CSCI316 – Big Data Mining Techniques and Implementation
Assignment 2

10 Marks
Deadline: Refer to the Assignment2 submission link on Moodle

Two (2) tasks are included in this assignment. The specification of each task starts in a separate page.

You must implement and run all your Python code in Jupyter Notebook. ***The deliverables include one Jupyter Notebook source file (with .ipybn extension) and one PDF document for each task.***

Note: To generate a PDF file for a notebook source file, you can either (i) use the Web browser's PDF printing function, or (ii) click "File" on top of the notebook, choose "Download as" and then "PDF via LaTex".

All results of your implementation must be reproducible from your submitted Jupyter notebook source files. In addition, the submission must include all execution outputs as well as clear explanation of your implementation algorithms (e.g., in the Markdown format or as comments in your Python codes).

Submission must be done online by using the submission link associated with assignment 1 for this subject on MOODLE. The size limit for all submitted materials is 20MB. DO NOT submit a zip file.

Submissions made after the due time will be assessed as late submissions. Late submissions are counted in full day increments (i.e. 1 minute late counts as a 1 day late submission). There is a 25% penalty for each day after the due date including weekends. The submission site closes four days after the due date. No submission will be accepted after the submission site has closed.

This is an individual assignment. Plagiarism of any part of the assignment will result in having 0 mark for the assignment and for all students involved.

Marking guidelines

Code: Your Python code will be assessed. The computers in the lab define the standard environment for code development and code execution. Note that the correctness, completeness, efficiency, and results of your executed code will be assessed. Thus, code that produces no useful outputs will receive zero marks. This also means that code that does not run on a computer in the lab would be awarded zero marks or code where none of the core functions produce correct results would be awarded zero marks.

Presentation and explanation: The correctness, completeness and clearness of your answers will be assessed.

Task1

(8 marks)

Dataset: Yoochoose Clicks Dataset (yoochoose-clicks.dat)
Source: http://recsys.yoochoose.net/challenge.html

Dataset Information

Download the data source from the above link. The size of the files after decompression is about 1.91GB. The data source contains three datasets. For this task, you just need the yoochoose-clicks.dat dataset.
This dataset is a set of click events collected from a website of an online retailer. Each record in the dataset has four (4) fields:
Session ID - the id of the session. In one session there are one or many clicks.
Timestamp - the time when the click occurred.
Item ID – the unique identifier of item.
Category – the category of the item.

The value "S" indicates a special offer, "0" indicates a missing value, a number between 1 to 12 indicates a real category identifier, and any other number indicates a brand. If an item has been clicked in the context of a promotion or special offer then the value is "S". If the category is a brand (i.e., BOSCH) then the value is an 8-10 digits number. If the item has been clicked under a regular category (e.g., sport) then the value is a number between 1 to 12.

The Task

You are to perform ***exploratory analysis*** on the given dataset of click events by using Spark's DataFrame API. The objective is to compute *the average time that users stay on items in each category*.

For analysis purposes in this task, use the following definitions:

(i) There are 15 item categories in the dataset: S, 0, 1 to 12, and B (for any 8-10 digits number)
(ii) In each session, the time that a user stays on some item is the timestamp difference between a user clicking on this item and the next item (if there is a next item).

Requirements

(i) Load yoochoose-clicks.dat into a Spark DataFrame with correct types and *print its schema*.
(ii) Implement a sequence of DataFrame transformations plus one action that produces the final output (as specified above). ***Do not*** convert any DataFrame into Python data structure such as Pandas dataframe, NumPy array, list, collection, etc. The entire analysis should be completed with the Spark DataFrame API.

Deliverables

(1) A Jupiter Notebook source file named `<your_name>_task1.ipybn` which contains your implementation source code in Python
(2) A PDF document named `<your_name>_task1.pdf` which is generated from your Jupiter Notebook source file, and presents clear and accurate explanation of your implementation and results. A poor presentation of results gains less marks in this task.

Task 2

(7 marks)

Dataset: YearPredictionMSD Data Set
https://archive.ics.uci.edu/ml/datasets/YearPredictionMSD

Dataset information
This dataset is a subset of the Million Song Dataset: (http://labrosa.ee.columbia.edu/millionsong/). The dataset has been pre-processed. In particular, the numerical audio features are extracted (by using. the Echo Nest API). The first attribute is the release year. Other attributes include two group: the timbre average (12 columns) and the timbre covariance (78 columns).

Objective
Implementation of data mining project in Spark ML.

Task requirements

(1) This is a regression problem. The task is to predict the release year of a song from audio features. Use the first 463,715 examples as the training dataset and the last 51,630 examples as the test dataset.

(2) Main steps of the project are (a) "discover and visualise the data", (b) "prepare the data for machine learning algorithms", (c) "select and train models", (d) "fine-tune the models" and (e) "evaluate the outcomes". You can structure the project in your own way. Some steps can be performed more than once.

(3) In the steps (c) and (d), you must work with at least three models.

(4) Explanation of each step together with the Python codes must be included.

(5) Only Spark MLlib can be used. Other ML libraries are not allowed in this task.

Deliverables

(1) A Jupiter Notebook source file named `<your_name>_task2.ipybn` which contains your implementation source code in Python

(2) A PDF document named `<your_name>_task2.pdf` which is generated from your Jupiter Notebook source file, and presents clear and accurate explanation of your implementation and results. A poor presentation of results gains less marks in this task.