BigData Diploma Project Task
Use PySpark!

Aim

Hi there! As a result of building current project you will be able to:

- Work with one of the most popular and trending frameworks Apache Spark. [1-6]
- Compose different queries to distributed data using SQL. [7-10]
- Get practical knowledge of the world-top VCS (Version Control System) Git and Repository Manager GitLab. [11-13]
- *Understand how to write unit tests for your code. [14-17]
- Look at the open source data [18-19]
- Accomplish the documentation of your code [20-22].

Overview

As BigData Developers you are expected to build your own ETL pipelines using Apache Spark with IMDB database.

You might use any of Spark-friendly languages, but our personal recommendation is to start with **PySpark**. The main reasons are high-level API and easy-to-use installation.

E - extract. You have to get the data from the open source IMDB dataset.

T - transform. Apply transformations (operations on data) in tasks described below.

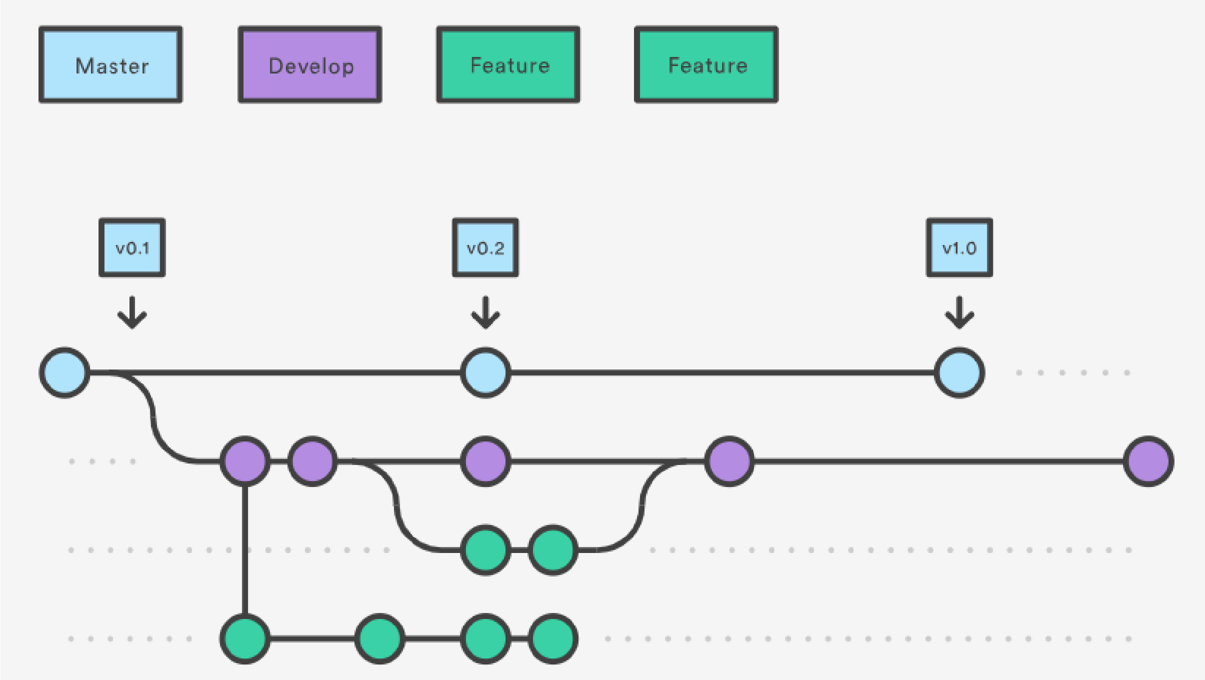
L - load. Load results to .csv files as described below.

Therefore, you will produce prepared and useful data for users of some abstract app. Or Backed Developers will be able to get this data and deliver it to user of the app.

Tasks

Preparation stage

1. Choose an appropriate programming language for Spark. Read the docs and useful resources. Pay careful attention to SQL modules.
2. Install Spark.
3. Create a Git repository on GitLab or Github and make sure it's public or accessible by mentor's email. Please, use meaningful names for your repository (e.g. imdb-spark-project).
4. Create 2 branches: main (old master) and develop (from main). It is made to enable teamwork and double-checking. In the future you will create your branches for each small piece of functionality from develop and create merge requests to develop (review process with your teammate). It will look like this:



5. Create a project locally. Please, use a meaningful name as below.
6. Add file .gitignore to your project. [23]
7. Download IMDB datasets to your PC. Do not put them into the project repository.
8. Look at the datasets. Explain, what data of what types do we have? It will be useful for further development.

Setup Stage

1. Depending on the language you chose, follow appropriate instructions to set up Spark. [24]
2. Try to create a test DataFrame and apply .show() method on it. If your data is shown as expected, congrats! We can move forward! If not, just manage the problem. Try to resolve the issue by yourself, but don't spend more than a couple of hours on it. In any case, if you and your teammate are stuck, text your tutor and ask for help.

Extraction Stage

Here is the place where all the magic starts.

1. Create appropriate schemas for all 7 datasets.
2. Depending on the schemas above create corresponding DataFrames by reading data from the PC.
3. Check if everything went right with any method on DataFrames.
4. *Wrap it up in a function and write unit tests for this function.

Transformation Stage

! *Prepare unit tests and documentation for all your functionality.

Hint: write your transformations as functions. It will be much easier to test particular functionality.

1. Get all titles of series/movies etc. that are available in Ukrainian.
2. Get the list of people's names, who were born in the 19th century.
3. Get titles of all **movies** that last more than 2 hours.
4. Get names of people, corresponding movies/series and characters they played in those films.
5. Get information about how many adult movies/series etc. there are per region. Get the top 100 of them from the region with the biggest count to the region with the smallest one.
6. Get information about how many episodes in each TV Series. Get the top 50 of them starting from the TV Series with the biggest quantity of episodes.

7. Get 10 titles of the most popular movies/series etc. by each decade.
8. Get 10 titles of the most popular movies/series etc. by each genre.

Loading Stage

1. Load all results of transformations to .csv files. Pay attention: you have to get **ONE** .csv file per **ONE** transformation (8 files overall). Do **not** push them into the repository (add your data folder to .gitignore).

**Extra task. Data Modelling.*

Use pyspark mllib to build a linear regression model to predict the ratings of the movies.

Useful Resources

1. Apache Spark: https://spark.apache.org/
2. Download Spark: https://spark.apache.org/downloads.html
3. Java API: https://spark.apache.org/docs/latest/api/java/index.html?org/apache/spark/sql/Dataset.html
4. Python API: https://spark.apache.org/docs/latest/api/python/
5. Scala API: https://spark.apache.org/docs/latest/api/scala/org/apache/spark/sql/Dataset.html
6. Book Learning Spark (2nd edition): https://www.oreilly.com/library/view/learning-spark/9781449359034/ OR write an email to abondarchuk@griddynamics.com to get PDF.
7. Tutorial https://www.w3schools.com/sql/
8. PySpark SQL module: https://spark.apache.org/docs/latest/api/python/pyspark.sql.html#pyspark-sql-module
9. Apache Spark SQL Getting Started: https://spark.apache.org/docs/latest/sql-getting-started.html
10. SQL Syntax: https://spark.apache.org/docs/latest/sql-ref-syntax.html
11. Git documentation: https://www.git-scm.com/doc
12. Git interactive tasks: https://learngitbranching.js.org/

13. Learn Git and get certificate: https://www.codecademy.com/learn/learn-git
14. JUnit intro: https://www.vogella.com/tutorials/JUnit/article.html
15. JUnit documentation: https://junit.org/junit4/
16. Python unittest documentation: https://docs.python.org/3/library/unittest.html
17. Pytest documentation: https://docs.pytest.org/en/stable/
18. IMDB Datasets' Description: https://www.imdb.com/interfaces/
19. Data Location: https://datasets.imdbws.com/
20. Javadoc: https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html
21. Python docstring formats: https://stackoverflow.com/questions/3898572/what-is-the-standard-python-docstring-format
22. Google documentation style for Python: https://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html
23. Resource to generate .gitignore file online: https://www.toptal.com/developers/gitignore
24. Set Spark up in different languages: https://intellipaat.com/blog/tutorial/spark-tutorial/downloading-spark-and-getting-started/