Data Intensive Computing Project Modelling Report

Project: Eco-Friendly but is it safe? Analysis and Predicting Trends in E-vehicle Accidents

Group members:
Abhijeet Sanjiv Bonde (50624352)
Raghav Potdar (50631527)
Hui Yu (50311663)

Date: 06 March 2025

Problem Statement:

With the increasing adoption of E-scooters/E-bikes as a convenient and eco-friendly mode of transportation, concerns regarding their safety have emerged. This project aims to analyze and predict accident trends using historical motor vehicle collision data. Specifically, we will investigate the severity of E-scooter/E-bikes accidents compared to other vehicle types, identify key contributing factors, and determine high-risk locations and time periods for accidents[1].

GOAL:

The goal of this phase is to train models to predict the severity of crashes involving electric vehicles in terms of the number of injuries and fatalities. [2] Before model training, we must perform feature engineering to tailor our dataset to the models we aim to develop.

Feature Engineering

The selected features for model training include:

- **Crash Date & Time**
- **ZIP Code**
- **Vehicle Type**
- **Contributing Factor**

Since machine learning models require numerical inputs, we utilized the OneHotEncoder from the sklearn.preprocessing library to convert categorical text values into numerical representations [3].

Additionally, we decomposed the **Crash Date & Time** into finer-grained components [4]:

- Year
- Month
- Day
- Hour
- Day of the Week [5]

To enhance the dataset, we introduced new time-related binary indicators:

- **Is Rush Hour**
- **Is Weekend**
- **Is Night Time**

The **Crash Date & Time** feature provides critical insights into road conditions at the time of an accident. For example, rush hour may correlate with heavy traffic, while weekends and nighttime conditions might relate to either reduced traffic or increased accident severity due to riskier driving behavior.

Similarly, location is a crucial factor, as it can be indicative of road conditions. For instance, local neighborhoods with narrower roads may experience different accident patterns compared to wider, high-speed highways. Our dataset includes several location-based features, such as borough, longitude, latitude, and ZIP code. However, based on our prior exploratory data analysis (EDA), we determined that ZIP code offers the best balance between granularity and generalizability, making it the most suitable choice for modeling.

Vehicle type and contributing factors provide valuable insights into the severity of a crash. For example, an EV colliding with a sedan is likely to result in less severe consequences than a collision with a truck. Similarly, intoxication as a contributing factor may lead to more severe crashes compared to other causes.

Model Pipeline Setup

Rather than training each model separately, we developed a streamlined pipeline for simplicity and consistency by using the same feature set, training data, and testing data. We first split the dataset into 80% training and 20% testing. Next, we implemented a preprocessing step using OneHotEncoder [6] to transform categorical text values into numerical representations, as previously described.

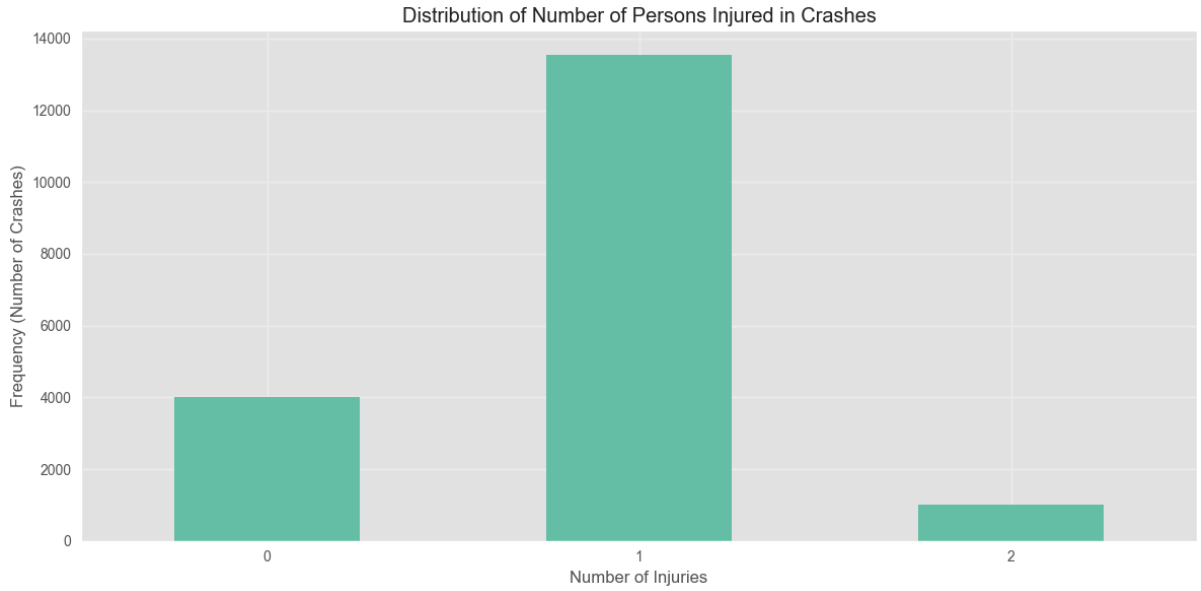
We then defined the models to be used and leveraged Optuna, an automated hyperparameter optimization framework, to fine-tune hyperparameters for each model, enhancing their overall accuracy.

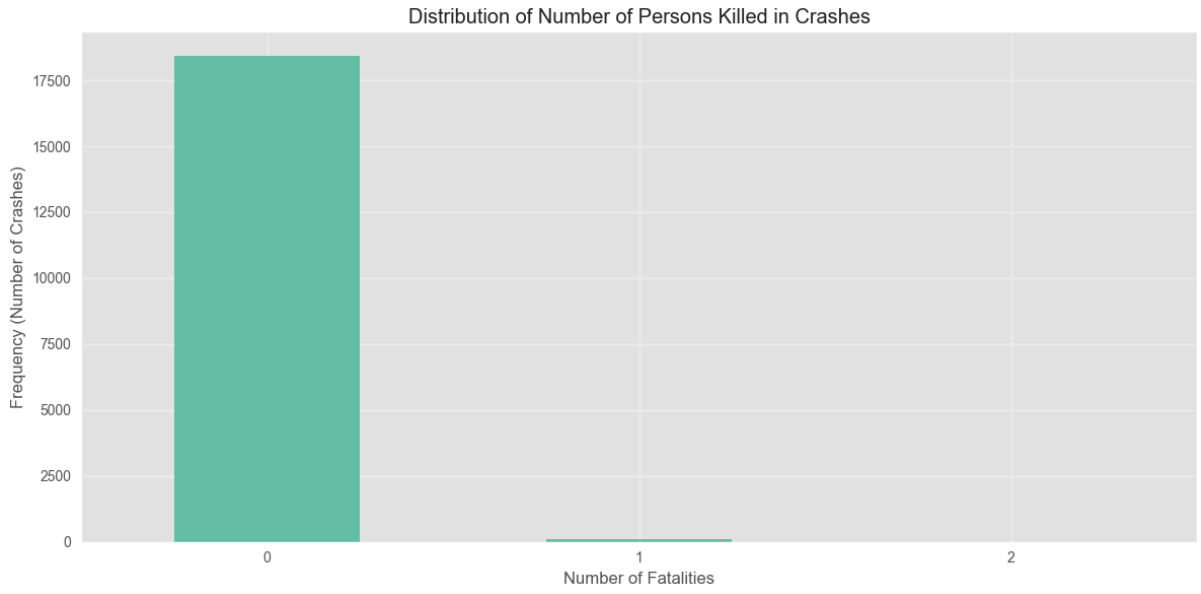
Regression Models

Initially, we attempted to use linear and polynomial regression to predict the number of injuries and fatalities, but the results were poor. The R^2 values for both models were below 0.03, and the mean squared error was approximately 0.2. We experimented with different polynomial degrees and adjusted the feature set by adding and removing variables, but these efforts did not yield significant improvements.

After extensive trial and error, we determined that a regression model is not well-suited for our dataset. Although our target variable is numerical, it is not continuous, which is a key

requirement for effective regression modeling. Furthermore, we found that the distribution of the number of injuries and fatalities is highly skewed.





As illustrated in the graph, most injury values are concentrated between 0 and 2, while the majority of fatality values are 0, with very few instances of 1 and only a single case of 2. Given this imbalance, we concluded that our problem is better framed as a classification task rather than a regression problem.

Classification Models

To classify the severity of accidents, we implemented three machine learning models:

- **K-Nearest Neighbors (KNN)**
- **Random Forest**
- **Gradient Boosting**
- **Logistic Regression**

The accident severity was categorized into three levels:

- **No Injuries (0)**
- **Minor Injuries (1)**
- **Severe Injuries (2)**

We also attempted to incorporate the number of fatalities into accident severity by categorizing datasets into those with fatalities and those without. However, due to the highly skewed nature of the dataset, where nearly all entries reported zero fatalities, the model was unable to extract meaningful patterns, resulting in predictive performance that was not only poor but entirely unusable.

KNN [7] was selected as a baseline classifier due to its simplicity and ease of interpretation. It classifies accident severity based on the majority class of the nearest neighbors, making it effective for understanding local patterns in the data. However, KNN suffers from the "curse of dimensionality [8]," where distance-based classification becomes less effective as the number of features increases. This results in reduced accuracy and makes KNN less suitable for high-dimensional datasets compared to tree-based models [9].

Random Forest was chosen for its ability to handle both categorical and numerical features while providing robustness against overfitting [10]. As an ensemble method [11], it constructs multiple decision trees on different subsets of data and aggregates their predictions, reducing variance and improving accuracy [12]. Unlike KNN, Random Forest does not suffer from the curse of dimensionality as much, since it selects only a subset of features at each decision split, making it efficient even with high-dimensional data.

Gradient Boosting, another ensemble method [13], was selected to further enhance predictive performance. Unlike Random Forest, which builds trees independently in parallel, Gradient Boosting constructs trees sequentially, with each new tree correcting the errors of the previous one [14]. This iterative learning process makes Gradient Boosting highly effective at capturing complex patterns in the data. Additionally, it is particularly useful for handling imbalanced datasets like ours, where the majority of accidents fall into the 'No Injuries' category, as it focuses more on difficult-to-classify cases. By leveraging

hyperparameter tuning with Optuna [15][16], Gradient Boosting can effectively balance bias and variance, leading to high predictive accuracy [17].

Logistic Regression:

Using the same set of features and the same 80%-20% train-test split, Logistic Regression was implemented as a binary classifier to predict whether an accident resulted in a casualty, categorizing the target variable into 'No Injuries' and 'Injuries/Death.' Unlike the other classifiers, which distinguish between three severity levels, Logistic Regression provides a broader classification by focusing on the presence or absence of injuries [19]. This model is well-suited for binary classification due to its efficiency, interpretability, and ability to estimate probabilities. It was trained using the same set of features and the same split of 80% training and 20% testing data, ensuring consistency across models [20].

Effectiveness of Model

To evaluate the effectiveness of our models in predicting accident severity, we utilized several performance metrics, including accuracy, precision, recall, and F1-score. These metrics provided insights into how well each model classified accident severity levels and whether the predictions were reliable [21].

Performance Metrics

- **Accuracy**: Measures the overall correctness of predictions across all severity levels [22].
- **Precision**: Evaluates the proportion of correctly predicted positive cases out of all predicted positive cases, which is critical for minimizing false positives.
- **Recall**: Assesses the ability to identify actual positive cases, ensuring that high-severity accidents are not overlooked.
- **F1-score**: Balances precision and recall, particularly useful in handling imbalanced datasets [23].

Insights Gained from Model Application

Regression Models Failed: As shown in our evaluation, regression models performed poorly due to the highly skewed distribution of injury and fatality counts. This confirmed that accident severity is better suited for classification.

```
Regression Models Results:

Linear Regression - Injury Prediction
Mean Squared Error: 0.2681
R-squared: -0.0200

Polynomial Regression (degree=2) - Injury Prediction
Mean Squared Error: 0.2791
R-squared: -0.0619

Polynomial Regression (degree=3) - Injury Prediction
Mean Squared Error: 0.2792
R-squared: -0.0623

Linear Regression - Death Prediction
Mean Squared Error: 0.0056
R-squared: -0.0147

Polynomial Regression (degree=2) - Death Prediction
Mean Squared Error: 0.0059
R-squared: -0.0673

Polynomial Regression (degree=3) - Death Prediction
Mean Squared Error: 0.0059
R-squared: -0.0677
```

KNN as a Baseline: While KNN provided a simple and interpretable model, it performed better than expected despite the curse of dimensionality. Although high-dimensional feature spaces can reduce the effectiveness of distance-based classification, KNN was still able to capture meaningful local patterns in the data, making it a competitive baseline model as shown in the final graph comparing the F1 score of all three models.

```
[*] Classification Models Results [Injuries]
[*] KNN_Injury Results:
[*] Accuracy: 0.7301
[*] Precision: 0.6902
[*] Recall: 0.7301
[*] F1 Score: 0.6609
[*] Classification Report:

              precision    recall  f1-score   support

           0       0.72      0.20      0.31       758
           1       0.73      0.97      0.84      2128
           2       0.00      0.00      0.00       160

    accuracy                           0.73      3046
   macro avg       0.48      0.39      0.38      3046
weighted avg       0.69      0.73      0.66      3046
```

Random Forest's Strength: Random Forest delivered strong results by handling categorical and numerical data effectively and reducing overfitting through ensemble learning. However, its performance did not surpass KNN by a significant margin. While Random Forest benefits from its ability to select subsets of features at each decision split, improving efficiency in high-dimensional data, it may not be as adept at capturing local patterns compared to KNN. Additionally, Random Forest's reliance on averaging multiple decision trees can sometimes smooth out finer distinctions in accident severity, limiting its edge over simpler models in certain cases.

```
[*] RF_Injury Results:
[*] Accuracy: 0.7403
[*] Precision: 0.7409
[*] Recall: 0.7403
[*] F1 Score: 0.6638
[*] Classification Report:

              precision    recall  f1-score   support

           0       0.93      0.18      0.30       758
           1       0.73      1.00      0.84      2128
           2       0.00      0.00      0.00       160

    accuracy                           0.74      3046
   macro avg       0.55      0.39      0.38      3046
weighted avg       0.74      0.74      0.66      3046
```

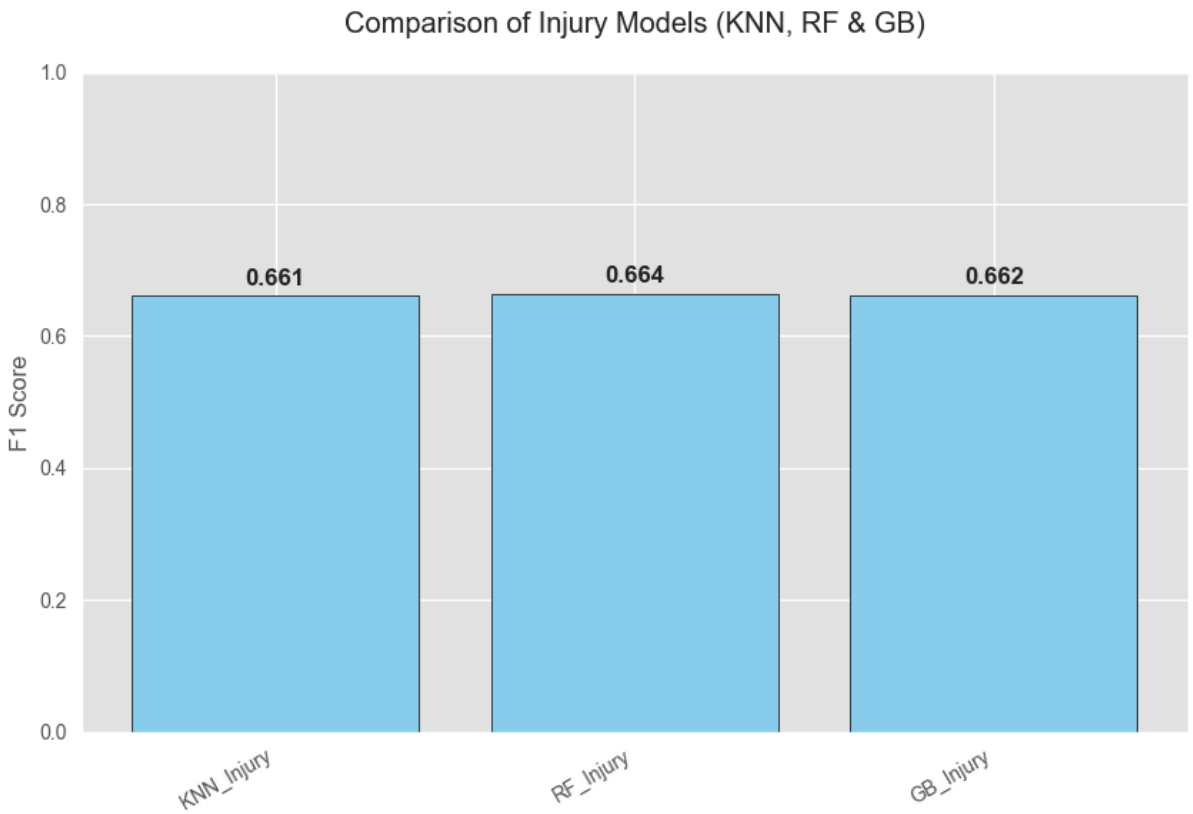
Gradient Boosting: While Gradient Boosting delivered strong predictive accuracy, particularly in identifying severe accidents, it outperformed other models, but not by a significant margin. KNN performed surprisingly well, capturing local patterns effectively despite the curse of dimensionality. However, Gradient Boosting remains the best-performing model overall due to its ability to handle complex relationships and imbalanced datasets, making it particularly useful for high-risk accident detection.

```
[*] GB_Injury Results:
[*] Accuracy: 0.7360
[*] Precision: 0.7349
[*] Recall: 0.7360
[*] F1 Score: 0.6617
[*] Classification Report:

              precision    recall  f1-score   support

           0       0.88      0.18      0.30       758
           1       0.73      0.99      0.84      2128
           2       0.09      0.01      0.01       160

    accuracy                           0.74      3046
   macro avg       0.57      0.39      0.38      3046
weighted avg       0.73      0.74      0.66      3046
```



Logistic Regression for Binary Classification: Logistic Regression provided a reliable method for distinguishing between accidents with and without casualties. This is especially important given that most accidents in the dataset result in zero or only one injury, reinforcing the need for a model that can effectively distinguish between low- and high-risk cases.

```
Model Performance Metrics:
Accuracy: 0.7607
Precision: 0.7613
Recall: 0.9987
F1 Score: 0.8640

Classification Report:
              precision    recall  f1-score   support

           0       0.25      0.00      0.00       727
           1       0.76      1.00      0.86      2319

    accuracy                           0.76      3046
   macro avg       0.51      0.50      0.43      3046
weighted avg       0.64      0.76      0.66      3046


Confusion Matrix:
[[   1  726]
 [   3 2316]]
```

Overall, our findings confirm that this problem is best approached as a classification task rather than a regression problem, which directly aligns with our goal of analyzing and predicting accident trends. The classification models enabled us to distinguish between different severity levels, providing a clearer understanding of high-risk situations for E-scooters and E-bikes. Among these models, Gradient Boosting proved to be the most effective for predicting accident severity, while Logistic Regression provided valuable insights for broad casualty prediction. These findings reinforce the importance of using classification

techniques to identify key contributing factors, high-risk locations, and accident-prone time periods. By leveraging these insights, policymakers and transportation planners can develop targeted safety measures to mitigate the risks associated with E-scooter and E-bike accidents.

Clustering Algorithms:

The primary reason for using clustering in this context was to identify natural groupings in accident data that might reveal patterns associated with injury/death outcomes. In total, we have tested 3 clustering algorithms [24].

K-Means:

This is the simplest and efficient clustering algorithm. It identifies distinct groups in our accident data based on feature similarity [25]. It identified 3 distinct clusters based on silhouette scores. KMeans performed well in segmenting accident severity levels but was sensitive to initialization and required careful tuning of n_clusters.

DBSCAN:

It is a density-based clustering. This algorithm failed to find any meaningful clusters. It classified 100% of the data (15,226 points) as noise points. This suggests the data doesn't have clear density-based clusters with the specified parameters [26].

Hierarchical Clustering [27]: It's a data analysis technique that groups data into a tree of clusters. It formed slightly different groupings than K-Means. It successfully separated data into meaningful groups based on temporal patterns. It was applied to study hierarchical relationships among different levels of severity such that clusters are analyzed through a dendrogram method.

Performance Metrics

Silhouette Score: Used to measure the quality of clustering by evaluating cohesion and separation.

Inertia (for KMeans): Measured within-cluster variance to assess compactness.

Insights Gained from Model Application

- K-means and Hierarchical clustering were effective
- DBSCAN completely failed with the chosen parameters

- Identified key accident groupings, such as high-severity clusters occurring during rush hours or weekends.
- Discovered that certain location-based features strongly influenced cluster formation.

```
K-Means Cluster Statistics:
               Total_Casualties       IsRushHour IsWeekend IsNightTime
                           mean count       mean      mean        mean
KMeans_Cluster
0                      0.831887  2189   0.000000  0.228415    1.000000
1                      0.814273  9767   0.484591  0.000000    0.000000
2                      0.817737  3270   0.395719  1.000000    0.099388

DBSCAN Cluster Statistics:
               Total_Casualties        IsRushHour IsWeekend IsNightTime
                           mean  count       mean      mean        mean
DBSCAN_Cluster
-1                     0.817549  15226   0.395836  0.247603    0.165112

Hierarchical Cluster Statistics:
                     Total_Casualties        IsRushHour IsWeekend IsNightTime
                                 mean  count       mean      mean        mean
Hierarchical_Cluster
0                            0.799980  10004   0.479108  0.016094    0.011295
1                            0.863390   2401   0.000000  0.328197    1.000000
2                            0.840837   2821   0.437434  1.000000    0.000000
```

References:

[1] NTSB, 'Micromobility: Data Challenges Associated with Assessing the Prevalence and Risk of Electric Scooter and Electric Bicycle Fatalities and Injuries', https://www.ntsb.gov/safety/safety-studies/Documents/SRR2201.pdf

[2] Wikipedia, National Highway Traffic Safety Administration, https://en.wikipedia.org/wiki/National_Highway_Traffic_Safety_Administration

[3] Vanderplas, J., Passos, A., Cournapeau, '6.3.4. Encoding categorical features', https://scikit-learn.org/stable/modules/preprocessing.html#encoding-categorical-features

[4] F. Noor, 'Working with Dates and Times in Pandas: A Comprehensive Guide', https://medium.com/@noorfatimaafzalbutt/working-with-dates-and-times-in-pandas-a-comprehensive-guide-fda47929ace4

[5] M. Lotfinejad, 'DateTime in Pandas: An Uncomplicated Guide (2023)', https://www.dataquest.io/blog/datetime-in-pandas/

[6] N. Rashvand, S. S. Hosseini, M. Azarbayjani, H. Tabkhi, 'Real-Time Bus Arrival Prediction: A Deep Learning Approach for Enhanced Urban Mobility', https://arxiv.org/html/2303.15495v3

[7] Sachinsoni, 'K Nearest Neighbours — Introduction to Machine Learning Algorithms', https://medium.com/@sachinsoni600517/k-nearest-neighbours-introduction-to-machine-learning-algorithms-9dbc9d9fb3b2

[8] S. Gode, 'K-Nearest Neighbors and Curse of Dimensionality', https://www.geeksforgeeks.org/k-nearest-neighbors-and-curse-of-dimensionality/

[9] P. Grant, 'k-Nearest Neighbors and the Curse of Dimensionality', https://medium.com/towards-data-science/k-nearest-neighbors-and-the-curse-of-dimensionality-e39d10a6105d

[10] Sruthi, 'Understanding Random Forest Algorithm With Examples', http://analyticsvidhya.com/blog/2021/06/understanding-random-forest/#:~:text=Random%20Forest%20Algorithm-,Advantages,data%20contains%20null%2Fmissing%20values.

[11] P. Barjatiya, 'Unleashing the Power of Random Forest: Why it Outperforms Decision Trees and Expert Rules',

https://pratikbarjatya.medium.com/unleashing-the-power-of-random-forest-why-it-outperforms-decision-trees-and-expert-rules-472a9bea1b8a

[12] IBM, 'What is random forest?', https://www.ibm.com/think/topics/random-forest

[13] M. D. Guillen, J. Aparicio, M. Esteve, 'Gradient tree boosting and the estimation of production frontiers', https://www.sciencedirect.com/science/article/pii/S0957417422021522

[14] A. Natekin, A. Knoll, 'Gradient boosting machines, a tutorial', https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2013.00021/full
[15] OPTUNA, 'Tutorial', https://optuna.readthedocs.io/en/stable/tutorial/
[16] T. Aggarwal, 'Master the Power of Optuna: A Step-by-Step Guide', https://medium.com/data-and-beyond/master-the-power-of-optuna-a-step-by-step-guide-ed43500e9b95
[17] OPTUNA, 'optuna.study', https://optuna.readthedocs.io/en/stable/reference/study.html
[18] T. Keldenich, 'Optuna: Get the Best out of your Hyperparameters – Easy Tutorial', https://inside-machinelearning.com/en/optuna-tutorial/
[19] P. Ranganathan, C. S. Pramesh, R. Aggarwal, 'Common pitfalls in statistical analysis: Logistic regression', https://pmc.ncbi.nlm.nih.gov/articles/PMC5543767/
[20] P. Hadjicostas, 'Consistency of logistic regression coefficient estimates calculated from a training sample', https://www.sciencedirect.com/science/article/abs/pii/S0167715203000361
[21] M. Treviso, Ji. Lee, T. Ji, 'Efficient Methods for Natural Language Processing: A Survey', https://direct.mit.edu/tacl/article/doi/10.1162/tacl_a_00577/116725/Efficient-Methods-for-Natural-Language-Processing
[22] N. Pogeant, 'Exploring NLP's Performance — Evaluation and Metrics as the Compass', https://npogeant.medium.com/exploring-nlps-performance-evaluation-and-metrics-as-the-compass-db2266422130
[23] D. Kirchhoff, 'NLP Model Evaluation - Metrics, Benchmarks, and Beyond', https://deconvoluteai.com/blog/evaluating-nlp-models
[24] M. Sena, 'Mastering K-Means Clustering', https://towardsdatascience.com/mastering-k-means-clustering-065bc42637e4/
[25] P. Sharma, 'What is K-Means Clustering?', https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/

[26] R. Kumar, 'A Guide to the DBSCAN Clustering Algorithm', https://www.datacamp.com/tutorial/dbscan-clustering-algorithm

[27] D. Dey, ‘Hierarchical Clustering in Machine Learning’, https://www.geeksforgeeks.org/hierarchical-clustering/