Задание №4 Deadline 18.04.22 17:59

В этом задании вам предстоит предсказать оценку товара по его текстовому обзору (review). Тренировка модели и предсказание должны быть сделаны с использованием Spark ML.

Описание датасета

Датасет состоит из json-строк следующего вида:


```
{
  "overall": 1,
  "vote": "25",
  "verified": false,
  "reviewTime": "12 19, 2008",
  "reviewerID": "APV13CM0919JD",
  "asin": "B001GXRQW0",
  "reviewerName": "LEH",
  "reviewText": "Amazon,\nI am shopping for Amazon.com gift cards for Christmas gifts and am really so disappointed that out of five choices there isn't one that says \"Merry Christmas\" or mentions Christmas at all!  I am sure I am not alone in wanting a card that reflects the actual \"holiday\" we are celebrating. On principle, I cannot send a Amazon gift card this Christmas.  What's up with all the Political Correctness?  Bad marketing decision.\nLynn",
  "summary": "Merry Christmas.",
  "unixReviewTime": 1229644800
}
```


где:

- overall - рейтинг, поставленный пользователем, в общем случае - вещественное число
- vote - скольким людям понравился обзор
- verified - True, если известно, что автор обзора купил этот товар на сайте.
- reviewTime - время написания или опубликования обзора
- reviewerID - идетнификатор автора
- asin - идентификатор товара

- reviewerName - имя автора
- reviewText - собственно текст обзора.
- summary - короткое резюме по товару, или заголовок обзора.
- unixReviewTime - время обзора в секундах эпохи.

Целевой переменной является overall, все остальные поля - признаки.

Местонахождение датасета

- /datasets/amazon/all_reviews_5_core_train.json - большой тренировочный датасет примерно на 20 миллионов записей.
- /datasets/amazon/all_reviews_5_core_train_small.json - маленький тренировочный датасет на 1 миллион записей.
- /datasets/amazon/all_reviews_5_core_test_features.json - тестовый датасет на примерно 83 миллиона записей.

Заметьте, что все значения asin из большого тренировочного датасета присутствуют и в тестовом датасете. В маленьком тренировочном датасете присутствуют не все значения asin.

Оформление работы

В подпапке вашего репозитория projects/4/ сохраните следующие файлы:

- model.py - с определение пайплайна модели
- train.py - с кодом для тренировки модели
- predict.py - с кодом для инференса модели

Модель

Модель должна быть в форме пайплайна, на который подается тренировочный или тестовый датафрейм без предварительной обработки. Вся обработка должна проводится в пайплайне средствами Spark ML. В этом задании вы можете использовать только стандартные классы из библиотеки Spark.

Файл с определением модели - model.py. Переменная определения модели должна называться pipeline.

Тренировка

Скрипт train.py должен принимать следующие аргументы:

- путь к тренировочному датасету (в HDFS)

- путь для сохранения модели (в HDFS)

Импортируйте модель в ваш тренировочный скрипт следующим образом:

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()
spark.sparkContext.setLogLevel('WARN')

from model import pipeline
```

Обратите внимание, что импорт модели из файла model.py должен идти после инициализации контекста.

После того как модель обучена, она сохраняется следующим образом:

```
pipeline_model.write().overwrite().save(model_path)
```

Предсказание

Скрипт для предсказания должен принимать следующие аргументы:

- путь к сохраненной в HDFS обученной модели
- путь к тестовому датасету
- путь для сохранения предсказаний в HDFS.

В скрипте для предсказания вам уже не нужно импортировать модель из model.py, но надо загрузить ее с HDFS.

```
from pyspark.sql import SparkSession

spark = SparkSession.builder.getOrCreate()
spark.sparkContext.setLogLevel('WARN')

from pyspark.ml import Pipeline, PipelineModel

model = PipelineModel.load(model_path)
```

Проверка

Запустите чекер: checker.sh 4

Чекер будет запускать ваш скрипт для тренировки примерно следующим образом:

```
PYSPARK_PYTHON=/opt/conda/envs/dsenv/bin/python3 spark-submit \
    --master yarn \
    --name checker-train \
    --py-files projects/4/model.py \
    projects/4/train.py $TRAIN_PATH $MODEL_PATH
```

обратите внимание на опцию --py-files - она копирует файл с определением модели на кластер, где он доступен для скрипта train.py

Затем чекер проверит, что модель сохранена в HDFS и запустит инференс на тестовом датасете:

```
PYSPARK_PYTHON=/opt/conda/envs/dsenv/bin/python3 spark-submit \
    --master yarn \
    --name checker-test \
    projects/4/predict.py $MODEL_PATH $TEST_PATH $PRED_PATH
```

Особенности проверки

Внимание! В этом задании мы запускаем нашу версию тестового скрипта и скорера.

Зачет

Зачет производится при RMSE<1.5 на тестовой выборке.

Подсказки по работе

Начните работать в Jupyter-ноутбуке, попробуйте разные классы трансформеров и эстиматоров с разными значениями параметров. Это поможет понять, какие модели работают, то есть дают больший скор. Затем воспользуйтесь подбором гиперпараметров на гриде для улучшения скора базовой модели.

Внимание - в вашем файле train.py не оставляйте код для подбора параметров на гриде! В этом скрипте модель должна обучаться один раз с уже подобранными ранее гиперпараметрами.

Однако приложите файл с кодом тренировки с подбором тоже (это может быть Юпитер ноутбук, или .py файл).

Сообщения об окончании проверки

Чекер должен выдать PASSED 1 для зачета.

Чекер может выслать вам сообщение об окончании проверки. Для этого создайте в корне вашего репозитория файл tguserid, в который запишите id своего пользователя в Telegram.

Чтобы узнать свой ID, пошлите сообщение /id нашему боту @ozonm_big_data_bot.