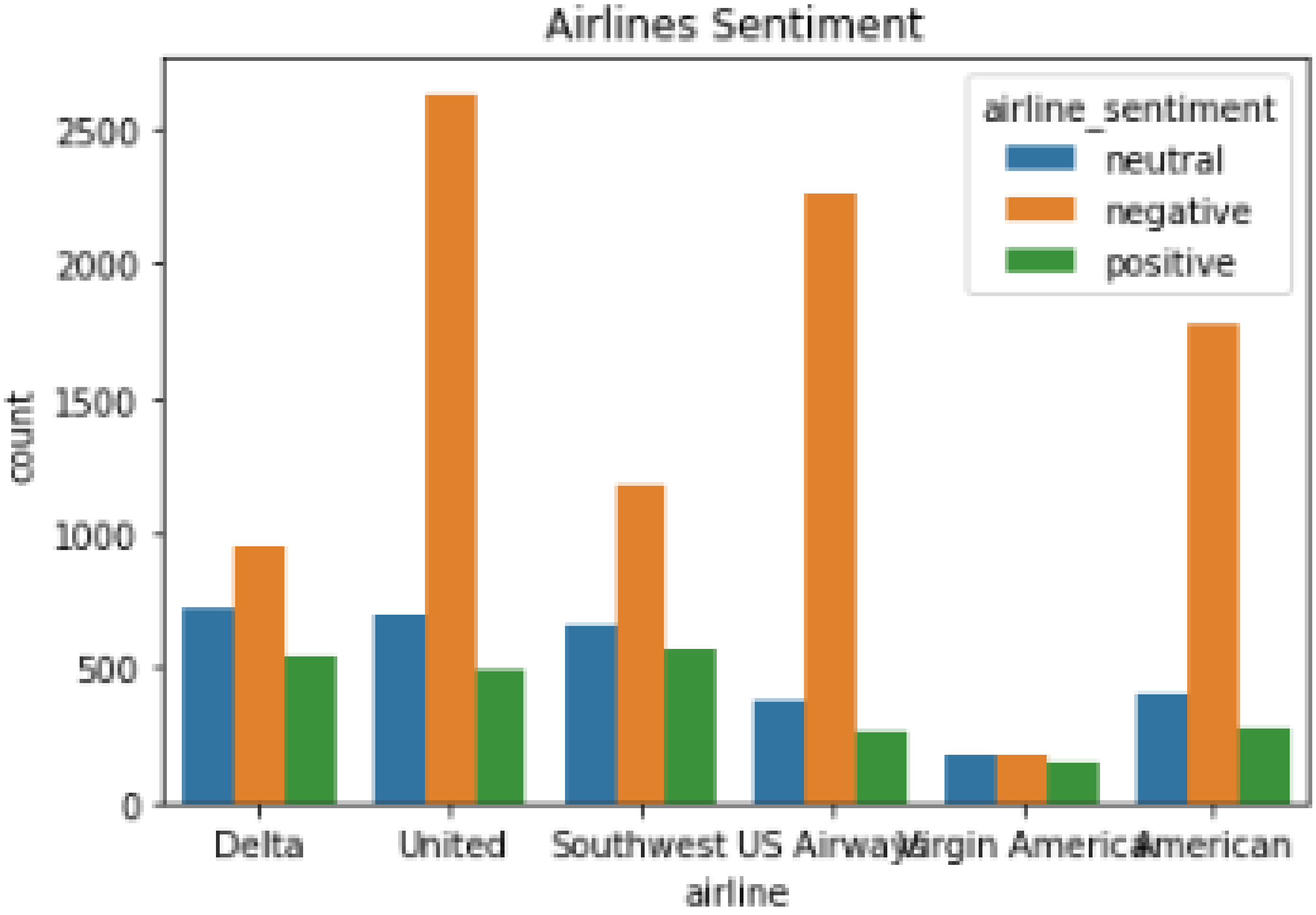
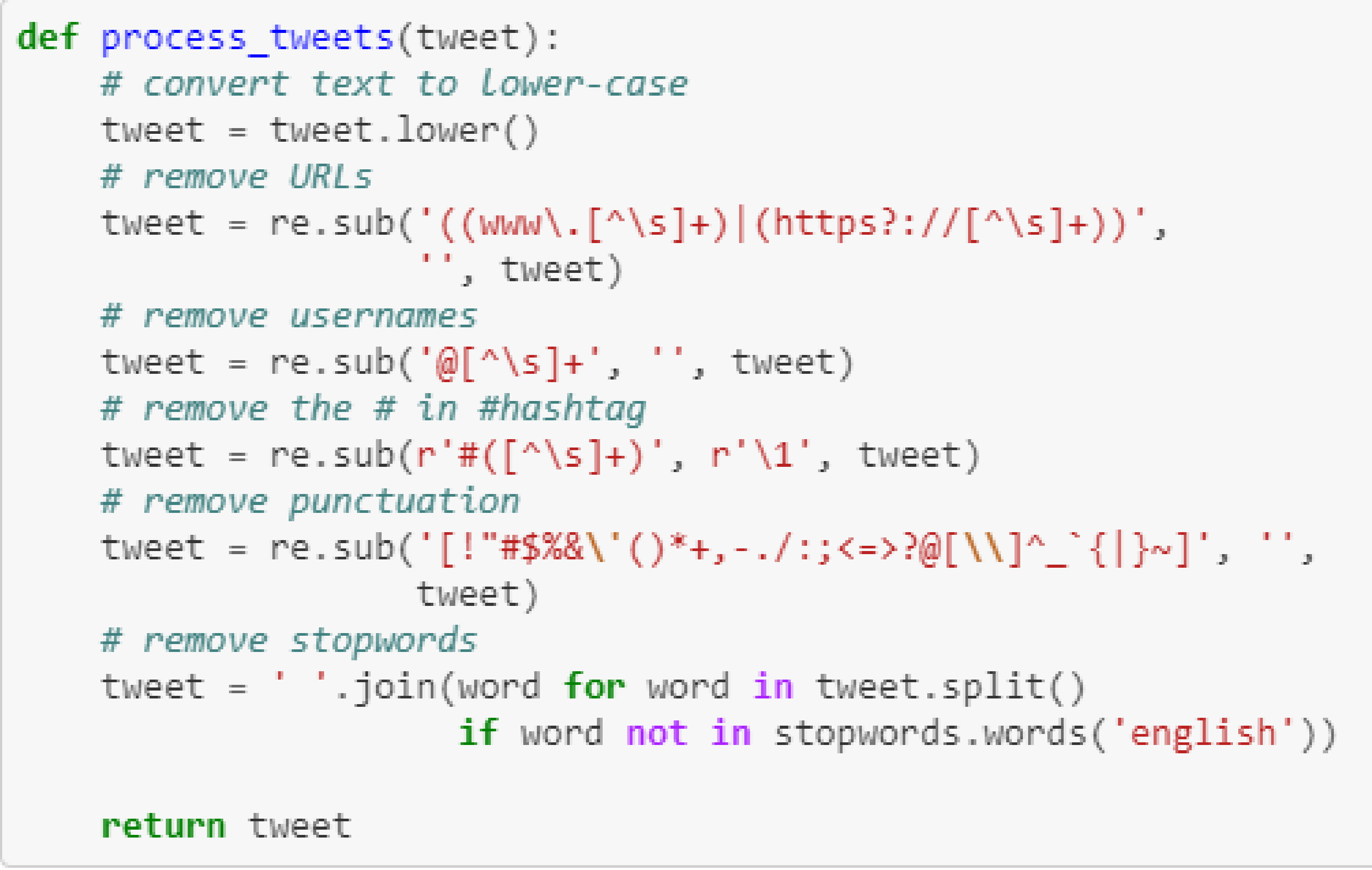
Twitter US Airline Sentiment Analysis

By William Dew

DATA


Airlines Sentiment
airline_sentiment
neutral
negative
positive
count
0
500
1000
1500
2000
2500
Delta
United
Southwest
US Airways
Virgin America
American
airline

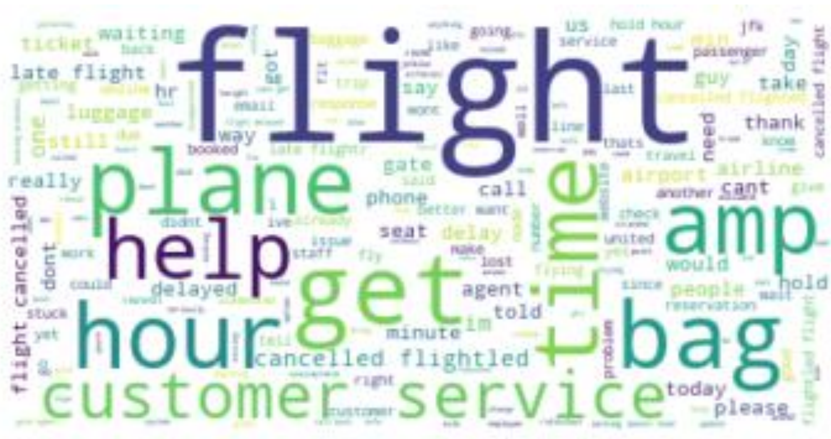
DATA WRANGLING



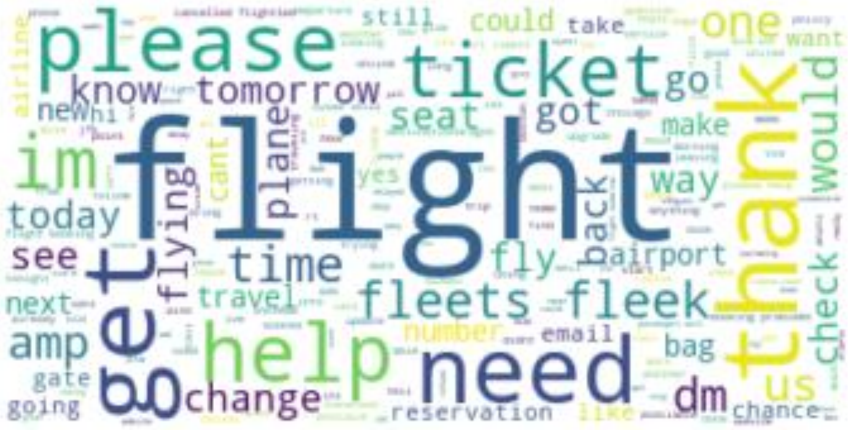
```
def process_tweets(tweet):
    # convert text to lower-case
    tweet = tweet.lower()
    # remove URLs
    tweet = re.sub('((www\.[^\s]+)|(https?://[^\s]+))',
                   '', tweet)
    # remove usernames
    tweet = re.sub('@[^\s]+', '', tweet)
    # remove the # in #hashtag
    tweet = re.sub(r'#([^\s]+)', r'\1', tweet)
    # remove punctuation
    tweet = re.sub('[!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~]', '',
                   tweet)
    # remove stopwords
    tweet = ' '.join(word for word in tweet.split()
                     if word not in stopwords.words('english'))

    return tweet
```

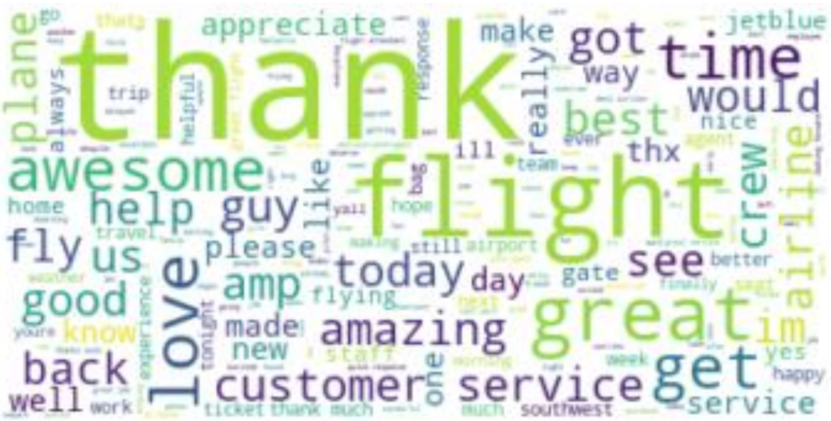
WORD BALLOONS



Negative



Neutral



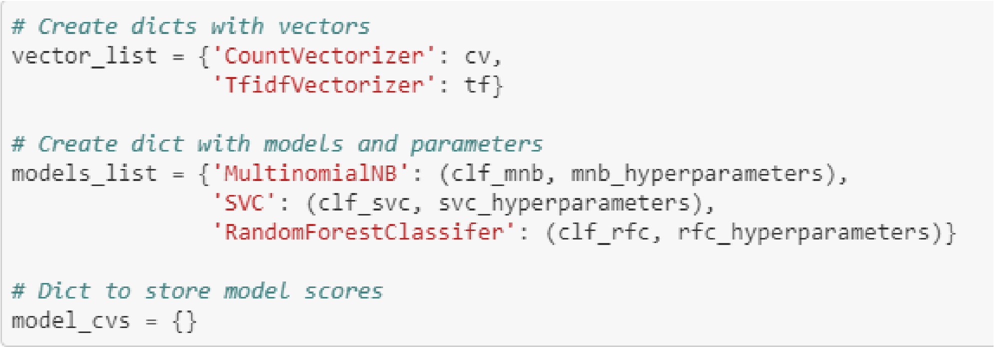
Positive

MODELS

Vectorizers
CountVectorizer
TfidfVectorizer

Classifiers
MultinomialNB
SVC
Random Forest

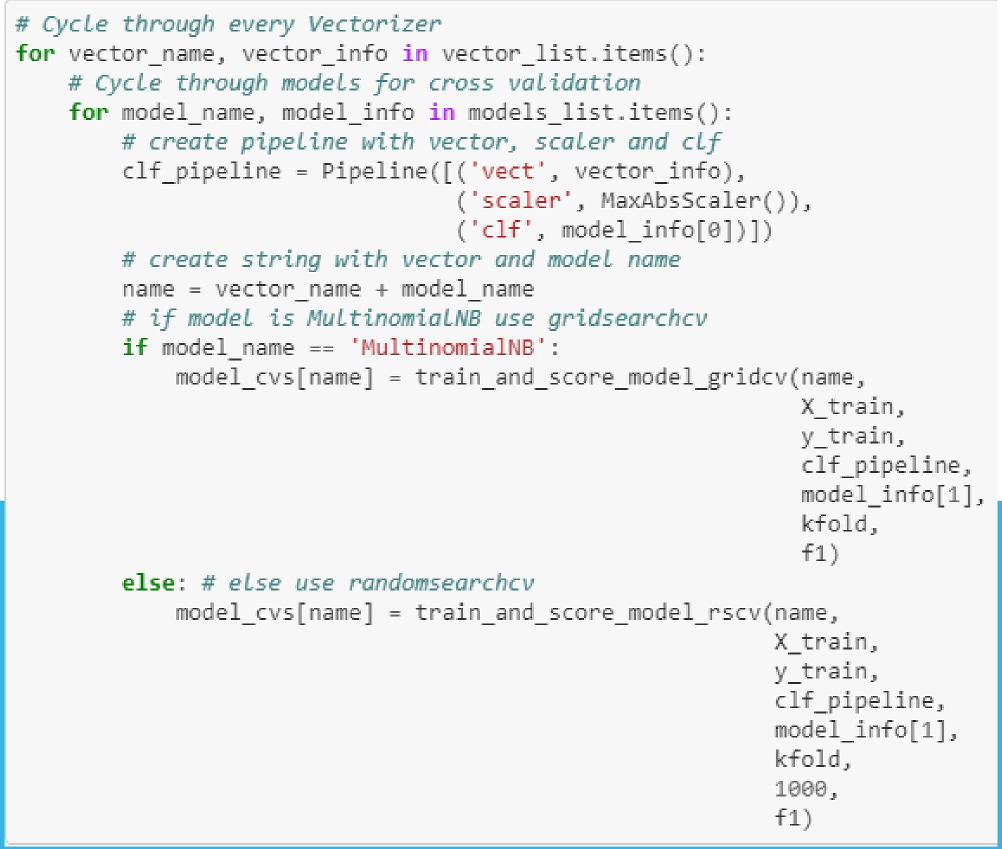
Cross Validation
Grid Search CV
Random Search CV



```python
# Create dicts with vectors
vector_list = {'CountVectorizer': cv,
               'TfidfVectorizer': tf}

# Create dict with models and parameters
models_list = {'MultinomialNB': (clf_mnb, mnb_hyperparameters),
               'SVC': (clf_svc, svc_hyperparameters),
               'RandomForestClassifer': (clf_rfc, rfc_hyperparameters)}

# Dict to store model scores
model_cvs = {}
```

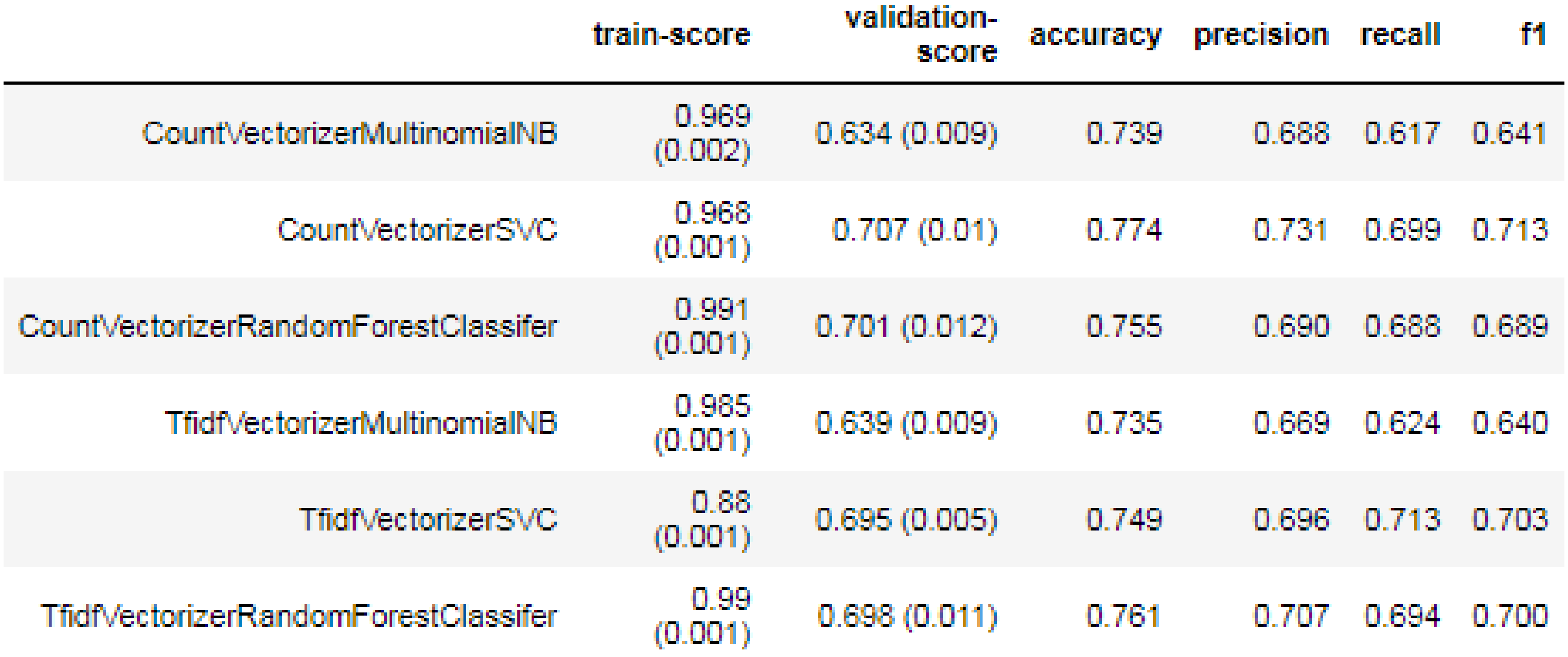


```python
# Cycle through every Vectorizer
for vector_name, vector_info in vector_list.items():
    # Cycle through models for cross validation
    for model_name, model_info in models_list.items():
        # create pipeline with vector, scaler and clf
        clf_pipeline = Pipeline([('vect', vector_info),
                                 ('scaler', MaxAbsScaler()),
                                 ('clf', model_info[0])])
        # create string with vector and model name
        name = vector_name + model_name
        # if model is MultinomialNB use gridsearchcv
        if model_name == 'MultinomialNB':
            model_cvs[name] = train_and_score_model_gridcv(name,
                                                           X_train,
                                                           y_train,
                                                           clf_pipeline,
                                                           model_info[1],
                                                           kfold,
                                                           f1)

        else: # else use randomsearchcv
            model_cvs[name] = train_and_score_model_rscv(name,
                                                         X_train,
                                                         y_train,
                                                         clf_pipeline,
                                                         model_info[1],
                                                         kfold,
                                                         1000,
                                                         f1)
```

RESULTS

- F1 score for Train-score and Validation Score from Cross Validation of Train Set
- Metric Scores for Test Set with Best Estimator Found



	train-score	validation-score	accuracy	precision	recall	f1
CountVectorizerMultinomialNB	0.969 (0.002)	0.634 (0.009)	0.739	0.688	0.617	0.641
CountVectorizerSVC	0.968 (0.001)	0.707 (0.01)	0.774	0.731	0.699	0.713
CountVectorizerRandomForestClassifer	0.991 (0.001)	0.701 (0.012)	0.755	0.690	0.688	0.689
TfidfVectorizerMultinomialNB	0.985 (0.001)	0.639 (0.009)	0.735	0.669	0.624	0.640
TfidfVectorizerSVC	0.88 (0.001)	0.695 (0.005)	0.749	0.696	0.713	0.703
TfidfVectorizerRandomForestClassifer	0.99 (0.001)	0.698 (0.011)	0.761	0.707	0.694	0.700