Big Data Systems and Techniques

This is an overview of the work that was done for the project, I am adding this as a guideline of what is attached.
I have separated each task in each own folder for simplicity

Repo: https://github.com/manosprom/aueb-big-data-systems-and-techniques

Notes:

- I have used jupyter for all of the tasks so the execution logs are inside the notebook i have exported the execution logs though and i have attached them.
- All the notebooks contain the output so you don't need to run them.

Task 1 - Get the data

For task 1 I have created a managed postgres instance through google cloud console and imported the temp_products data through psql

- Task-1.pdf

I have included print screens of the creation of the managed postgres, the additional steps to connect s01 to the managed postgres and the logs from importing the temp_products

Task 2 - Create the parquet file

For task 2 I have select with spark-ql all the rows with categories that contains 'shoes' and created the parquet file in `hdfs:///data/exercise/shoes.parquet`

- Task-2.pdf: The output of the notebook for connect to managed-pg, selecting the relevant data and creating the parquet file
- Task-2.ipynb: The notebook for connect to managed-pg, selecting the relevant data and creating the parquet file

- print-schema.txt The schema of the selected df
- parquet-ls.png ls of the hdfs folder with the `shoes.parquet` file png.
- parquet-ls.txt txt output

Task 3 - ML

For task 3 I have trained 3 models

The whole execution can be found in

- Task-3.pdf: The output of the notebook for training cross validate and save the 3 models
- Task-3.ipynb: The notebook for training cross validate and save the 3 models

I have extracted the cross validation logs per model for simplicity

1. Logistic Regression (Cross Validated)
    - cross-validation-logistic-regression.txt
    - best-model-logistic-regression
2. Random Forest (Cross Validated)
    - cross-validation-random-forest.txt
    - best-model-random-forest
3. OneVsRestClassifier with the Best Logistic Regression model that I got from step 1
    - best-model-one-vs-rest-logistic-regression

Task 4 - Kafka

For task 4 i have created a twitter developer account and created a stream consumer for the twitter api with the credentials
and push every tweet that i was receiving to the kafka topic that i have created named offers.

You can find the commands used to create the topic on

- Topic Creation
- Topic List

You can find the producer in python file

- Twitter-Kafka-Producer python app

And the notebook that I used to execute it in the end

- Twitter-Kafka-Producer pdf output
- Twitter-Kafka-Producer notebook
- Consumer tweet logs

The logs contain

- the creation date of the tweet to prove that it works
- the tweet text

For filter i have used

```
'shop shoes',
'shopping shoes',
'shopping offers shoes',
'offers shoes',
'sell shoes',
'shoes offer',
'shoes gift'
```

Task 5 - Spark Streaming

Task 4 and and Task 5 notebooks were running in parallel so whatever tweet was retrieved, it was picked by the
streaming app that was running on the notebook of task 5 and was evaluated to a shoe category.

So for task 5 I have:

- Loaded the saved best LogisticRegression from Task 3
- Moved the data transformers that I used on task 3 to preprocess the data
- Created a spark streaming application
- Connected it to kafka and listen on the `offers` topic to stream the tweets produced from Task 4
- For each even I applied the same preprocessing steps and predicted the category of the tweet.
- Appended each prediction to
  - /data/exercise/shoes_predictions.csv
  - /data/exercise/shoes_predictions.parquet

You can find

The spark streaming app

- Task-5-Notebook-pdf
- Task-5-Notebook

along with the extracted logs of its execution

- Task-5-Notebook-logs

The printSchema of the parquet file

- df-prediction-printschema

The notebook that was used to create the printschema with an example of the saved predictions.

- Task-5-PrintSchema-Notebook-Pdf
- Task-5-PrintSchema-Notebook

hdfs ls of the parquet files and the text files

- hdfs ls root
- hdfs csv ls
- hdfs csv cat