Climate Impact on Urban Mobility: Analyzing Bike-Sharing Demand

Apu Das - 23186407

Urban bike-sharing systems have emerged as a popular and eco-friendly mode of transportation in many cities, playing a significant role in mitigating climate change by lowering CO2 emissions. Understanding the factors that impact bike share rentals is crucial for optimizing efficiency and ensuring rider demand in these systems. This study analyzes the factors influencing bike rentals using two datasets from two countries. By exploring the impact of climate change (temperature, humidity, and wind speed), we will try to find trends and patterns in bike rental numbers. Thus, this analysis will provide valuable insights for optimizing eco-friendly mobility in the city. It will also ensure sufficient bike availability and contribute to the fight against climate change by promoting sustainable transportation alternatives.

I. Question

How does climate change (temperature, humidity, and wind speed) affect bike rentals?

II. Data Sources

For this project, I have chosen two datasets that offer comprehensive data on bike share rentals: Bike Sharing by Capital bikeshare system [1] and Seoul Bike Sharing Demand by Seoul Bike Sharing System [2]. The datasets are from the UCI Machine Learning Repository, a well-known repository for machine learning datasets. The primary reasons for selecting these datasets include their rich detail on rental activities, especially weather conditions (temperature, humidity, and wind speed), which is crucial for analyzing factors influencing bike rentals.

A. Data Structure

The Capital bikeshare dataset is structured with temporal variables (date, season, year, month, hour), categorical variables (holiday, weekday, working day, weather situation), and continuous variables (temperature, feeling temperature, humidity, wind speed). Notably, the dataset doesn't contain any missing values, ensuring a high quality of data for analysis.

The Seoul Bike Sharing dataset is structured with temporal, categorical, and continuous features as well. It includes temporal variables (Date and Hour), categorical variables (Seasons, Holiday, Functioning Day), and continuous variables (Rented Bike Count, Temperature, Humidity, Wind Speed, Visibility, Dew Point Temperature, Solar Radiation, Rainfall, Snowfall). It doesn't contain any missing values also.

B. Data Quality

The **Capital Bikeshare system dataset** has several key dimensions of data quality. `Accuracy` is ensured as the data accurately reflects real-world bike-sharing activities, with features such as temperature, humidity, season, user counts, and other real-world information. In terms of `timeliness`, the data covers two full years for understanding patterns and trends in bike-sharing usage. However, given that the data is from 2011-12, the dataset may not accurately reflect current conditions and demand for bike sharing. `Relevancy` is achieved by focusing on the key factors that influence bike rentals, such as weather conditions, holidays, and working days.

Histograms and Kernel Density Estimation (KDE) plots were used to visually analyze the distributions and identify anomalies within the Capital bikeshare dataset. The histograms indicate that the data is balanced across season, year, month, hour, and weekday, while holiday, workingday, and weathersit accurately reflect real-world ratios.

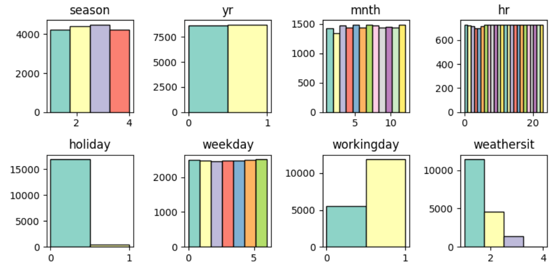

Fig. 1. Categorical features from the Capital Bikeshare dataset

Similarly, the KDE plots for continuous features suggest that the data for temperature, feeling temperature, and humidity approximates a normal distribution (bell-shaped curves).

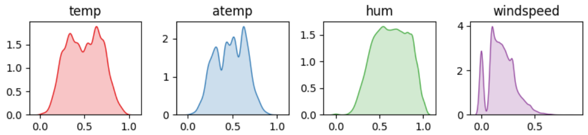

Fig. 2. Continuous Features from the Capital Bikeshare dataset.

The **Seoul Bike Sharing dataset** also shows robust data quality. It accurately reflects real-world bike-sharing activities, including temperature, humidity, season, and user counts. Regarding `timeliness`, the dataset spans multiple years, ensuring relevance for understanding bike-sharing trends and patterns (2017-18). The dataset also remains relevant for bike-sharing analysis as it focuses on essential factors in weather conditions.

I also used Histograms and Kernel Density Estimation (KDE) plots to analyze the distributions of the Seoul Bike Sharing dataset. The histograms indicate that the data is balanced across 'Seasons' and 'Hour.' Also, 'Holiday' and 'Functioning Day' accurately reflect real-world ratios.

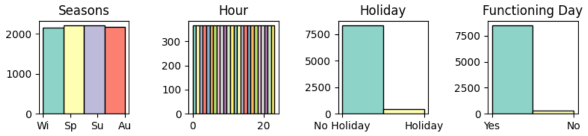

Fig. 3. Categorical features from the Seoul Bike Sharing dataset.

Similarly, the KDE plots for continuous features indicate that the data for 'Temperature (°C),' 'Humidity (%),' and 'Dew point temperature (°C)' approximates a normal distribution, displaying bell-shaped curves. Furthermore, the features 'Wind speed (m/s),' 'Solar Radiation (MJ/m2),' 'Rainfall (mm),' and 'Snowfall (cm)' are right-skewed, while 'Visibility (10m)' is left-skewed.

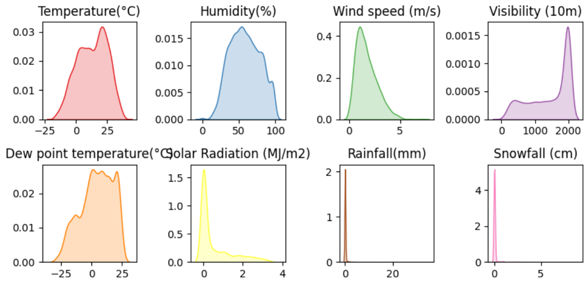

Fig. 4. Continuous Features from the Seoul Bike Sharing dataset.

C. License

Both datasets are licensed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license [3]. This license permits sharing and adapting the dataset for any purpose provided that appropriate credit is given. To comply with the licensing terms, I will ensure proper attribution is given to the UCI Machine Learning Repository and the creators of the datasets in all uses and publications. This will include citing the source in any reports, presentations, or publications resulting from this analysis. Additionally, any shared or adapted versions of the datasets will include the same CC BY 4.0 license, maintaining transparency and allowing further use by other researchers.

III. Data Pipeline

In this project, I used Python to build an automated data pipeline that pulls the dataset from the internet, transforms and cleans it, and saves it in the /data directory. The Data Pipeline Architecture I used is ETL (Extraction, Transformation, and Loading). The process begins with pulling raw datasets from online (Extraction). Then, the data goes through a series of transformation steps to clean, impute missing values, and reformat the data (as required) to ensure usability (Transformation). Finally, the transformed data is saved into a structured database, enabling easy retrieval and further analysis (Loading).

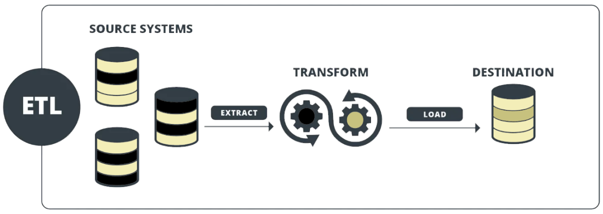

Fig. 5. The ETL data pipeline architecture [4].

A. Data Extraction: The extraction process begins with downloading the two datasets as zip files using the `requests` library. After downloading, I opened the zip archive using the `ZipFile` module and extracted the relevant CSV file. Then read the file using `pandas` read_csv function.

B. Data Transformation and Cleaning: After extraction, the data goes through several transformation and cleaning steps. Unnecessary columns, such as the initial index column from the Capital bikeshare dataset, are removed. Rows with significant missing values (at least 3 missing values) are dropped to maintain reliability. For other missing values, backward fill (bfill) imputation is applied to replace NaNs with the next valid observation. Replacing the missing values with backward fill will ensure data consistency and continuity and maintain temporal relationships between consecutive data points. For all these transformations, I used the `pandas` module.

C. Loading Data into the Sink: The final step involves saving the transformed and cleaned data into an SQLite database. The `SQLAlchemy` module is used to create a connection to the database and save the datasets into two specified tables (”Capital Bikeshare” and ”Seoul Bikeshare”). This process ensures that the data is stored efficiently and is readily available for analysis in future tasks.

During the development of the pipeline, one of the practical issues was managing the temporary storage of downloaded files on the local machine. To solve this, I automated the pipeline to automatically delete the files after successfully processing and saving them to the SQLite database.

Unfortunately, the pipeline can’t adapt to changing data. For example, continuous features (temp, atemp, hum, and windspeed) in the Capital Bikeshare system dataset are Min-Max normalized. So, new samples can’t be normalized as minimum and maximum values are unknown.

IV. Result and Limitations

A. Output Data of the pipeline: The output of the data pipeline is an SQLite database that consists of two distinct tables (”Capital Bikeshare” and ”Seoul Bikeshare”) with cleaned and transformed data from the datasets. The dataset is complete with no significant missing values. Both the tables are structured with temporal, categorical, and continuous variables, ensuring maximum coverage of relevant factors.

B. Why SQLite?: I chose the SQLite format as the pipeline output due to its lightweight nature and ease of integration with various programming languages such as Python and R. Another reason is that SQLite databases are portable and easy to share with collaborators.

C. Critical Reflection on Data and Potential Issues: The datasets in the pipeline are mostly accurate and complete, reflecting real-world bike-sharing and weather information with no missing values. However, normalizing new samples without the original min-max values may lead to inconsistencies. The cross-country datasets enrich the analysis, offering a broader perspective on how weather affects bike rentals globally, but introduce issues like differences in data collection methods, seasonal variations, and local climate patterns. Similarly, though the datasets provide historical insights, they are not continuous, as one dataset is from 2011-2012 (USA), and another is from 2017-18 (South Korea). Thus, it may not capture current trends, posing a challenge to timeliness and relevance in the analysis.

While some features may not appear significantly important at the moment, I kept them in the dataset as they may still provide valuable insights for the final analysis of how climate change affects bike rentals. Including these columns allows for a more comprehensive analysis, potentially uncovering subtle correlations and patterns that could be critical in understanding the broader impact of climate change on urban mobility.

V. Conclusion

In conclusion, the data pipeline effectively managed data extraction, transformation, and loading from the Capital Bikeshare and Seoul Bike Sharing datasets using Python and ETL architecture. By utilizing SQLite for data storage, I ensured efficient, portable, and easy data integration for the project. While the pipeline proved robust in handling structured datasets, it does face limitations, such as the inability to normalize new samples.

References

[1] Fanaee-T,Hadi. (2013). Bike Sharing. UCI Machine Learning Repository. https://doi.org/10.24432/C5W894.

[2] Seoul Bike Sharing Demand. (2020). UCI Machine Learning Repository. https://doi.org/10.24432/C5F62R.

[3] Creative Commons Attribution 4.0 International (CC BY 4.0): https://creativecommons.org/licenses/by/4.0/deed.en

[4] “Data Pipeline Architecture - A Deep Dive — StreamSets,” Software AG. (accessed Jun. 03, 2024).