In the Name of God

Amirkabir University of Technology (Tehran Polytechnic)

Course: Big Data Analytics

Human Analysis, Machine Learning Classification and Clustering of Big Data, and Mathematical Operations using MapReduce

Instructor: Dr. Pourfard

TA Manager: Dr. Talebi

Assignment Author (TA): Nima Mousavi Monfared

Assignment No. 1

Fall 2025

Knowledge Objectives:

- Data inspection and classification using Big Data processing techniques
- Working with unlabeled data and performing clustering with Big Data processing methods
- Word counting and calculating average temperature using the MapReduce approach
- Performing matrix addition and multiplication using the MapReduce approach

Skills to be Acquired:

- Enhancing Python programming skills
- Working with PySpark (the Python API of Apache Spark)
- Using the MRJob Python library for implementing MapReduce
- Implementing Classification (Naive Bayes) using PySpark
- Implementing Clustering (K-Means++) using PySpark
- Implementing MapReduce-based algorithms using PySpark and MRJob

Important Notes Before You Begin:

- The goal of these exercises is to improve your knowledge and practical skills.
- You may seek help and guidance from classmates or AI tools, but you are fully responsible for your code, results, and analysis.
- Make sure you understand the concepts and reflect your understanding in both code and report.
- Although the questions may appear long, most of their length comes from step-by-step guidance and clarifications; the actual coding workload is moderate.
- Follow the provided instructions carefully—they are designed to simplify the tasks and remove potential ambiguities.
- Focus first on completing the requested tasks correctly. If you have extra time, improve or extend your work afterward.
- The core parts of these exercises must be executed within the Big Data frameworks (Spark, MRJob). However, for visualization and side computations (e.g., charts, printing results), you may use NumPy, Pandas, Matplotlib, etc.
- Please read the Appendix before starting.

Keywords:

Big Data Analytics, PySpark, MRJob, MapReduce, Classification, Clustering, K-Means++, AI-Generated Texts, Matrix Addition and Multiplication, Average Temperature

Question 1

With the increasing use of AI-based text generation tools and the growing presence of fake or synthetic data, distinguishing between AI-generated and human-written texts has become a major challenge in fields such as education, journalism, and content creation.

In this exercise, you will perform a classification task on such data to help develop models capable of identifying AI-generated content — which is vital for preventing the spread of misinformation or unreliable materials.

You are provided with a dataset containing around 500,000 textual articles, including both human-written and AI-generated texts:

Dataset:

https://www.kaggle.com/datasets/shanegerami/ai-vs-human-text

Tasks:

A. Set up Google Colab for working with Spark (see Appendix).
B. Load the dataset as a Spark DataFrame, and identify the contents of the target column.
C. Preprocess the data using Spark:
   i. Convert all words to lowercase and remove all non-alphanumeric characters.
   ii. Tokenize each text (split into words).
   iii. Remove stop words
   (Help:
      - Spark's built-in remover: StopWordsRemover
      - You can use, NLTK stopword lists: How to use NLTK stopwords)
   iv. Explode each token so that each word appears on a separate row. (Exmple)
D. Display each word along with its frequency count. (Showing the top 20 words is sufficient.)
E. Display the words with the highest frequencies, then display the words with the lowest frequencies. (Showing 10 words from each group is sufficient.)
F. Display the unique (non-repeated) words. (Showing 20 examples is sufficient.)
   Report the total number of unique words.
G. Now, for each class separately, find the most frequent words and display the top 10 for each. Is there a noticeable difference between the most frequent words used by humans and those used by AI? In your opinion, could this difference help in detecting AI-generated content or plagiarism?
H. Re-run the previous step (finding the most frequent words per class) while varying the number of CPU cores, up to at least 8 cores. Report the execution time for each configuration and plot the execution time versus number of cores. Discuss whether the observed times and core numbers have any specific reason or pattern.

Question 2

Now that we have examined vocabulary differences between human and AI texts, we move toward a machine learning–based approach to automate the classification.

Using the same dataset from Question 1.

Tasks:

A. Preprocessing
   - Load and preprocess data as before (lowercasing, cleaning, tokenizing, stopword removal).
   - Vectorize the text for use as model input. (Suggested vectorizer: CountVectorizer)
   - Split the dataset into training (80%) and testing (20%) subsets using a fixed seed(seed=123).

B. Research and briefly explain Bayesian Classifier and Naive Bayes algorithms. Which is more suitable for big data analytics? Why?

C. Implement a Naive Bayes model using PySpark's ML library, train it, and evaluate it.

D. Report the classification metrics: Accuracy, Precision, Recall, and F1-score. (Evaluation ref)

Question 3

Clustering is a key unsupervised learning method, especially useful in Big Data analytics. Here, you will perform clustering using PySpark.

Two 2D coordinate datasets are provided: *dataset_clustring_1.csv* and *dataset_clustring_2.csv*, each containing over 1 million samples and their corresponding ground truth labels.

Perform the following steps for each dataset separately:

A. Plot the data points, colored by their true classes.
    - Tip: Due to high sample size, set the scatter parameter 's' to a small value like 1. (Matplotlib)

B. Briefly explain K-Means and K-Means++ algorithms.

C. Describe how K-Means++ can be applied in PySpark. Cluster the points using K-Means++, testing cluster numbers(k) from 2 to 25. Tips:
    - KMeans
    - Use VectorAssembler to combine features into a single vector column.

D. Plot cost vs number of clusters, and use the *Silhouette* score to evaluate performance (ClusteringEvaluator).

E. Determine the optimal number of clusters using an algorithm such as *KneeLocator* (GitHub).

F. Identify cluster centers and label each data point by cluster.

G. Plot the clustered data (colored by cluster numbers).

H. Compare results with true class plots. How many true classes and how many detected clusters are there? Was K-Means++ successful for both datasets? Briefly explain why or why not.

Question 4

One of the tools that simplifies MapReduce operations is the Python library MRJob, which abstracts away cluster management and simplifies the MapReduce process.

A. Using MRJob, write a program that counts word occurrences in text using the MapReduce paradigm.
   - **Dataset**: *harrypotter_processed.txt* (preprocessed Harry Potter stories).
   - Include the main code parts in your report and explain, theoretically, what each part does.

B. Using MRJob and MapReduce, compute the average monthly temperature for Tehran based on daily data in *Tehran_temperature_data.txt*. (GitHub example)

Using Spark and the MapReduce approach, implement:

C. Matrix Multiplication, based on this repository. (GitHub example)
Compute:

$$\begin{bmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{bmatrix}\begin{bmatrix}7 & 8\\ 9 & 10\\ 11 & 12\end{bmatrix} \quad , \quad \begin{bmatrix}10 & 11\\ 12 & 13\end{bmatrix}\begin{bmatrix}14 & 15\\ 16 & 17\end{bmatrix}$$

D. Matrix Addition using MapReduce:

$$\begin{bmatrix}0 & 0 & 0\\ 1 & 1 & 1\\ 2 & 2 & 2\end{bmatrix}+\begin{bmatrix}3 & 3 & 3\\ 4 & 4 & 4\\ 5 & 5 & 5\end{bmatrix} \quad , \quad \begin{bmatrix}10 & 11\\ 12 & 13\end{bmatrix}+\begin{bmatrix}14 & 15\\ 16 & 17\end{bmatrix}$$

For each part(C, D):

- Include and explain main code sections and their theoretical purpose.
- Ensure your implementation can handle matrices of arbitrary size.

Appendix

Sample code for:

- Connecting Google Drive to Colab

```
from google.colab import drive
drive.mount('/content/drive')
```

- Setting up PySpark in Google Colab

```
#  java installion
!apt-get install openjdk-17-jdk-headless -qq > /dev/null

# install spark (note! : check version number)
!wget -q https://archive.apache.org/dist/spark/spark-3.5.0/spark-3.5.0-bin-hadoop3.tgz
# unzip the spark file to the current folder
!tar xf spark-3.5.0-bin-hadoop3.tgz

# path of spark folder
import os
os.environ["JAVA_HOME"] = "/usr/lib/jvm/java-17-openjdk-amd64"
os.environ["SPARK_HOME"] = "/content/spark-3.5.0-bin-hadoop3"
os.environ["PATH"] = os.environ["SPARK_HOME"] + "/bin:" + os.environ["PATH"]
```

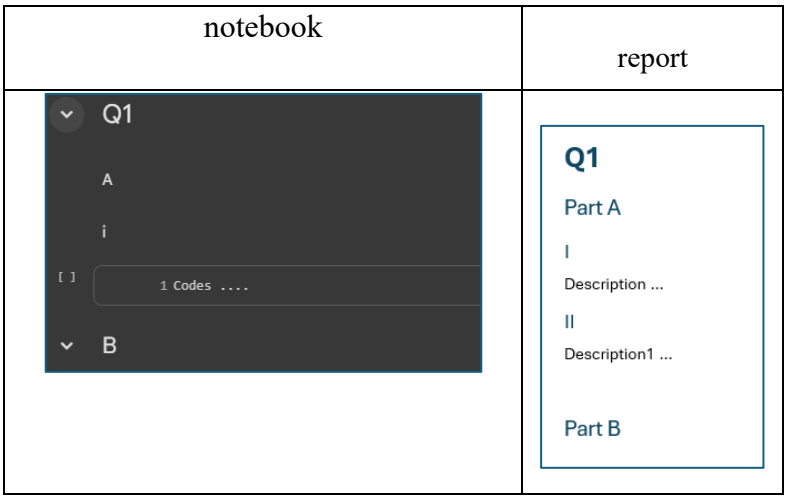
Deliverables

1. Source code files for each question
2. A comprehensive report, including:
    - Problem descriptions
    - Designed algorithms and ideas
    - Explanation of implemented programs
    - Required outputs, results, and analysis

Submit all materials in a single *.zip* file on the uni Courses site.

Important Notes

- Write code in a .ipynb notebook (preferably in Google Colab).
- Comment your code thoroughly; keep it clean and readable.
- Include a Table of Contents in the report and follow the same structure as the questions.
- You do not need to explain code line-by-line — a clear overview of each block is sufficient.
- Always include the requested outputs or screenshots of logs and results for each part.
- To receive full credit, your explanations must be clear, concise, and focused. Avoid unnecessary verbosity.
- Please follow the writing format shown below:



Wishing you success in your work.