README.md

Assignment №2. Introduction to Big Data. Stream Processing with Spark

GitHub repository

Contents

Team Members and Roles

Group: DS-02

- **Artem Bakhanov**

Team management, stream reading, model creation and tuning (linear SVC), code refactoring

- **Dmitry Podpryatov**

Model creation (random forest) and tuning (all but SVC), data preprocessing, report

- **Kamil Kamaliev**

Model creation (logistic regression), stream, data preprocessing and merging datasets, report

- **Marina Nikolaeva**

Model evaluation (scripts and labeling), support internationality of the team, report

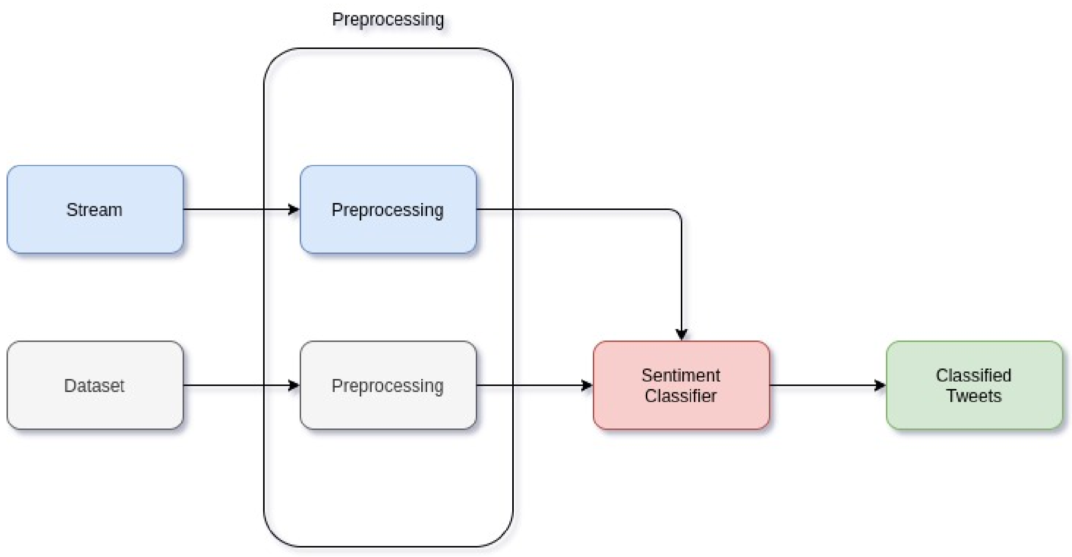
The Problem

What: Given a stream of tweets, analyze their sentiment using Apache Spark. The task is to build a model which will predict whether a given tweet possesses a positive or negative emotions.

Why: First of all, to practice our Spark skills. Second of all, this can be useful for companies because they can get customers' opinion about their products — and thus fix what's bad or understand what people like.

The Solution

We divided the work into stages and created a pipeline that can be summarized as follows:



1. Selecting data

We selected datasets from assignment description that we thought would suit us the most, and merged them together. Check out references for the sources that we selected. Basically, we were looking for a pair (sentiment, sentence/paragraph/text) for our classification models.

You can find `.ipynb` notebook where the datasets were merged together in the `merging datasets` folder. The result of the merge is situated in the `data` folder. The data was split on the train and test at this point, so we will not have to do it at the preprocessing stage.

2. Preprocessing

Before feeding the data to the classification models, we need to preprocess it. Since both our data and tweets from the stream have similar format, the preprocessing module will be the same for the dataset and for the stream.

The preprocessing consists of several parts:

```
1. Get rid of unimportant information such as tags (@elonmusk, #twitter), hyperlinks
(https://google.com), html tags (<br /><br />), and repeated symbols (amaaaazing).

2. Tokenization - split text into words (lowercasing is done in this step as well)

3. Remove "stopwords" - words that do not influence the sentiment of the text (they, to, be,
because, that, etc.)

4. Transform words to vector model so that it can be fed into the classification model (Word2Vec)
```

`Stopwords` were taken from the default set of `org.apache.spark.ml.feature.StopWordsRemover` . We decided to keep words `against` , `no` , `not` as they may help to identify such negative tweets as `not cool!` .

As it was mentioned before, we have already split the data into train and test samples, so we do not have to do it here.

Training dataset is preprocessed by calling `prep_train(df_train)` and testing dataset is preprocessed by calling `prep_test(df_test)` . Steps are almost the same except the ones related to `word2vec` .

`Word2Vec` is trained on our training dataset in the preprocessing step. After that, `word2vec` is saved to HDFS. When preprocessing test and stream datasets, we just load the trained `word2vec` and use it. Also, we wanted to try pretrained `word2vec` but each time ran out of memory. Thus, we gave up that idea.

```
IMPORTANT

Do not delete folders/files
- word2VecModel
- test_data.csv
- train_data.csv
```

The preprocessing code is situated at the `src/main/scala` folder within `Preprocessing.scala` and `TrainWord2Vec.scala` .

3. Classification

After the data is preprocessed, it is fed into the classification models. Among all classification models we have chosen Logistic Regression, Random Forest, and Support Vector Clustering (SVC). We also tried to implement Naive Bayes classifier, however it accepted only vectors with nonnegative values, so we got to apply scaling which would mess up the `Word2Vec` , so we rejected this idea.

`ml` and `mllib` packages contain all listed classifiers. We will tune the models and compare their performance on the data. Hyperparameters were tuned with cross validation and grid search for each classifier. Below are the grids and the best parameters for each model. For detailed description of the hyperparameters check out links in the references at the end.

Logistic Regression

Parameter	Grid	Best Value
`elasticNetParam`	0, 0.5, 0.8, 1	0.5
`fitIntercept`	true, false	true
`maxIter`	1000	1000
`regParam`	0, 0.1, 0.2	0.1
`threshold`	0.5	0.5

Random Forest

Parameter	Grid	Best Value
`impurity`	entropy, gini	gini
`maxDepth`	3, 5, 7	7
`numTrees`	20, 40	40

SVC

Parameter	Grid	Best Value
`aggregationDepth`	2, 3	2
`maxIter`	100, 150	150
`threshold`	0.4, 0.5, 0.6	0.6
`regParam`	0, 0.1, 0.2	0

We can assess the quality of our models using F1 Score on our testing dataset. Below is the table with models' F1 Score:

Classifier	F1 Score
Logistic Regression	0.721
Random Forest	0.718
SVC	0.586

After training and testing on corresponding datasets, models are saved to HDFS, so they can be loaded and used while processing stream.

The code for classifiers is situated in the `Classifiers.scala` (cross validation and grid search) and in the `TrainClassifier.scala` the best parameters have been set explicitly to save time.

```
IMPORTANT

Do not delete folders/files
- logRegModel
- randomForestModel
- svcModel
```

4. Processing the Stream

`Stream Address: 10.90.138.32:8989`

Performed steps:

```
1. Load trained models from HDFS.
2. Take the tweet(s) from stream
3. Put them into new empty dataframe called stream.
4. Preprocess it by calling prep_test(stream) function that was mentioned in Preprocessing section
5. For each model, send the preprocced data and acquire predictions.
6. Add column with current timestamp and save to csv.
7. Repeat from step 2 until finish.
```

For each model, results are stored in `stream/modelname/part-r-...`

For example, the output of logistic regression model is stored in `stream/logRegModel/part-r-...`

Outputs are in the form of:

```
timestamp, twit, model prediction
```

Example of the output:

```
2020-10-08T09:59:04.199+03:00,@brendonuriesays helloooo brendon(: reply please,1.0
```

Results may be stored in several files. To make our lives a bit easier, at the end, we unite all the outputs to a single file for each model in `stream_final` folder.

For example, for random forest model, the final file is as follows `stream_final/randomForestModel/part-r-00000...`

And its content is:

```
2020-10-08T04:48:01.310+03:00,@breedimetria mannnn hell yea dat nigga be parkin lot pumpin.....I made a mistake &amp;saw a parkin lot scene in his phone one day! Kilt me,0.0
2020-10-08T00:33:01.783+03:00,@DavidRozansky We had a VERY active BBS community here $ yrs. 100s of 'em. Even 1st national relay FIDO-Nets.  AOL killed BBSing.    LOSS!,0.0
2020-10-08T02:06:01.352+03:00,@Davinche I know  Them times there having a phone that could save more than 10 messages was big so how was I 2 know it wud evolve so quick,0.0
2020-10-08T04:46:00.968+03:00,@breedimetria @blackaricanma alright ladies! I'll be back to chat l8r. Bout to take care of Lil A bcuz he doesnt feel well  ttyl twitches!,0.0
2020-10-08T07:20:01.351+03:00,@dawllyllama Sorry for the trauma - to u AND ur car  maybe coyote thought he was a super genius. But duh roadrunner always has last laugh.,0.0
2020-10-08T04:29:01.256+03:00,@breckpetekaren Not a big winter fan. the reason im still here- love the heat (&amp; job)but missing my nephews grow up  Im torn and stuck,1.0
2020-10-08T10:20:04.114+03:00,@BrennanAnnie - did a nice long stretch post easy spin and knee feeling good  only bummer is missing Britney tonight due to LDN transport!,1.0
...
```

What can be Improved

1. Applying pretrained `Word2Vec` model

Our model is trained on a very small text corpus and can be biased since only twitter and reviews datasets were used.

2. Improving accuracy of the classifier by using advanced deep learning models (e.g. LSTM).

Unfortunately, we did not have enough resources on the cluster

3. Better visualisation for the stream

For example, create a web dashboard where all new tweets will appear, and some aggregated statistics will be shown.

4. Apply stemming or lemmatization in preprocessing step.

We saw some libraries on the internet, but we could not install/configure them to work properly.

5. Better structure of the code.

As we have been working with Scala for less than a month, it is impossible to be familiar with all syntactic capabilities that Scala has.

Testing and Comparing models on Stream

`stream_final/logRegModel` , `stream_final/randomForestModel` , `stream_final/svcModel` folders were downloaded to our local machines.

After that, we put correct labels by our hands and calculated `precision` , `recall` and `f1 score` using the code in `test_scripts`.

You can find labeled stream files in `stream labeled` folder

Model	Precision	Recall	F1
`Logistic Regression`	0.75	0.896	0.81
`Random Forest`	0.82	0.92	0.88
`SVM Classifier`	0.53	0.94	0.68

All in all, `random forest` did the best job obtaining the highest precision and its recall being only 0.02 points behind `SVM` .

Conclusion

To sum up, we got practice with Apache Spark DataFrames, ML models, and Streams as well as improved our skills in Scala. Apache Spark provides great abstractions and interfaces for working with batched data and streams. It was a complicated industry-level project, however by dividing it into small parts and managing responsibilities it did not seem extremely hard, and overall, it was a great experience for all of us.

How to Run

To train the models

```
spark-submit --master yarn --class TrainClassifier sentiment-analysis-spark_2.12-3.0.1_1.0.jar
```

This will create several folders containing corresponding models

```
1. logRegModel
2. randomForestModel
3. svcModel
4. word2VecModel
```

If you want to run the code again, please, delete those folders.

To run the Stream Processing

```
spark-submit --master yarn --class StreamProcesser sentiment-analysis-spark_2.12-3.0.1_1.0.jar "5 minutes" logreg
randforest svc
```

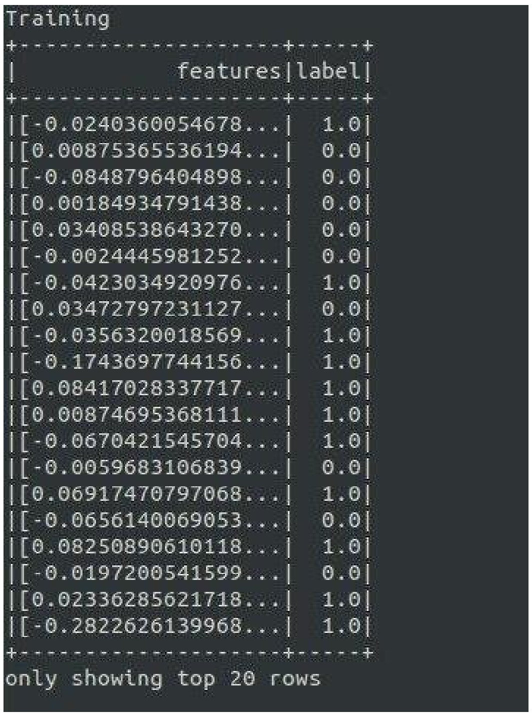
This will create several folders containing records in the form of `timestamp, twit, model prediciton`

```
1. stream
2. stream_final
```

If you want to run the code again, please, delete those folders and make sure that models created in training stage exist.

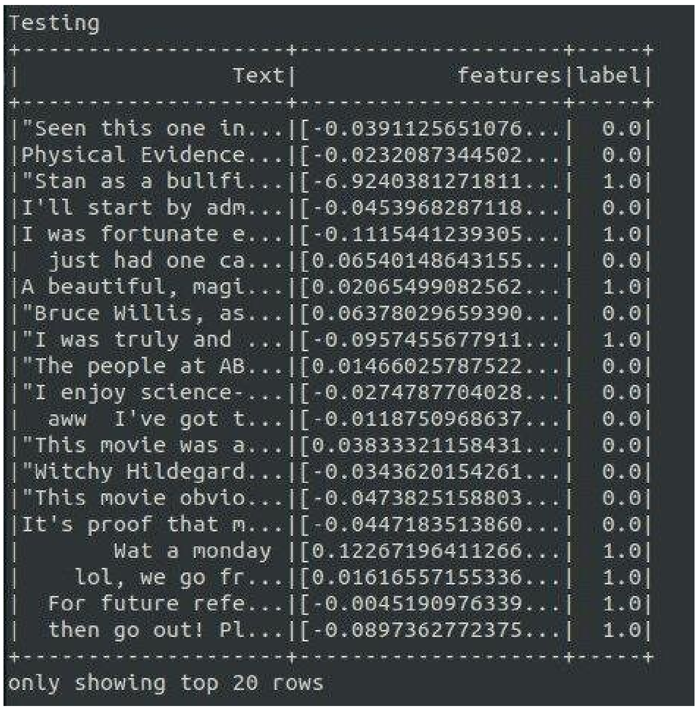
Output

Preprocessed features from training dataset



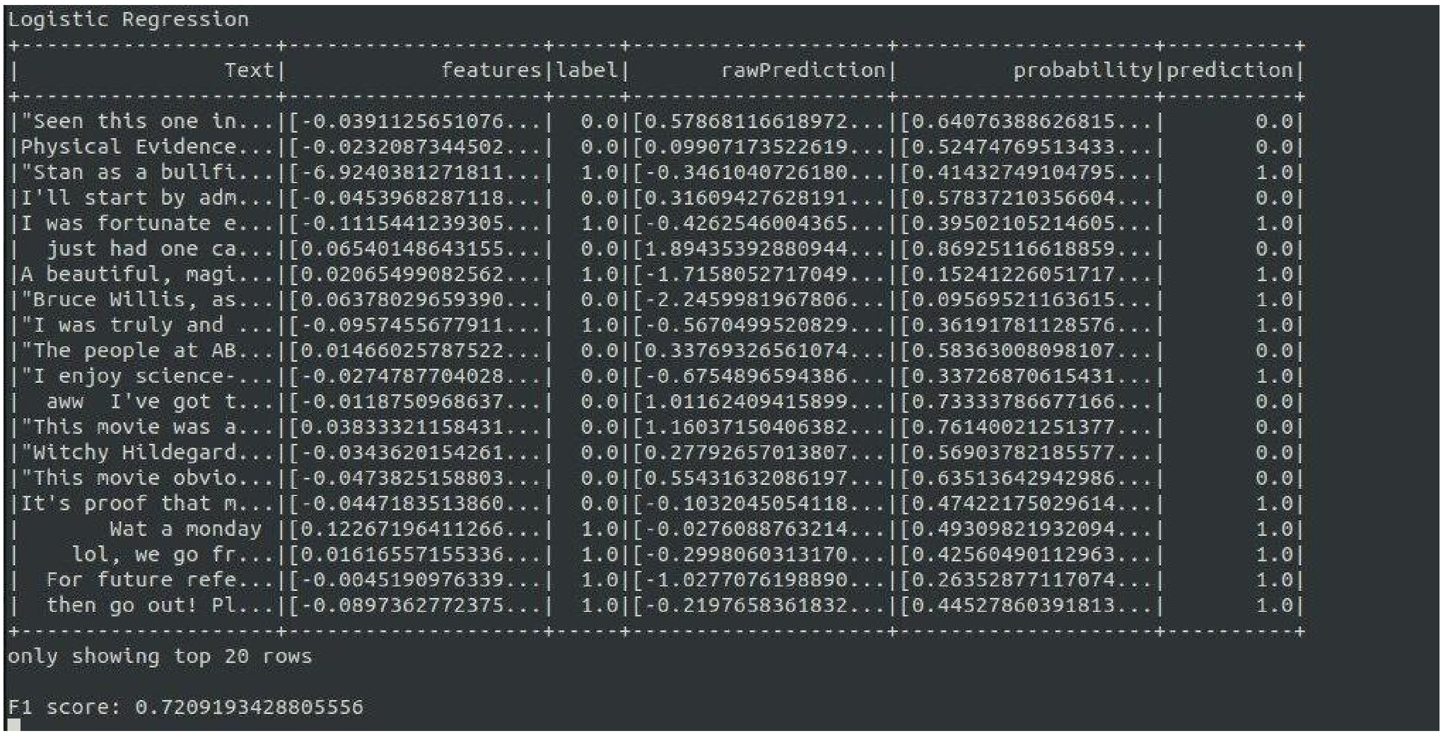
```
Training
+--------------------+-----+
|            features|label|
+--------------------+-----+
|[-0.0240360054678...|  1.0|
|[0.00875365536194...|  0.0|
|[-0.0848796404898...|  0.0|
|[0.00184934791438...|  0.0|
|[0.03408538643270...|  0.0|
|[-0.0024445981252...|  0.0|
|[-0.0423034920976...|  1.0|
|[0.03472797231127...|  0.0|
|[-0.0356320018569...|  1.0|
|[-0.1743697744156...|  1.0|
|[0.08417028337717...|  1.0|
|[0.00874695368111...|  1.0|
|[-0.0670421545704...|  1.0|
|[-0.0059683106839...|  0.0|
|[0.06917470797068...|  1.0|
|[-0.0656140069053...|  0.0|
|[0.08250890610118...|  1.0|
|[-0.0197200541599...|  0.0|
|[0.02336285621718...|  1.0|
|[-0.2822626139968...|  1.0|
+--------------------+-----+
only showing top 20 rows
```

Text and its vector representation from testing dataset



```
Testing
+--------------------+--------------------+-----+
|                Text|            features|label|
+--------------------+--------------------+-----+
|"Seen this one in...|[-0.0391125651076...|  0.0|
|Physical Evidence...|[-0.0232087344502...|  0.0|
|"Stan as a bullfi...|[-6.9240381271811...|  1.0|
|I'll start by adm...|[-0.0453968287118...|  0.0|
|I was fortunate e...|[-0.1115441239305...|  1.0|
|  just had one ca...|[0.06540148643155...|  0.0|
|A beautiful, magi...|[0.02065499082562...|  1.0|
|"Bruce Willis, as...|[0.06378029659390...|  0.0|
|"I was truly and ...|[-0.0957455677911...|  1.0|
|"The people at AB...|[0.01466025787522...|  0.0|
|"I enjoy science-...|[-0.0274787704028...|  0.0|
|  aww  I've got t...|[-0.0118750968637...|  0.0|
|"This movie was a...|[0.03833321158431...|  0.0|
|"Witchy Hildegard...|[-0.0343620154261...|  0.0|
|"This movie obvio...|[-0.0473825158803...|  0.0|
|It's proof that m...|[-0.0447183513860...|  0.0|
|       Wat a monday |[0.12267196411266...|  1.0|
|     lol, we go fr...|[0.01616557155336...|  1.0|
|  For future refe...|[-0.0045190976339...|  1.0|
|  then go out! Pl...|[-0.0897362772375...|  1.0|
+--------------------+--------------------+-----+
only showing top 20 rows
```

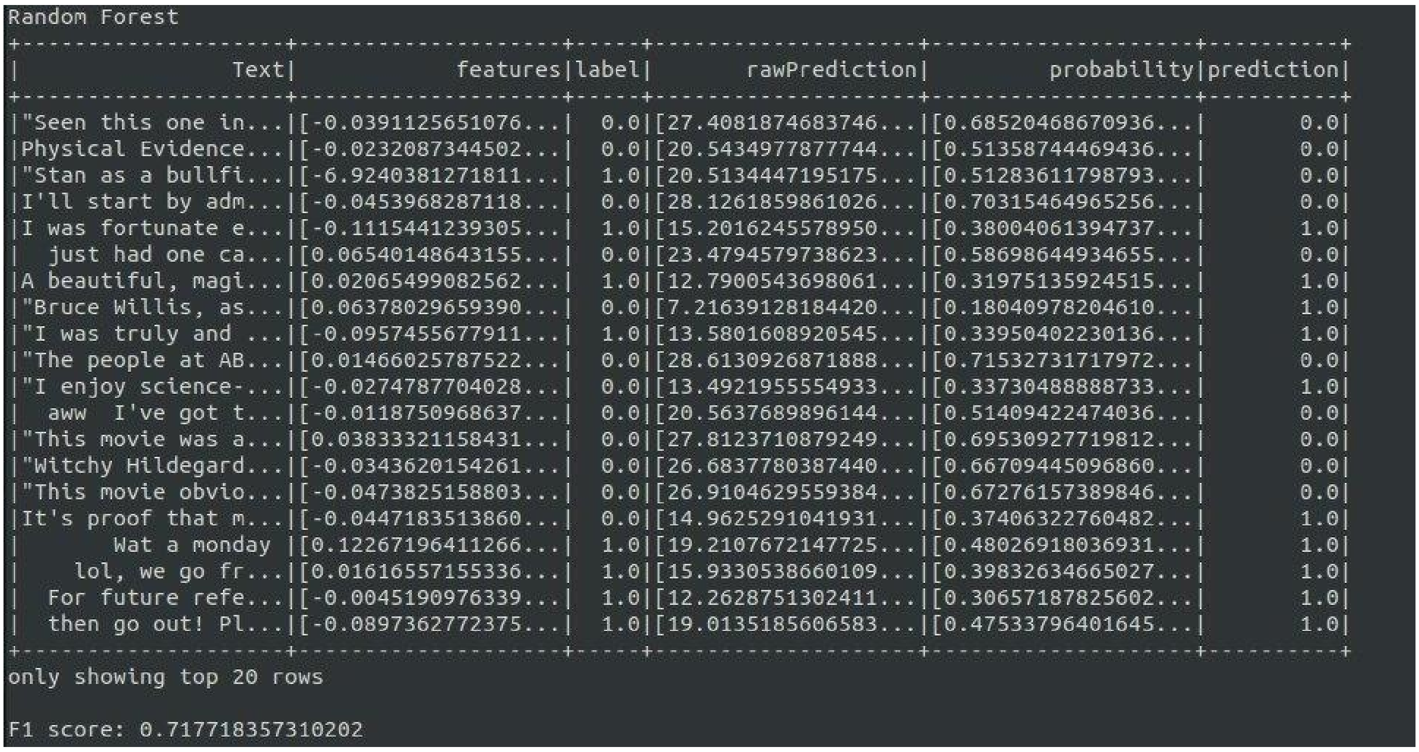
Trained Logistic Regression



```
Logistic Regression
+--------------------+--------------------+-----+--------------------+--------------------+----------+
|                Text|            features|label|       rawPrediction|         probability|prediction|
+--------------------+--------------------+-----+--------------------+--------------------+----------+
|"Seen this one in...|[-0.0391125651076...|  0.0|[0.57868116618972...|[0.64076388626815...|       0.0|
|Physical Evidence...|[-0.0232087344502...|  0.0|[0.09907173522619...|[0.52474769513433...|       0.0|
|"Stan as a bullfi...|[-6.9240381271811...|  1.0|[-0.3461040726180...|[0.41432749104795...|       1.0|
|I'll start by adm...|[-0.0453968287118...|  0.0|[0.31609427628191...|[0.57837210356604...|       0.0|
|I was fortunate e...|[-0.1115441239305...|  1.0|[-0.4262546004365...|[0.39502105214605...|       1.0|
|  just had one ca...|[0.06540148643155...|  0.0|[1.89435392880944...|[0.86925116618859...|       0.0|
|A beautiful, magi...|[0.02065499082562...|  1.0|[-1.7158052717049...|[0.15241226051717...|       1.0|
|"Bruce Willis, as...|[0.06378029659390...|  0.0|[-2.2459981967806...|[0.09569521163615...|       1.0|
|"I was truly and ...|[-0.0957455677911...|  1.0|[-0.5670499520829...|[0.36191781128576...|       1.0|
|"The people at AB...|[0.01466025787522...|  0.0|[0.33769326561074...|[0.58363008098107...|       0.0|
|"I enjoy science-...|[-0.0274787704028...|  0.0|[-0.6754896594386...|[0.33726870615431...|       1.0|
|  aww  I've got t...|[-0.0118750968637...|  0.0|[1.01162409415899...|[0.73333786677166...|       0.0|
|"This movie was a...|[0.03833321158431...|  0.0|[1.16037150406382...|[0.76140021251377...|       0.0|
|"Witchy Hildegard...|[-0.0343620154261...|  0.0|[0.27792657013807...|[0.56903782185577...|       0.0|
|"This movie obvio...|[-0.0473825158803...|  0.0|[0.55431632086197...|[0.63513642942986...|       0.0|
|It's proof that m...|[-0.0447183513860...|  0.0|[-0.1032045054118...|[0.47422175029614...|       1.0|
|       Wat a monday |[0.12267196411266...|  1.0|[-0.0276088763214...|[0.49309821932094...|       1.0|
|    lol, we go fr...|[0.01616557155336...|  1.0|[-0.2998060313170...|[0.42560490112963...|       1.0|
|  For future refe...|[-0.0045190976339...|  1.0|[-1.0277076198890...|[0.26352877117074...|       1.0|
|  then go out! Pl...|[-0.0897362772375...|  1.0|[-0.2197658361832...|[0.44527860391813...|       1.0|
+--------------------+--------------------+-----+--------------------+--------------------+----------+
only showing top 20 rows

F1 score: 0.7209193428805556
```

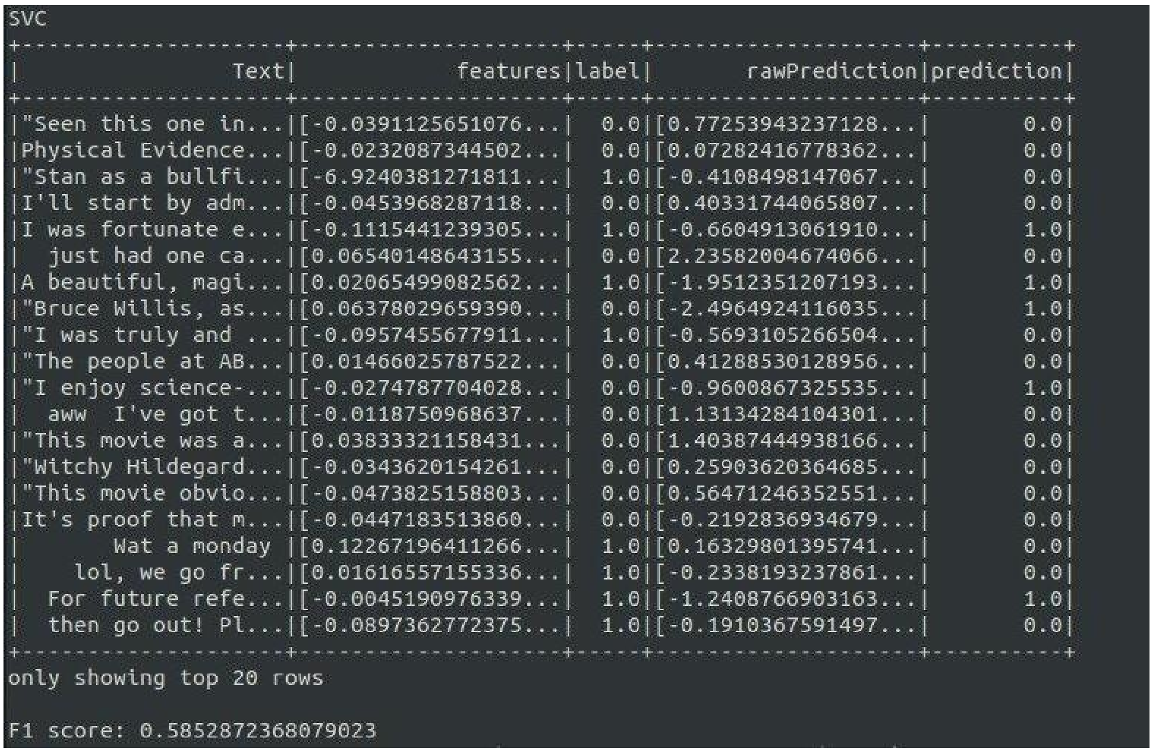
Trained Random Forest



```
Random Forest
+--------------------+--------------------+-----+--------------------+--------------------+----------+
|                Text|            features|label|       rawPrediction|         probability|prediction|
+--------------------+--------------------+-----+--------------------+--------------------+----------+
|"Seen this one in...|[-0.0391125651076...|  0.0|[27.4081874683746...|[0.68520468670936...|       0.0|
|Physical Evidence...|[-0.0232087344502...|  0.0|[20.5434977877744...|[0.51358744469436...|       0.0|
|"Stan as a bullfi...|[-6.9240381271811...|  1.0|[20.5134447195175...|[0.51283611798793...|       0.0|
|I'll start by adm...|[-0.0453968287118...|  0.0|[28.1261859861026...|[0.70315464965256...|       0.0|
|I was fortunate e...|[-0.1115441239305...|  1.0|[15.2016245578950...|[0.38004061394737...|       1.0|
|  just had one ca...|[0.06540148643155...|  0.0|[23.4794579738623...|[0.58698644934655...|       0.0|
|A beautiful, magi...|[0.02065499082562...|  1.0|[12.7900543698061...|[0.31975135924515...|       1.0|
|"Bruce Willis, as...|[0.06378029659390...|  0.0|[7.21639128184420...|[0.18040978204610...|       1.0|
|"I was truly and ...|[-0.0957455677911...|  1.0|[13.5801608920545...|[0.33950402230136...|       1.0|
|"The people at AB...|[0.01466025787522...|  0.0|[28.6130926871888...|[0.71532731717972...|       0.0|
|"I enjoy science-...|[-0.0274787704028...|  0.0|[13.4921955554933...|[0.33730488888733...|       1.0|
|  aww  I've got t...|[-0.0118750968637...|  0.0|[20.5637689896144...|[0.51409422474036...|       0.0|
|"This movie was a...|[0.03833321158431...|  0.0|[27.8123710879249...|[0.69530927719812...|       0.0|
|"Witchy Hildegard...|[-0.0343620154261...|  0.0|[26.6837780387440...|[0.66709445096860...|       0.0|
|"This movie obvio...|[-0.0473825158803...|  0.0|[26.9104629559384...|[0.67276157389846...|       0.0|
|It's proof that m...|[-0.0447183513860...|  0.0|[14.9625291041931...|[0.37406322760482...|       1.0|
|       Wat a monday |[0.12267196411266...|  1.0|[19.2107672147725...|[0.48026918036931...|       1.0|
|    lol, we go fr...|[0.01616557155336...|  1.0|[15.9330538660109...|[0.39832634665027...|       1.0|
|  For future refe...|[-0.0045190976339...|  1.0|[12.2628751302411...|[0.30657187825602...|       1.0|
|  then go out! Pl...|[-0.0897362772375...|  1.0|[19.0135185606583...|[0.47533796401645...|       1.0|
+--------------------+--------------------+-----+--------------------+--------------------+----------+
only showing top 20 rows

F1 score: 0.717718357310202
```

Trained SVC



```
SVC
+--------------------+--------------------+-----+--------------------+----------+
|                Text|            features|label|       rawPrediction|prediction|
+--------------------+--------------------+-----+--------------------+----------+
|"Seen this one in...|[-0.0391125651076...|  0.0|[0.77253943237128...|       0.0|
|Physical Evidence...|[-0.0232087344502...|  0.0|[0.07282416778362...|       0.0|
|"Stan as a bullfi...|[-6.9240381271811...|  1.0|[-0.4108498147067...|       0.0|
|I'll start by adm...|[-0.0453968287118...|  0.0|[0.40331744065807...|       0.0|
|I was fortunate e...|[-0.1115441239305...|  1.0|[-0.6604913061910...|       1.0|
|  just had one ca...|[0.06540148643155...|  0.0|[2.23582004674066...|       0.0|
|A beautiful, magi...|[0.02065499082562...|  1.0|[-1.9512351207193...|       1.0|
|"Bruce Willis, as...|[0.06378029659390...|  0.0|[-2.4964924116035...|       1.0|
|"I was truly and ...|[-0.0957455677911...|  1.0|[-0.5693105266504...|       0.0|
|"The people at AB...|[0.01466025787522...|  0.0|[0.41288530128956...|       0.0|
|"I enjoy science-...|[-0.0274787704028...|  0.0|[-0.9600867325535...|       1.0|
|  aww  I've got t...|[-0.0118750968637...|  0.0|[1.13134284104301...|       0.0|
|"This movie was a...|[0.03833321158431...|  0.0|[1.40387444938166...|       0.0|
|"Witchy Hildegard...|[-0.0343620154261...|  0.0|[0.25903620364685...|       0.0|
|"This movie obvio...|[-0.0473825158803...|  0.0|[0.56471246352551...|       0.0|
|It's proof that m...|[-0.0447183513860...|  0.0|[-0.2192836934679...|       0.0|
|       Wat a monday |[0.12267196411266...|  1.0|[0.16329801395741...|       0.0|
|    lol, we go fr...|[0.01616557155336...|  1.0|[-0.2338193237861...|       0.0|
|  For future refe...|[-0.0045190976339...|  1.0|[-1.2408766903163...|       1.0|
|  then go out! Pl...|[-0.0897362772375...|  1.0|[-0.1910367591497...|       0.0|
+--------------------+--------------------+-----+--------------------+----------+
only showing top 20 rows

F1 score: 0.5852872368079023
```

References

- Assignment description
- Labeled datasets:
  - Large Movie Review Dataset — binary classified sentiment dataset that contains a total of 50000 movie reviews;
  - UCI Sentiment Labelled Sentences — binary classified reviews from imbd.com, amazon.com, and yelp.com;
  - Twitter Sentiment — sentimentally labeled twitter messages.
- Reasons for twitter sentiment analysis
- Classifiers:
  - Logistic Regression and Parameters
  - Random Forest and Parameters
  - Support Vector Clustering (SVC) and Parameters
- Classification Metrics:
  - Precision and Recall
  - F1 Score