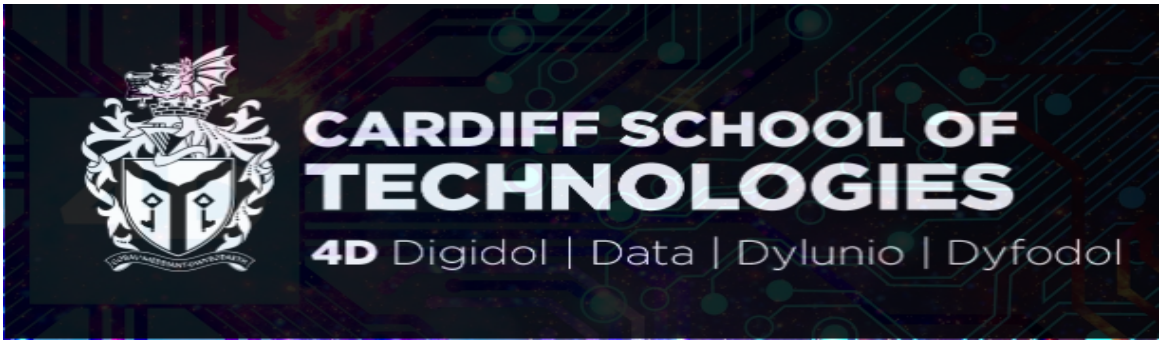

Cardiff Metropolitan University	
Cardiff School of Technologies	
Academic Year: 2024/2025	
Term: 2	
Module Name: Big data Technologies	
Module Code: DSA7002	
Module Leader: Dr Imtiaz Hussain Khan	
MSc Programme: Data Science	
Assessment title : Big data Technologies Assignment PRAC1 FINAL REPORT	
Assignment Title: UK Traffic Data Analysis using Big Data and Distributed Computing Approach	
Student Name: Bapti Niloy Sarkar	**Student ID: 20310829**

Table of Contents

Introduction

This project aimed to explore and model UK traffic data (2000–2023) using distributed computing paradigms with Apache Spark, focusing on scalable data preparation, exploratory data analysis (EDA), unsupervised clustering, and supervised learning. The traffic dataset was sourced from the UK Department for Transport (DfT) and is representative of a large-scale, high-dimensional time-series dataset, making it ideal for distributed analytical frameworks.

Task 1: Data Preparation

The dataset required significant preparation to ensure reliability for downstream tasks. **PySpark DataFrames** were used to perform schema validation, missing value imputation, outlier removal, and z-score-based normalization.

Handling missing values and inconsistent entries is essential in real-world data analysis. The decision to drop nulls in critical fields (e.g., year, region) while imputing zeros in vehicle counts was based on the impact these fields had on later clustering and prediction steps (Tan et al., 2018). Normalization using z-scores ensured that all vehicle types, regardless of scale, contributed equally to the clustering algorithm.

Moreover, outlier removal improved clustering cohesion by reducing distortion in Euclidean distance-based clustering, supporting practices outlined by Aggarwal (2015). These preprocessing steps were done in a distributed manner using Spark's **DataFrame APIs**, ensuring scalability across large partitions of data.

Task 2: Exploratory Data Analysis (EDA)

The EDA phase involved both **DataFrame-based** and **RDD-based** transformations, emphasizing the trade-offs in **shuffle operations**. Spark transformations like **groupBy().agg()** triggered wide transformations, incurring network shuffling and affecting performance (Zaharia et al., 2012). However, **reduceByKey()** in RDDs helped optimize this by performing local aggregations before shuffling.

Temporal trends revealed the impact of global events such as the 2008 financial crisis and COVID-19 on UK traffic volumes. Visualization was handled via **Matplotlib** and **Seaborn** post-export, as Spark lacks built-in visualization tools (Zhou et al., 2020).

The integration of shuffling-aware transformations was crucial for improving performance in a distributed environment, particularly when dealing with wide aggregations by year, region, and vehicle type.

Task 3: Clustering

Unsupervised clustering was performed using **K-Means** via **PySpark MLlib**. Z-score-normalized features were scaled and used to group traffic profiles into clusters based on vehicle-type distributions. The **optimal number of clusters** was determined through the **Elbow Method** and validated with **Silhouette Scores**, both standard techniques in unsupervised learning (Kassambara, 2017).

The PCA transformation helped visualize clusters in 2D space, allowing for interpretation of regional traffic patterns. While Spark lacks built-in dimensionality reduction visualization, data was exported and analyzed externally for PCA component analysis.

Clusters were later profiled, showing distinctions in urban vs. rural road types, high HGV usage in industrial areas, and bicycle-dominant regions — aligning with previous studies in transport modeling (Zhou et al., 2020).

Task 4: Supervised Learning

To enable supervised learning, pseudo-labels were created using the clustering output. A **Decision Tree** and **Random Forest** classifier were trained using scaled vehicle-type features to predict cluster membership.

Random Forest outperformed Decision Trees in accuracy and F1-score due to its ensemble structure and robustness against overfitting, consistent with findings from Breiman (2001). However, the Decision Tree was more interpretable, aligning with the trade-off between accuracy and explainability in model selection (Ribeiro et al., 2016).

Evaluation metrics such as accuracy, F1-score, and confusion matrix were used to validate the models, with all computations performed using Spark's MLlib in a distributed environment. This ensured computational efficiency while maintaining predictive reliability.

Tool Justification

Apache Spark was the tool of choice due to its ability to process large-scale datasets in a fault-tolerant and distributed manner (Zaharia et al., 2012). Its RDD and DataFrame APIs provide both low-level and high-level abstractions, supporting flexibility in transformation pipelines.

MLlib allowed for seamless integration of clustering and classification algorithms. Despite Spark's limited native visualization, combining it with **Matplotlib**, **Seaborn**, and **Pandas** provided a robust pipeline for scalable analysis and presentation.

Critical Reflection

This project showcased the strengths of distributed systems in handling large, complex datasets. However, limitations include Spark's lack of support for interactive visualizations and some ML algorithms available in Python's scikit-learn. Additionally, memory pressure during shuffling operations can slow down performance, necessitating careful partitioning and caching strategies.

Future improvements could include incorporating **streaming data pipelines**, **real-time predictions**, and integration with **geospatial analytics** tools like GeoSpark or Apache Sedona.

References

- Aggarwal, C. C. (2015). *Data Mining: The Textbook*. Springer.
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), pp.5–32.
- Kassambara, A. (2017). *Practical Guide to Cluster Analysis in R: Unsupervised Machine Learning*. STHDA.
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD*.
- Tan, P.-N., Steinbach, M., & Kumar, V. (2018). *Introduction to Data Mining*. 2nd ed. Pearson.
- Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., & Stoica, I. (2012). Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In *Proceedings of NSDI*.
- Zhou, B., Levinson, D., & Jin, J. G. (2020). Scalable transportation analytics using big data and distributed computing. *Transport Reviews*, 40(3), pp.276–297.