MDS Master of Data Science Universidad de Chile

Amazon Review Classifier Using PySpark

CC5212-Massive Data Processing
Group number: 8
Cristóbal Alcázar
Christopher Stears
Yerko Garrido

Main objective

Create a term-frequency inverse document frequency (TF-IDF) feature matrix from Amazon reviews and train a binary sentiment classifier (i.e. positive/negative).

Modest goal: computing the feature matrix of TF-IDF in a distributed way for training, and then using sklearn or some Python library to train the classifier.

Until here, we are happy, but if we have time, we can try to achieve a more ambitious goal:

Do some research to find ways to train a binary classifier end-to-end that can learn in a distributed way using some of the technologies viewed in classes.

About Dataset

The Amazon reviews dataset consists of reviews from amazon. The data span a period of 18 years, including ~35 million reviews up to March 2013.

This subset contains 1,800,000 training samples and 200,000 testing samples in each polarity sentiment.

The CSVs contain `polarity`, `title`, `text`. These 3 columns in them, correspond to class index (1 or 2), review title and review text.

- Polarity - 1 for negative and 2 for positive
- Title - review heading
- Text - review body

Link de información:Kaggle - Amazon Reviews

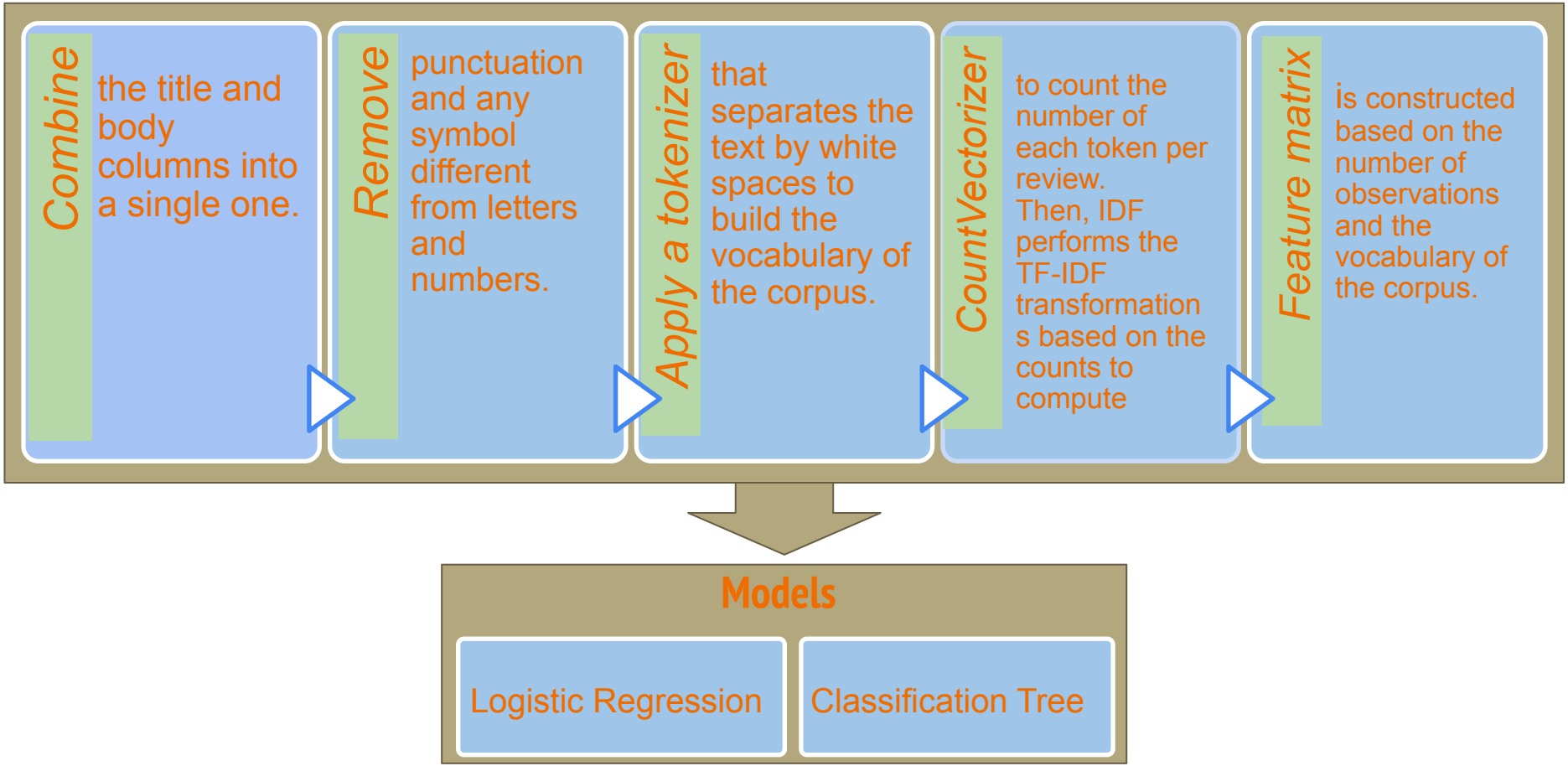
Example Data

Here is an observation of the dataset:

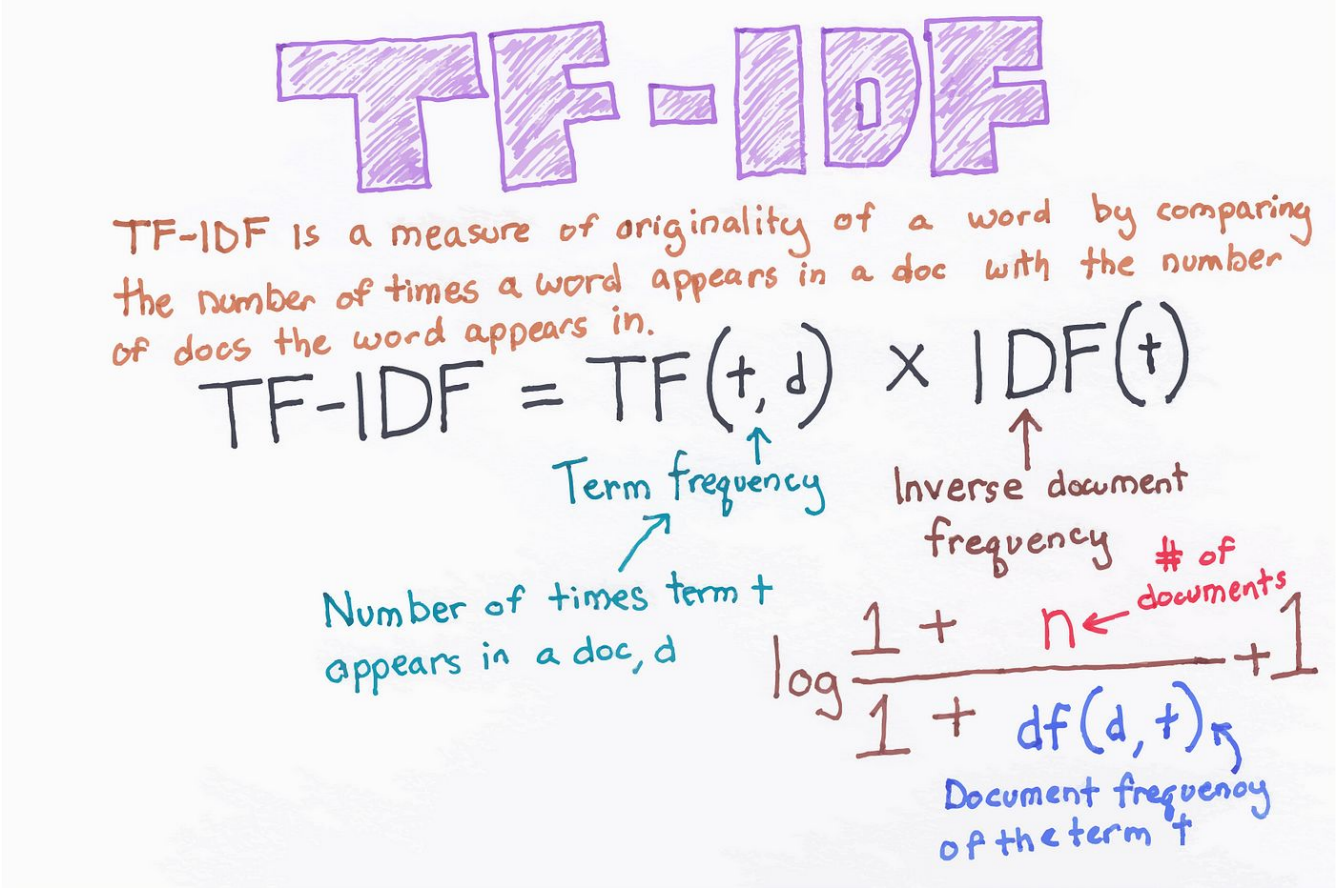
> "2","Makes My Blood Run Red-White-And-Blue","I agree that every American should read this book -- and everybody else for that matter. I don't agree that it's scholarly. Rather, it's a joy to read -- easy to understand even for a person with two master's degrees! Between McElroy's chapter on How American Culture was Formed and Ken Burns' Lewis & Clark, I don't know which makes my blood run red-white-and-bluer. And as a child of the anti-establishment `60s, it's done a lot toward helping me understand why we Americans do what we do. It's the best history book I've ever read, the best history course I've ever taken or taught. I'm buying it for my home library for my grandchildren to use as a resource. We're also using it as a resource for a book on urban planning."

Each observation has three values: the polarity (1 for negative and 2 for positive), the heading and the body of the product review, respectively.

Methodology

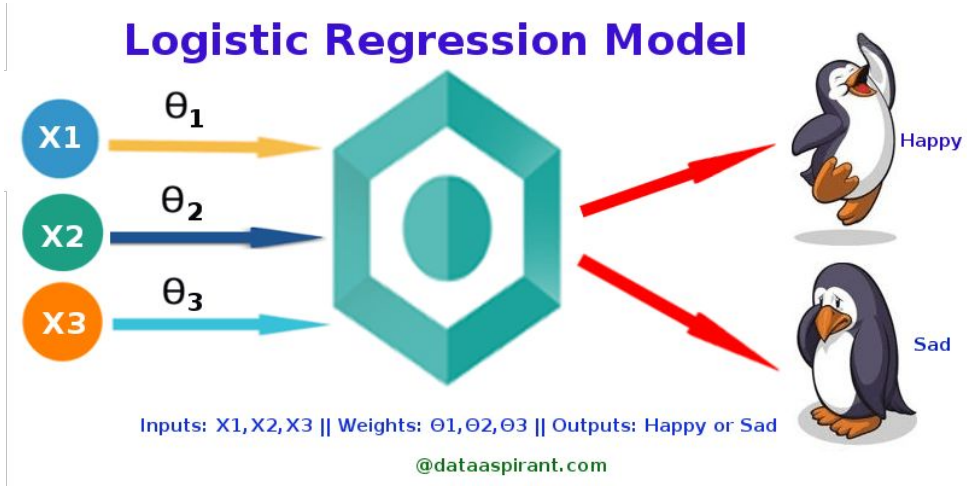


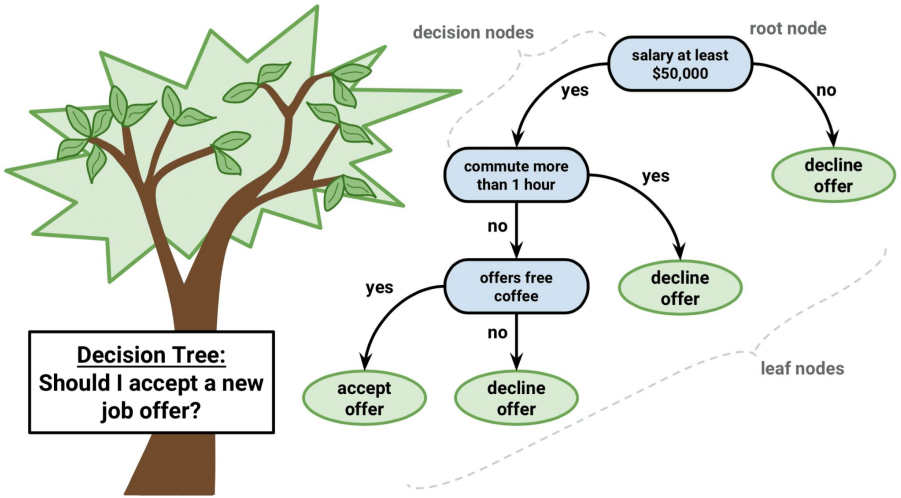
Methodology



Models

We trained two models for analyzing Amazon reviews: Logistic Regression and Classification Tree.





PySpark Pipeline (MLlib): train and inference

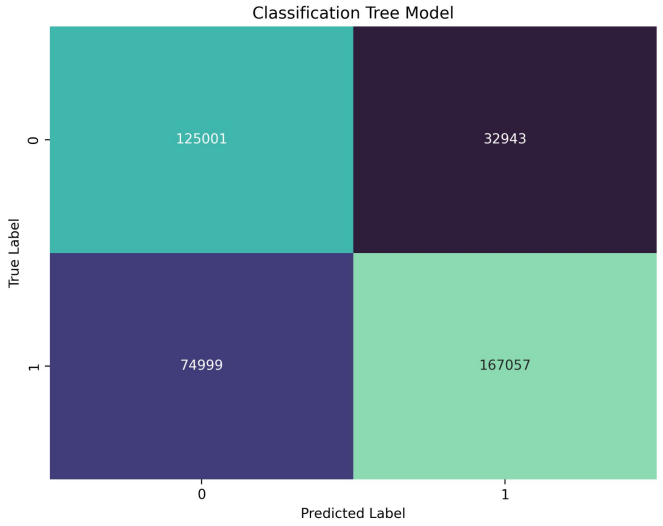
train.py -> using a pipeline with all transformations and fit the model:

```
# Create pipeline for tokenization, count vectorization, IDF, and a classification model
tokenizer = Tokenizer(inputCol='review_text', outputCol='tokens')
countVectorizer = CountVectorizer(inputCol='tokens', outputCol='raw_features')
idf = IDF(inputCol='raw_features', outputCol='transformed_features')
model = LogisticRegression(labelCol='polarity', featuresCol='transformed_features', maxIter=10,
pipeline = Pipeline(stages=[tokenizer, countVectorizer, idf, model])

# Fit the pipeline and save to make inference
pipelineModel = pipeline.fit(data)
```

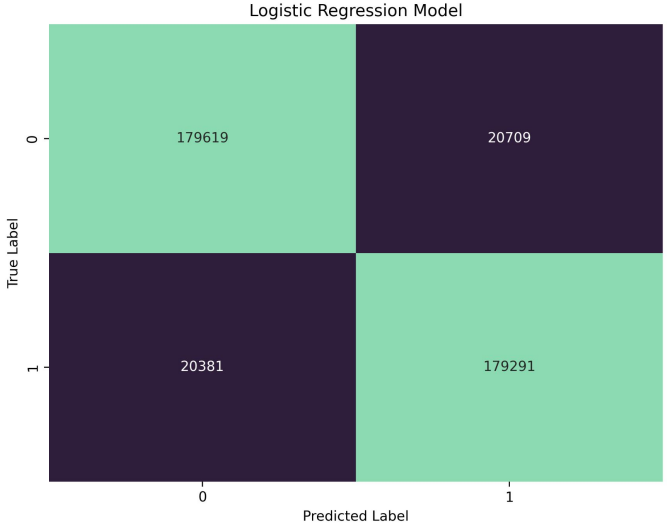
inference.py -> load the fitted pipeline and make predictions on new data

```
# Use the pipeline to get the predictions
predictions = pipeline.transform(new_data)

# Save the output in the folder OUTPUT
#predictions.select('polarity', 'prediction', 'review_text').write.csv(OUTPUT_PATH + 'predictions')
predictions.select('polarity', 'prediction', 'review_text').write.mode('overwrite').csv(OUTPUT_PATH + 'predictions')
```

Results (~400k test set)



73% accuracy, macro-f1 73%



90% accuracy, macro-f1 90%

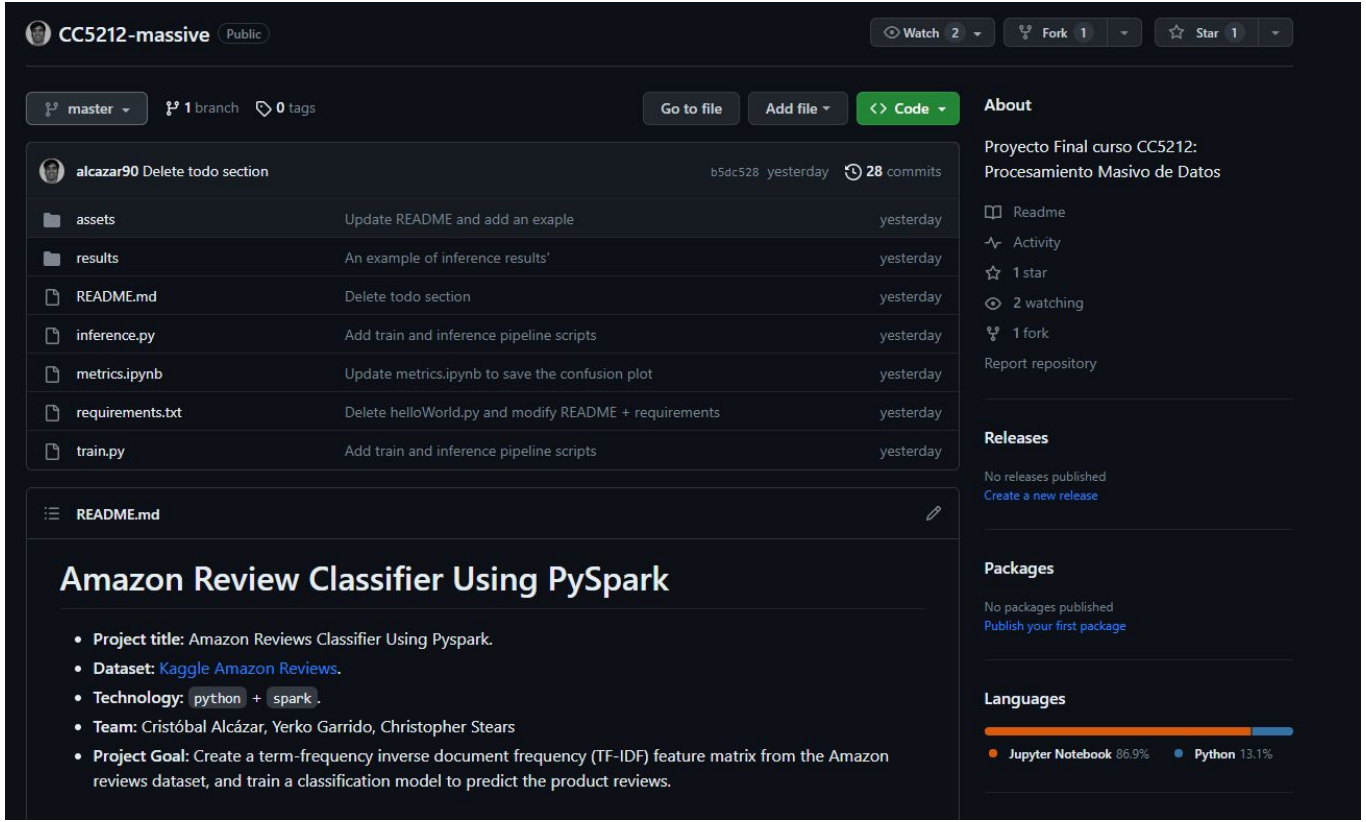
Conclusions

1. PySpark seamlessly integrates Spark functionality with Python, enabling smooth interoperability with other machine learning frameworks and libraries.

2. Spark MLib offers a diverse range of distributed training models, including linear and tree-based models, eliminating the need to transfer intermediate steps between HDFS and the local system.

3. Despite the simplicity of the dataset (balanced with 2 classes), impressive average F1 scores of 90% are achieved even without hyperparameter optimization.

Repository (GitHub)



https://github.com/alcazar90/CC5212-massive

MDS Master of Data Science Universidad de Chile

Amazon Review Classifier Using PySpark

CC5212-Massive Data Processing
Group number: 8
Cristóbal Alcázar
Christopher Stears
Yerko Garrido

Reference


```
@misc{CC5212-massive-manco,
  authors = {Alcázar, Cristóbal}, {Garrido, Yerko}, {Stears, Christopher}
  title = {Project group 8: Amazon Review Classifier Using Pyspark},
  year = {2023},
  publisher = {GitHub},
  journal = {GitHub repository},
  howpublished = {\url{https://github.com/alcazar90/CC5212-massive}},
}
```