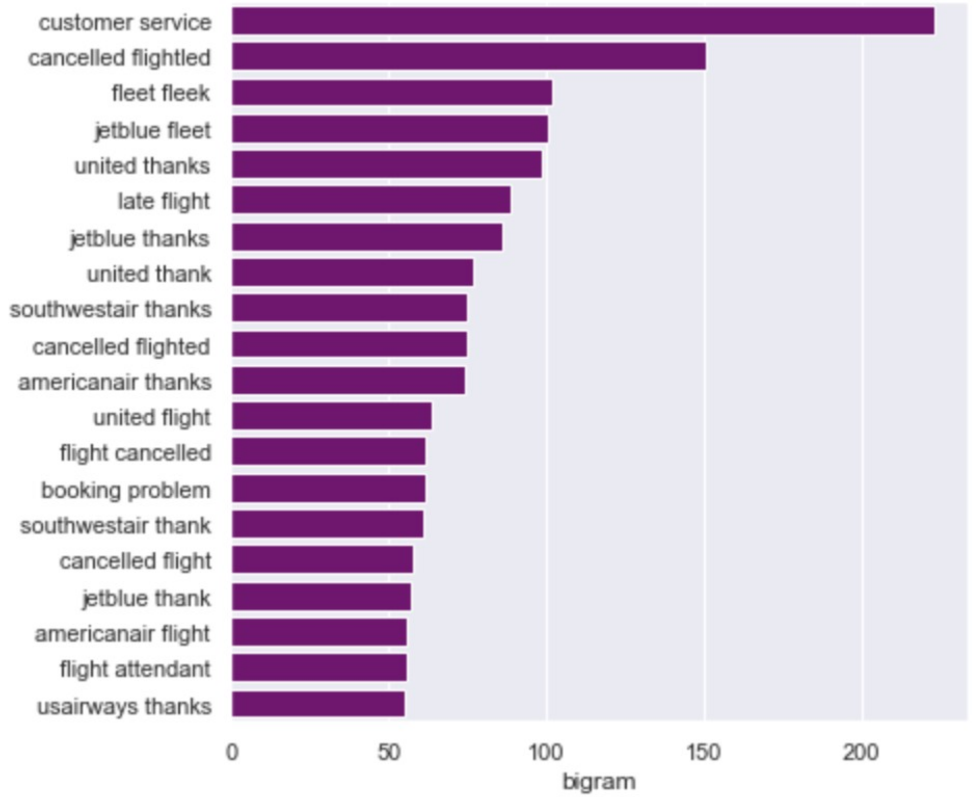
- The number of negative sentiments are equal to positive sentiments after downsampling the majority class without replacement.

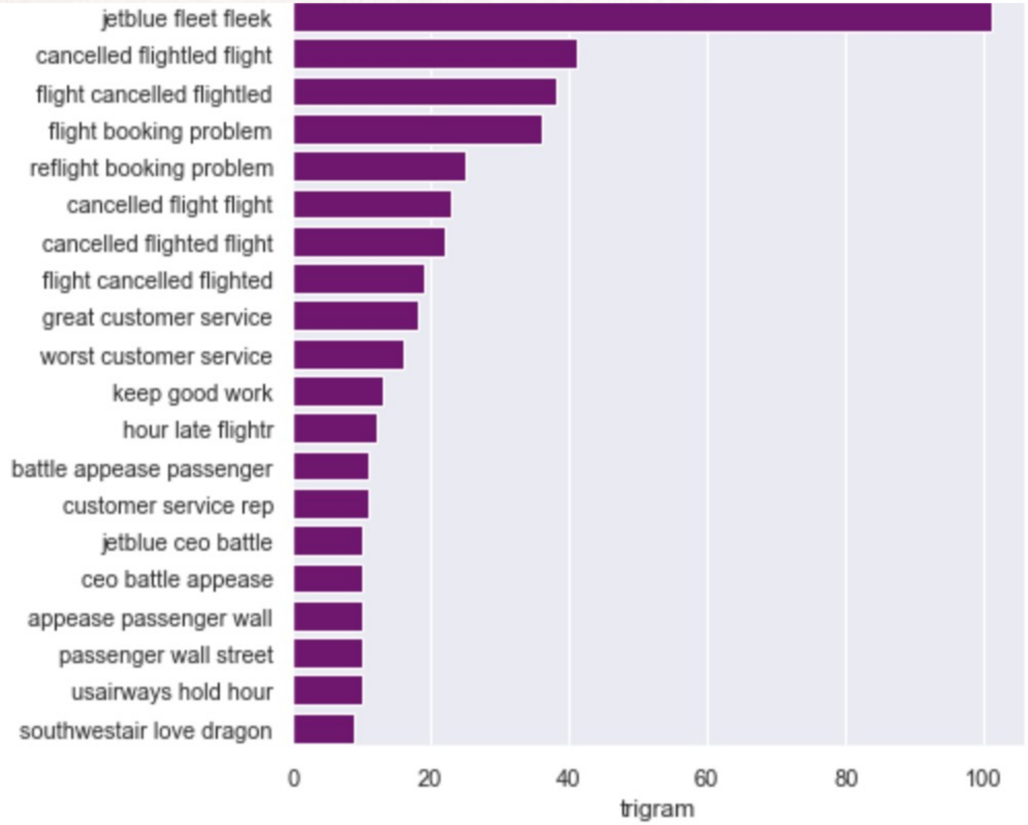
Data Visualization contd.

- Diverging bars of positive and negative words after pre-processing.



Data Visualization contd.

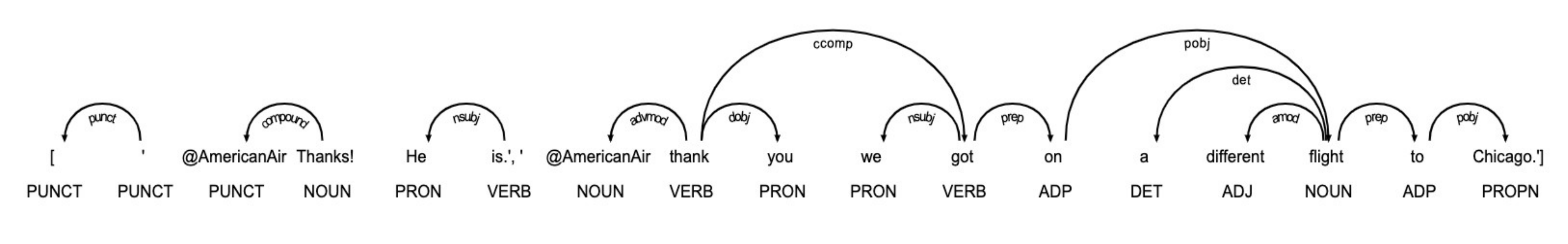


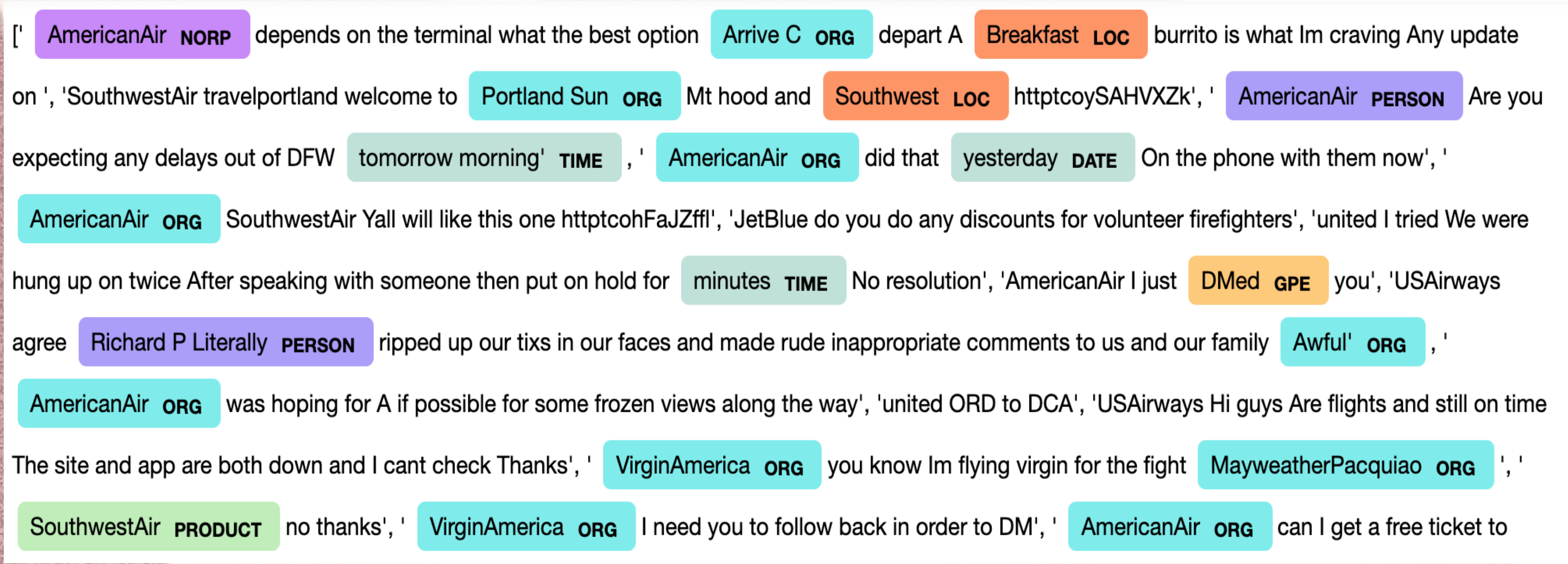


- Bigrams and Trigrams

Data Visualization contd.

- SpaCy's dependency tree and NER





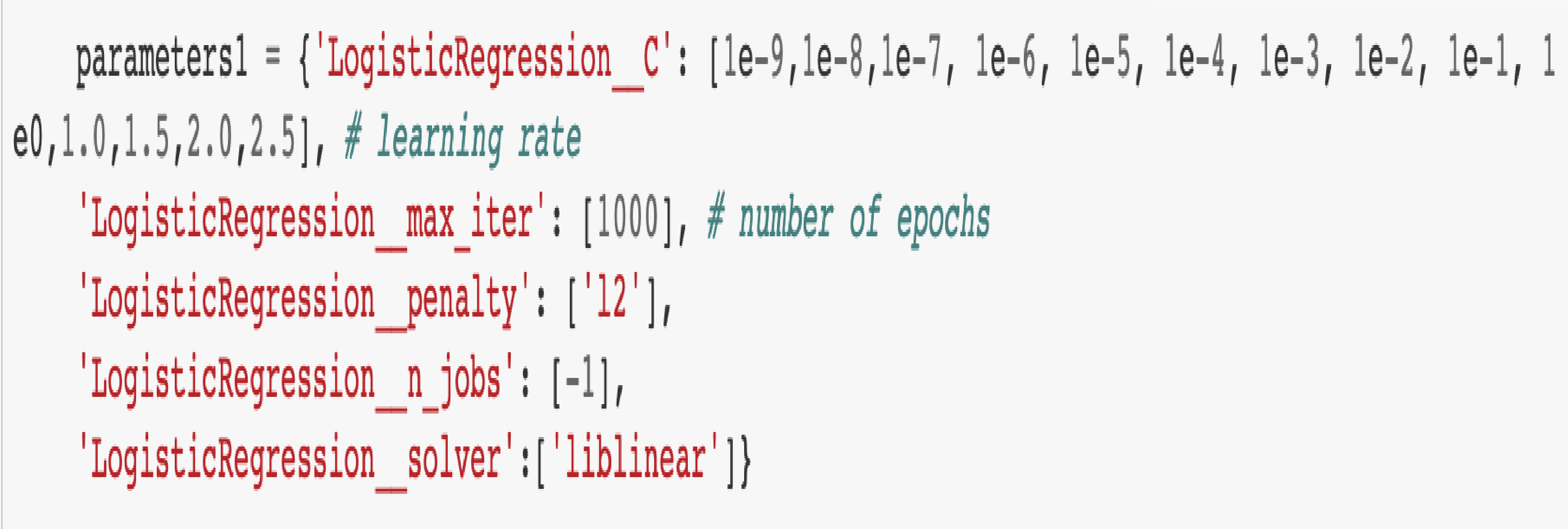
[' AmericanAir NORP depends on the terminal what the best option Arrive C ORG depart A Breakfast LOC burrito is what Im craving Any update
on ', 'SouthwestAir travelportland welcome to Portland Sun ORG Mt hood and Southwest LOC httptcoySAHVXZk', ' AmericanAir PERSON Are you
expecting any delays out of DFW tomorrow morning' TIME , ' AmericanAir ORG did that yesterday DATE On the phone with them now', '
AmericanAir ORG SouthwestAir Yall will like this one httptcohFaJZffl', 'JetBlue do you do any discounts for volunteer firefighters', 'united I tried We were
hung up on twice After speaking with someone then put on hold for minutes TIME No resolution', 'AmericanAir I just DMed GPE you', 'USAirways
agree Richard P Literally PERSON ripped up our tixs in our faces and made rude inappropriate comments to us and our family Awful' ORG , '
AmericanAir ORG was hoping for A if possible for some frozen views along the way', 'united ORD to DCA', 'USAirways Hi guys Are flights and still on time
The site and app are both down and I cant check Thanks', ' VirginAmerica ORG you know Im flying virgin for the fight MayweatherPacquiao ORG ', '
SouthwestAir PRODUCT no thanks', ' VirginAmerica ORG I need you to follow back in order to DM', ' AmericanAir ORG can I get a free ticket to

Machine Learning

- Logistic Regression
- KneighborsClassifier
- SVC
- DecisionTreeClassifier
- RandomForestClassifier
- AdaBoostClassifier
- Gaussian Naive Bayes
- Multinomial Naive Bayes
- SGDClassifier

Logistic Regression

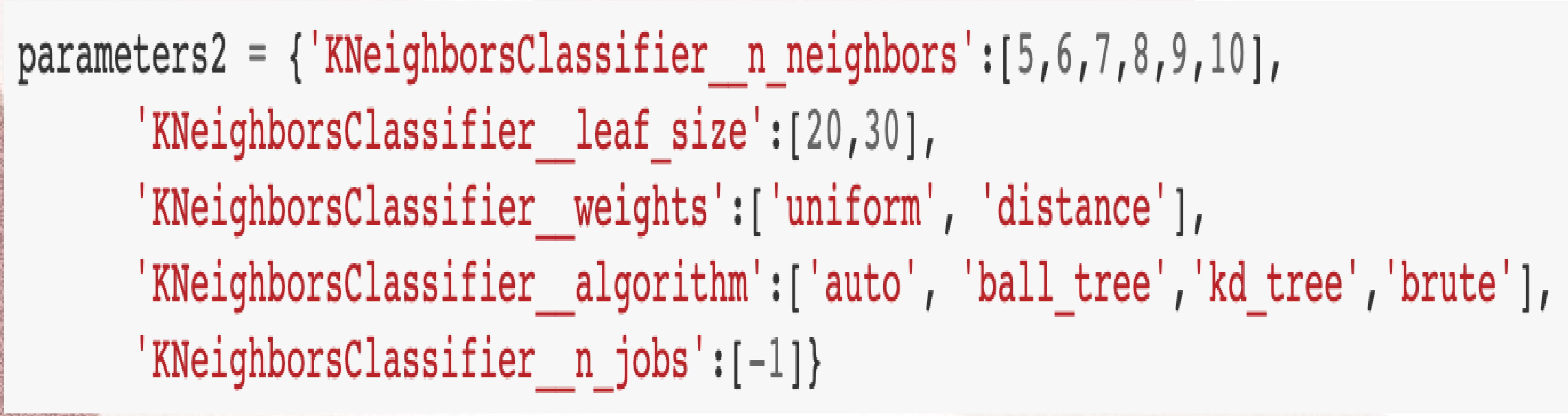
- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.



```
    parameters1 = {'LogisticRegression__C': [1e-9,1e-8,1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1
e0,1.0,1.5,2.0,2.5], # learning rate
     'LogisticRegression__max_iter': [1000], # number of epochs
     'LogisticRegression__penalty': ['l2'],
     'LogisticRegression__n_jobs': [-1],
     'LogisticRegression__solver':['liblinear']}
```

KNeighborsClassifier

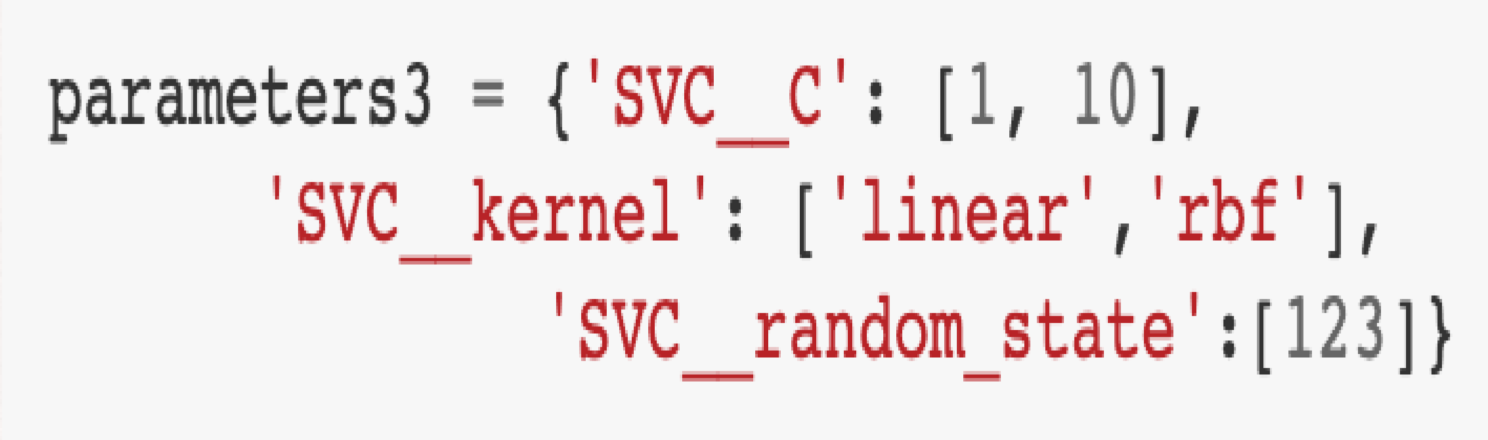
- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.



```
parameters2 = {'KNeighborsClassifier__n_neighbors':[5,6,7,8,9,10],
         'KNeighborsClassifier__leaf_size':[20,30],
         'KNeighborsClassifier__weights':['uniform', 'distance'],
         'KNeighborsClassifier__algorithm':['auto', 'ball_tree','kd_tree','brute'],
         'KNeighborsClassifier__n_jobs':[-1]}
```

SVC

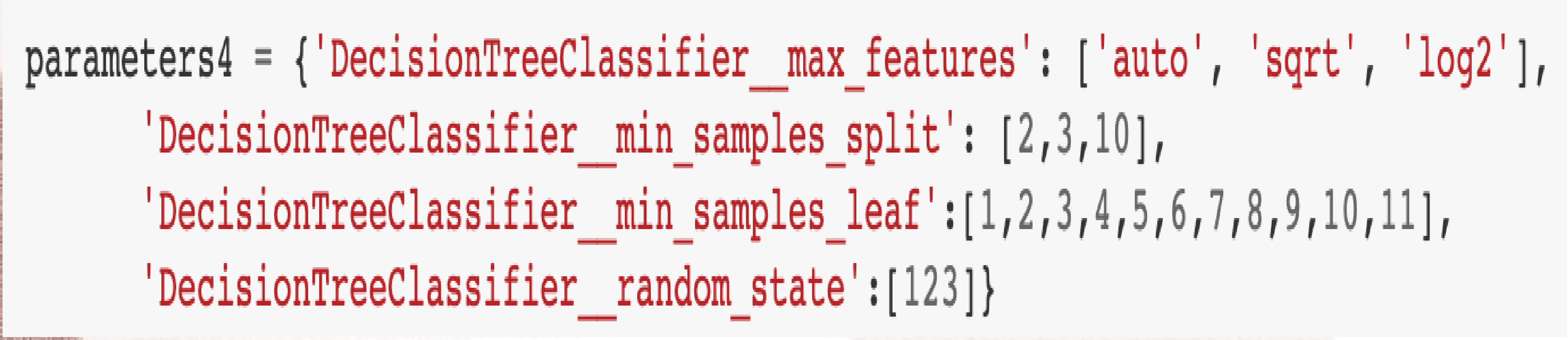
- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.



```
parameters3 = {'SVC__C': [1, 10],
         'SVC__kernel': ['linear','rbf'],
                'SVC__random_state':[123]}
```

Decision Tree Classifier

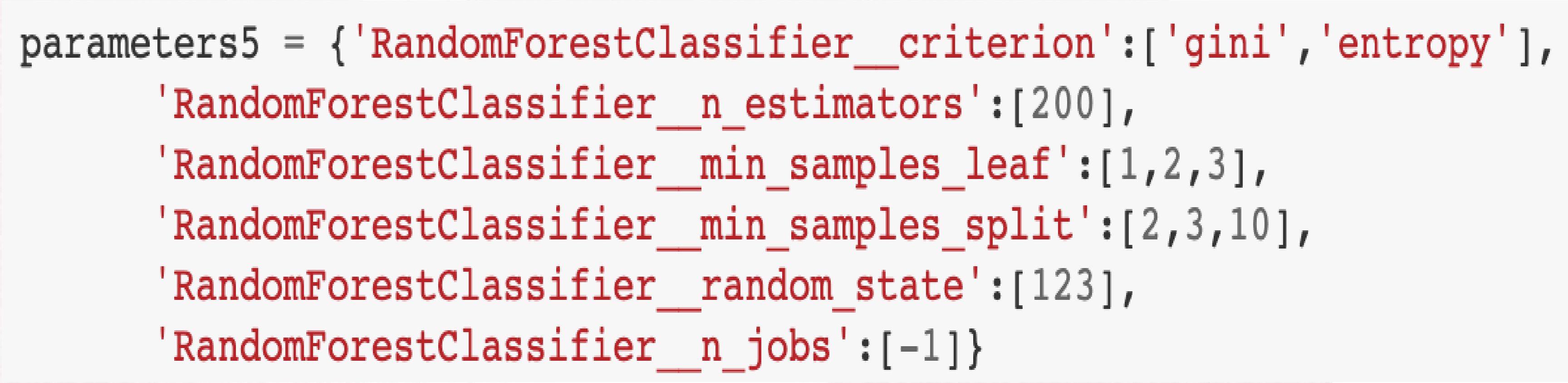
- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.



```
parameters4 = {'DecisionTreeClassifier__max_features': ['auto', 'sqrt', 'log2'],
      'DecisionTreeClassifier__min_samples_split': [2,3,10],
      'DecisionTreeClassifier__min_samples_leaf':[1,2,3,4,5,6,7,8,9,10,11],
      'DecisionTreeClassifier__random_state':[123]}
```

Random Forest Classifier

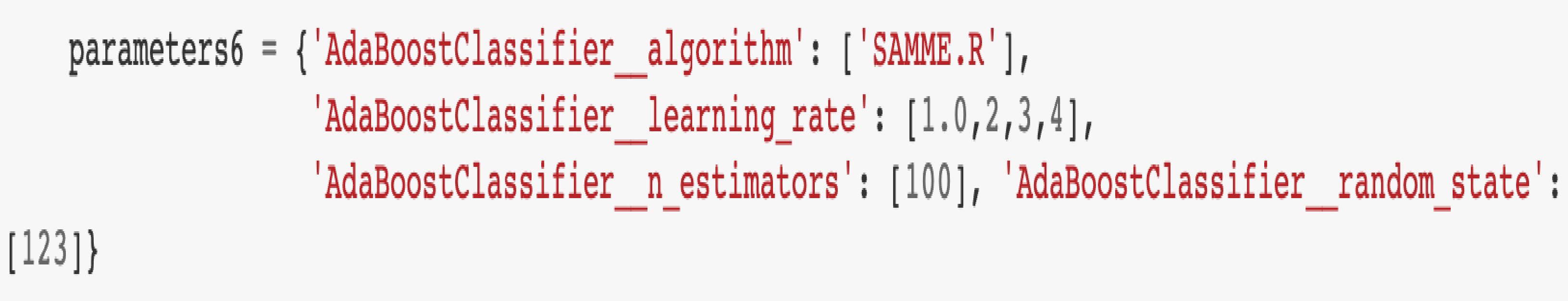
- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.



```
parameters5 = {'RandomForestClassifier__criterion':['gini','entropy'],
      'RandomForestClassifier__n_estimators':[200],
      'RandomForestClassifier__min_samples_leaf':[1,2,3],
      'RandomForestClassifier__min_samples_split':[2,3,10],
      'RandomForestClassifier__random_state':[123],
      'RandomForestClassifier__n_jobs':[-1]}
```

Ada Boost Classifier

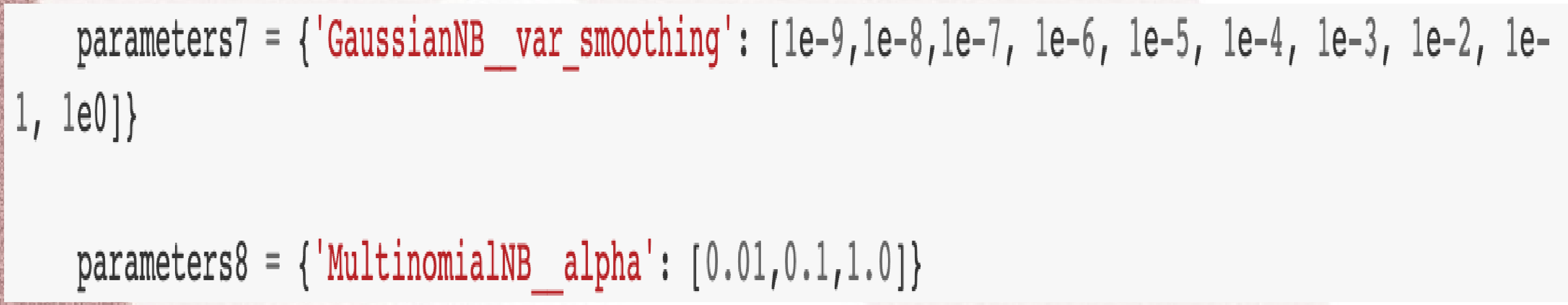
- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.



```
   parameters6 = {'AdaBoostClassifier__algorithm': ['SAMME.R'],
                  'AdaBoostClassifier__learning_rate': [1.0,2,3,4],
                  'AdaBoostClassifier__n_estimators': [100], 'AdaBoostClassifier__random_state':
[123]}
```

Gaussian and Multinomial Naive Bayes

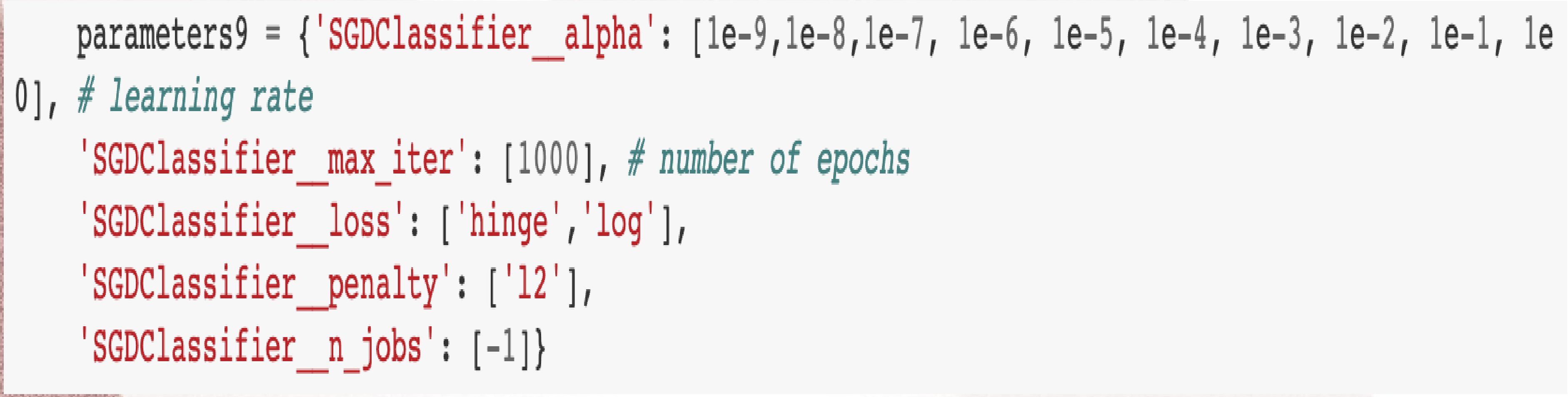
- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.



```
    parameters7 = {'GaussianNB__var_smoothing': [1e-9,1e-8,1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-
1, 1e0]}


    parameters8 = {'MultinomialNB__alpha': [0.01,0.1,1.0]}
```

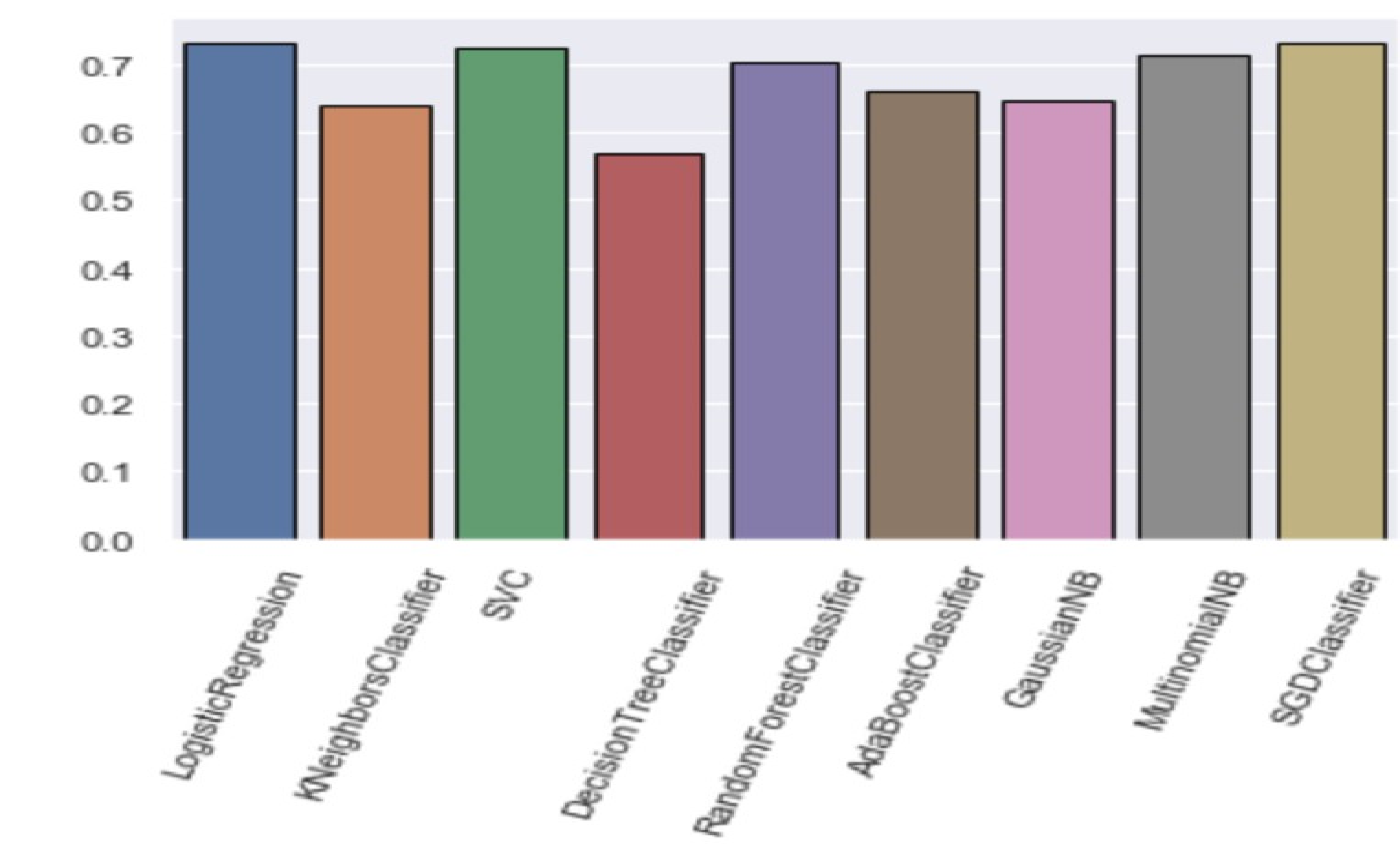
SGD Classifier

- Split into training(80%) and testing set(20%).
- Hyperparameter tuning using GridSearch.

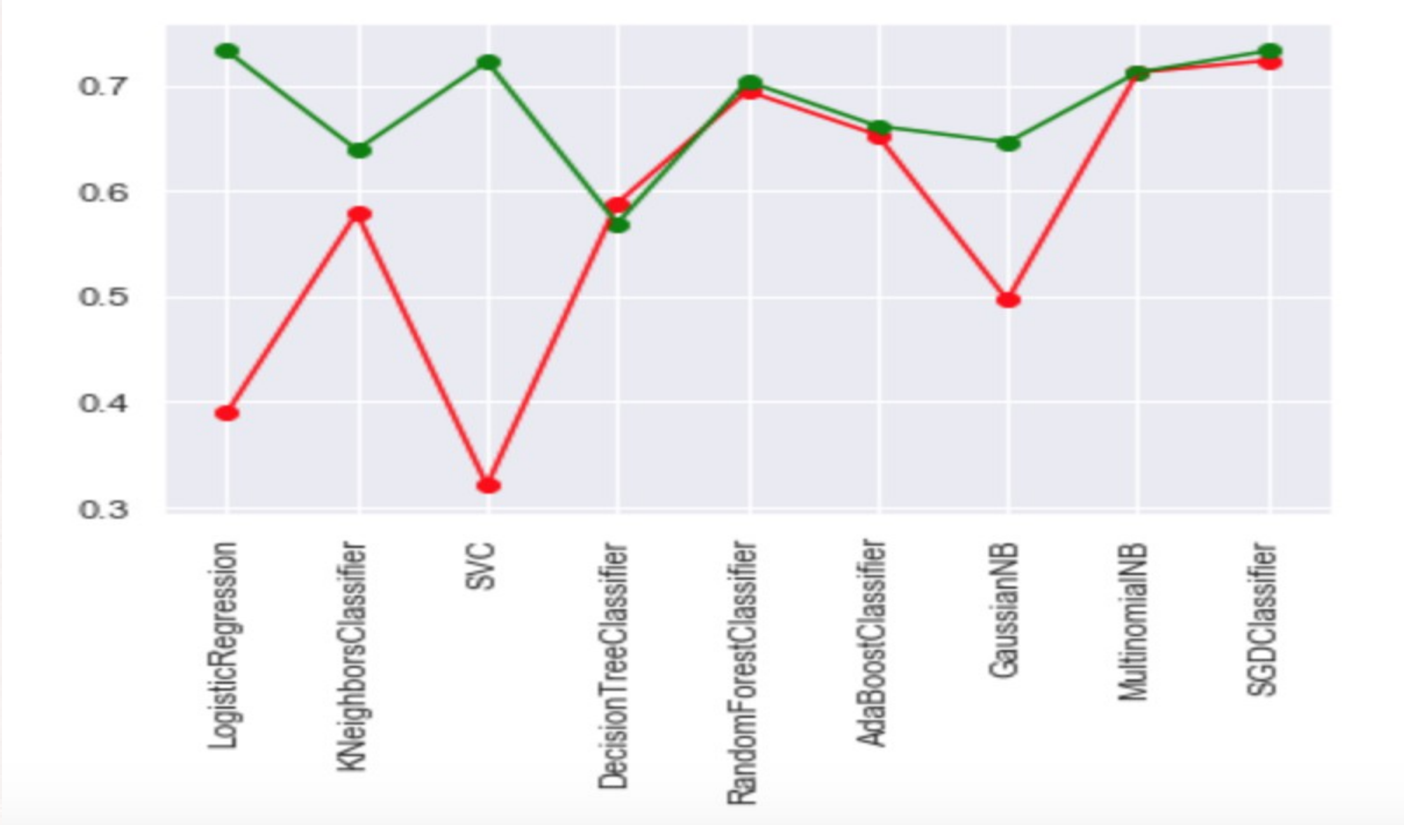


```
    parameters9 = {'SGDClassifier__alpha': [1e-9,1e-8,1e-7, 1e-6, 1e-5, 1e-4, 1e-3, 1e-2, 1e-1, 1e
0], # learning rate
    'SGDClassifier__max_iter': [1000], # number of epochs
    'SGDClassifier__loss': ['hinge','log'],
    'SGDClassifier__penalty': ['l2'],
    'SGDClassifier__n_jobs': [-1]}
```

Model Accuracies



Comparisons



- Hyperparameter-tuned results(green) against non-hyperparameter-tuned results(red).