**Predicting Passenger Count Using Neural Networks with Weather and Transport Data**

**1. Introduction**

The goal of this project was to create a predictive model for estimating passenger counts using bus transportation data combined with weather data. By utilizing machine learning and deep learning techniques, particularly neural networks, the model aims to forecast demand based on temporal and environmental factors. The project involved comprehensive data preprocessing, feature engineering, hyperparameter tuning, and model evaluation.

**2. Data Preprocessing**

**2.1 Dataset Overview**

* **Bus Data**: Contains records of passengers boarding at various stations, with fields including timestamps, station names, and passenger counts.
* **Weather Data**: Provides meteorological data, such as temperature, precipitation, wind speed, and humidity, essential for understanding environmental influences on passenger demand.

**2.2 Data Cleaning and Preparation**

* **Column Removal**: Irrelevant columns, such as Snow Depth and Ground Condition Code, were removed from the weather dataset.
* **Handling Missing Values**: Rows with less than 70% valid data were discarded. Remaining missing values in numeric columns were filled using K-Nearest Neighbors (KNN) imputation, leveraging data from nearby observations to fill gaps.
* **Date and Time Conversion**: Datetime columns were transformed into day, hour, and day-of-week features, adding time-sensitive dimensions crucial for passenger prediction.

**2.3 Feature Engineering**

* **Passenger Aggregation**: Passenger counts were aggregated by station, date, day of the week, and hour to capture patterns at specific times and places.
* **Data Merging**: Bus and weather datasets were merged on date, hour, and day of the week, aligning weather conditions with passenger demand.
* **One-Hot Encoding**: Categorical variables like station names and day of the week were encoded into numerical formats through one-hot encoding, making them compatible with machine learning models.

**3. Model Development**

**3.1 Neural Network Fundamentals**  
A neural network is a computational model inspired by the human brain’s network of neurons. Composed of multiple interconnected layers, it processes data through layers of nodes, or neurons, each having an activation function to introduce non-linearity. Neural networks are effective in modeling complex relationships in large datasets, as they can capture intricate patterns that traditional methods may miss.

**3.2 Feedforward Neural Networks**  
A **feedforward neural network** is a basic form of neural network where information flows in one direction—from input to output—without any cycles. This type of network, suitable for tabular data, learns to minimize a loss function by adjusting the weights connecting neurons. Key components include:

* **Input Layer**: Holds the input features of the data, in this case, various factors like weather metrics, station details, and time of day.
* **Hidden Layers**: Capture the relationships within the data. By stacking multiple layers, the model can learn complex, non-linear relationships.
* **Output Layer**: Produces the prediction. For this model, the output layer predicts the number of passengers expected at a station based on the input features.

**3.3 Activation Functions**  
The model uses **ReLU (Rectified Linear Unit)** as the activation function in hidden layers. ReLU is effective because it introduces non-linearity, helping the network capture intricate patterns in the data while being computationally efficient.

**3.4 Backpropagation and Optimization**  
Neural networks learn through **backpropagation**, where errors from predictions are propagated backward through the network. By adjusting weights to minimize the error, the network iteratively improves. An **Adam optimizer** with a learning rate of 0.001 was used to optimize weight adjustments, balancing between speed and precision.

**4. Model Tuning and Evaluation**

**4.1 Feature Scaling**  
Data was standardized using **StandardScaler** for features and **MinMaxScaler** for the target variable. This approach ensures that features are on a similar scale, improving convergence and model accuracy.

**4.2 Hyperparameter Tuning**  
The neural network was tuned by varying parameters such as the number of neurons in each layer, learning rates, and batch sizes. A grid search evaluated different configurations, and the best-performing model had layers with sizes (64, 32, 16).

**4.3 Cross-Validation**  
To validate the model’s stability, **5-fold cross-validation** was used. Cross-validation splits data into five subsets, training the model on four and validating on the fifth. This method provided insights into the model’s performance across different data splits, offering a more robust measure of accuracy.

**5. Outlier Removal and Final Model Results**

**5.1 Outlier Detection and Removal**  
To enhance model accuracy, **outliers** were removed based on the interquartile range (IQR) of the target variable, passenger\_count. This step helped reduce the influence of extreme values that could skew predictions, resulting in improved model performance.

**5.2 Final Model Evaluation**  
The final model was trained on the refined dataset, and the evaluation on the test set showed:

* **MAE**: 1.59
* **MSE**: 5.32
* **R-squared (R²)**: 0.66

**5.3 Model Predictions**  
The model’s predictions were closer to the actual values after outlier removal, demonstrating its robustness and enhanced accuracy.

**6. Conclusion**

This project successfully demonstrated the use of neural networks for forecasting passenger demand by leveraging temporal, spatial, and environmental data. Through data preprocessing, feature engineering, and iterative model tuning, the model achieved a final MAE of 1.59. This level of accuracy can aid in optimizing public transportation resources, especially when integrated with real-time data sources. The approach highlights the impact of neural networks in complex predictive tasks and provides a framework for future demand forecasting models in public transport systems.