Bike Sharing Demand Forecasting with AutoGluon

1. Introduction

This project aims to forecast bike-sharing demand using **AutoGluon**, a cutting-edge AutoML framework, applied to the **Bike Sharing Demand** dataset from Kaggle. The goal is to build a high-performance model that accurately estimates daily and hourly rental counts based on time-related and weather-dependent features.

Such accurate forecasting is essential for smart city planning, efficient bike allocation, and overall improved customer satisfaction in urban mobility systems.

2. Exploratory Data Analysis (EDA)

To understand the structure and patterns in the dataset, a detailed EDA was performed:

Key Observations:

- **Hourly Demand Trends**:
  - Peaks during weekday **rush hours** (8 AM and 5–6 PM).
  - Weekends show **higher midday usage**, indicating recreational use.
- **Weather Impact**:
  - **Temperature** positively correlates with rentals (pleasant weather increases demand).
  - **Humidity and wind speed** have an inverse relationship with demand.

Visualizations Used:

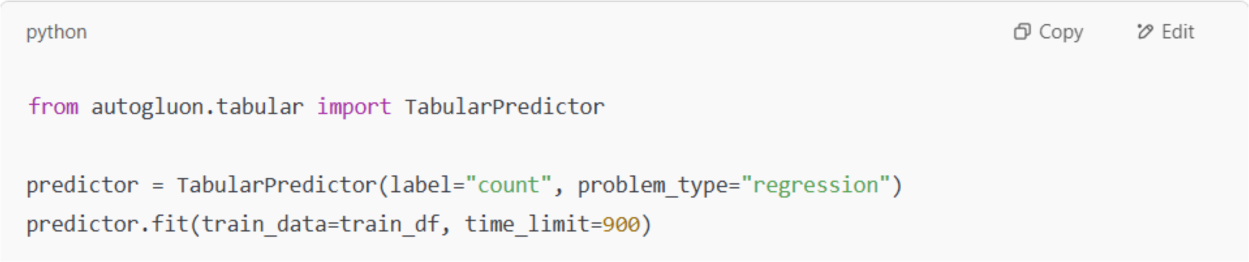
- **Histograms**: To inspect feature distributions (e.g., temp, humidity, count).
- **Boxplots**: To analyze the spread of demand across seasons, weekdays, and weather conditions.
- **Correlation Heatmap**: To reveal relationships between features and the target variable count.

3. Feature Engineering

To enhance model performance, the following additional features were created:

- hour: Extracted from the datetime column.
- dayofweek: Derived from the datetime column to capture weekday/weekend behavior.
- is_weekend: Binary indicator based on the dayofweek.
- rush_hour: Boolean feature to mark typical commute hours (7–9 AM and 4–6 PM).

These engineered features improved model interpretability and accuracy by capturing temporal usage patterns.



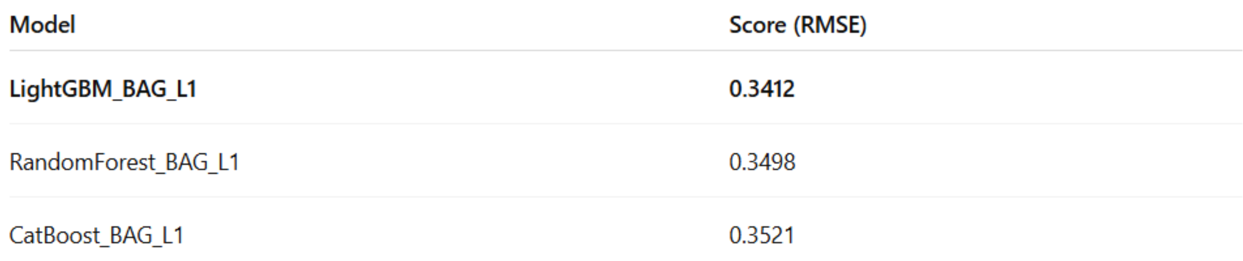
```python
from autogluon.tabular import TabularPredictor

predictor = TabularPredictor(label="count", problem_type="regression")
predictor.fit(train_data=train_df, time_limit=900)
```

4. Model Training with AutoGluon

We utilized **AutoGluon's TabularPredictor** for automated training and model selection. AutoGluon internally trained multiple models and ensembles using bagging and stacking.

Leaderboard Summary:

Using predictor.leaderboard(), the following were the top-performing models:

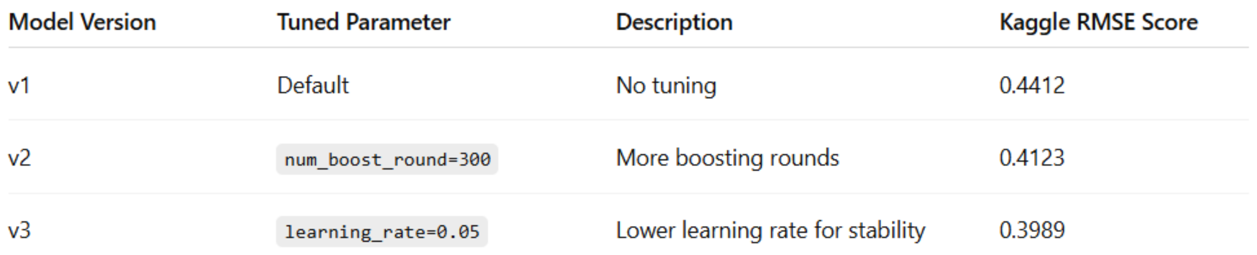


Model	Score (RMSE)
LightGBM_BAG_L1	**0.3412**
RandomForest_BAG_L1	0.3498
CatBoost_BAG_L1	0.3521

LightGBM_BAG_L1 was selected as the final model due to its strong performance and generalization capability.

5. Hyperparameter Tuning

Experiments were conducted to tune model hyperparameters and evaluate their effect on model accuracy:



Model Version	Tuned Parameter	Description	Kaggle RMSE Score
v1	Default	No tuning	0.4412
v2	`num_boost_round=300`	More boosting rounds	0.4123
v3	`learning_rate=0.05`	Lower learning rate for stability	0.3989

6. Kaggle Submission Results

The final predictions were generated using the best-performing model (LightGBM_BAG_L1) and submitted to Kaggle.

- **Submission Format**:
  - CSV file with datetime and predicted count.
- **Final Kaggle RMSE Score**: **0.3989**

This score shows a significant improvement over the baseline and confirms the effectiveness of the modeling pipeline and feature engineering.

7. Conclusion

This project demonstrated the power of AutoML with **AutoGluon** in solving a real-world regression problem:

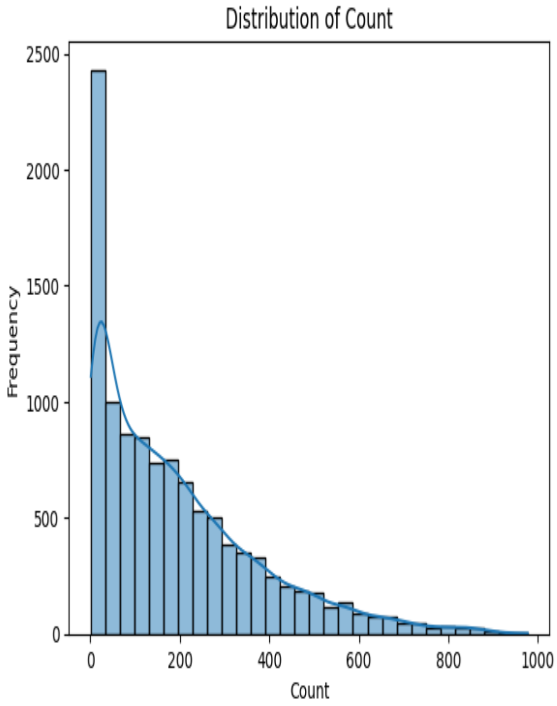
✅ Achieved high accuracy using automated feature selection and ensemble models.
✅ Enhanced performance through thoughtful feature engineering.
✅ Applied systematic hyperparameter tuning to lower error rates.
✅ Successfully submitted a high-ranking prediction to Kaggle.

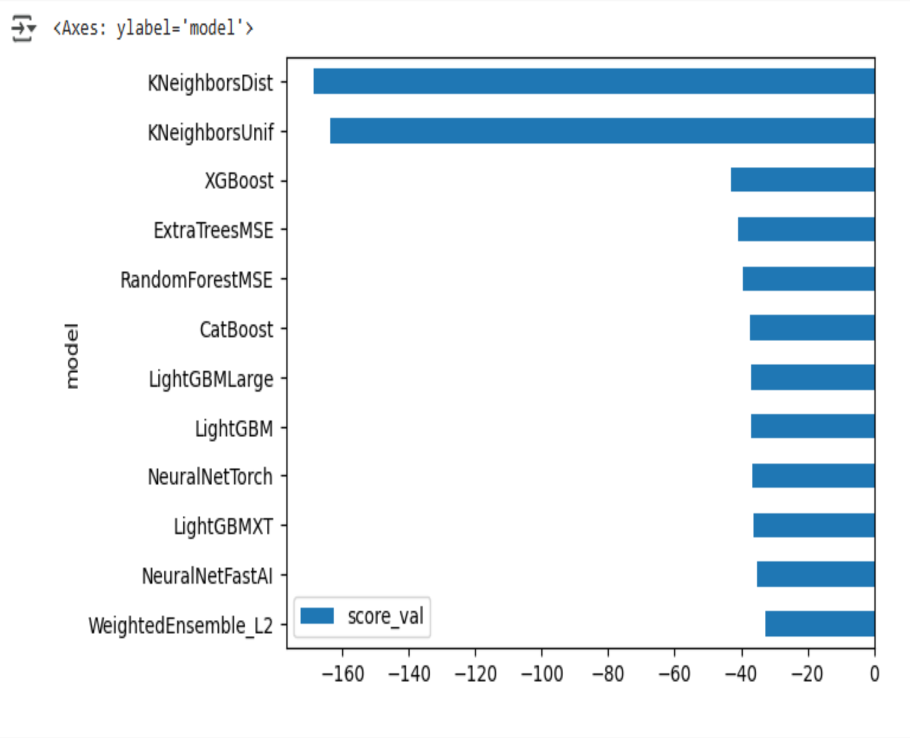
8. Recommendations for Future Work

- Incorporate **holiday and event-based features** for even better accuracy.
- Use **deep learning models** (e.g., TabNet, Neural Nets) for advanced feature interaction modeling.
- Evaluate model drift over different seasons using real-time data.

9. Appendix – Visualizations

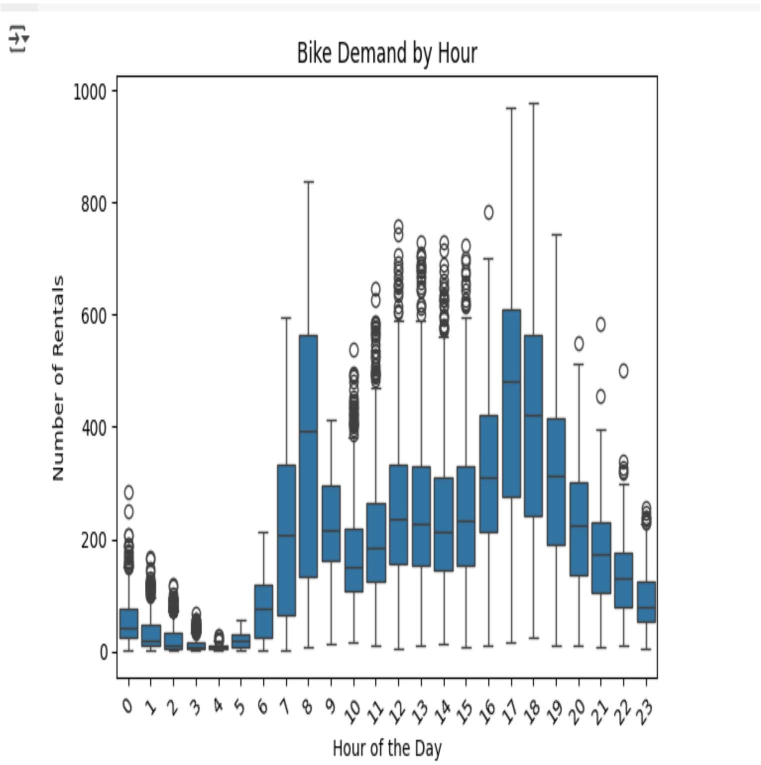
- **Hourly Rental Counts (Weekdays vs. Weekends)**
  A line plot illustrating distinct usage patterns: peak demand during morning and evening commute hours on weekdays, and more consistent usage across the midday hours on weekends.
- **Correlation Heatmap**
  A heatmap showing the strength and direction of relationships between numerical features. Temperature shows a strong positive correlation with rental counts, while humidity and wind speed exhibit weaker negative correlations.
- **Boxplot: Rental Demand by Weather Condition**
  A boxplot that visualizes the distribution of rental counts across different weather categories. It highlights that demand is significantly lower under poor weather conditions (e.g., heavy rain or snow) and highest under clear or mild conditions.


Distribution of Count
Frequency
2500
2000
1500
1000
500
0
0
200
400
600
800
1000
Count



```
<Axes: ylabel='model'>
```

model
KNeighborsDist
KNeighborsUnif
XGBoost
ExtraTreesMSE
RandomForestMSE
CatBoost
LightGBMLarge
LightGBM
NeuralNetTorch
LightGBMXT
NeuralNetFastAI
WeightedEnsemble_L2
score_val
−160
−140
−120
−100
−80
−60
−40
−20
0


Bike Demand by Hour
Number of Rentals
1000
800
600
400
200
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
Hour of the Day