CAPITAL BIKESHARE

BIKE SHARING

Present by Luqman Ilman M

linkedin.com/https://www.linkedin.com/in/luqmanilman/

luqman.ilman@gmail.com

CONTENTS

BUSINESS PROBLEM

- Context
- Problem Statement
- Goals
- Analytic Approach
- Metric Evaluation

CONTEXT

- Automated Rentals
- Global Presence
- Impactful Role
- Data-Driven Insights
- Research Potential

PROBLEM STATEMENT

- Balancing Availability
- Customer Trust
- Data Complexity
- Urban Insights
- Resource Optimization

GOALS

- Reliable Forecasting
- Variable Factors
- Profit and Efficiency
- User Satisfaction
- Urban Insights

ANALYTIC APPROACH

- Exploratory Data Analysis (EDA)
- Feature Engineering
- Regression Model Development
- Impact Analysis
- Model Deployment
- Actionable Recommendations

METRIC EVALUATION

- Mean Absolute Error (MAE) : To calculate the mean absolute value of the errors produced by the model.
- Mean Absolute Percentage Error (MAPE) : MAPE is used to calculate the percentage error produced by the model.
- R-Squared : R-Squared is used to see how significantly the independent variables affect the dependent variable.

DATA UNDERSTANDING

Information of Dataset “Bike Sharing”

Attributes Information

Attribute	Data Type	Description
dteday	Object	Date
hum	Float	Normalized humidity (the values are divided to 100)
weathersit	Integer	1: Clear, Few clouds, Partly cloudy, Partly cloudy 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
holiday	Integer	0: Not holiday 1: Holiday
season	Integer	1: Winter 2: Spring 3: Summer 4: Fall
atemp	Float	"Feels like" temperature in Celsius
temp	Float	Normalized temperature in Celsius
hr	Integer	Hour (0 to 23)
casual	Integer	Count of casual users
registered	Integer	Count of registered users
cnt	Integer	Count of total rental bikes including both casual and registered users

DATA PREPROCESSING

- Adjustment for name and value columns
- Change datatypes and separate 'date' columns
- Check missing values and duplicate data
- Drop column (Feature Selection)
- Data Correlation
- Checking Outliers
- Clean Dataset

DATA PREPROCESSING

1. Adjustment for name and value columns

	date	hour	humidity	weather	holiday	season	atemp	temp	casual	registered	count
0	2011-01-01	0	0.81	clear	0	winter	0.2879	0.24	3	13	16
1	2011-01-01	1	0.80	clear	0	winter	0.2727	0.22	8	32	40
2	2011-01-01	2	0.80	clear	0	winter	0.2727	0.22	5	27	32
3	2011-01-01	3	0.75	clear	0	winter	0.2879	0.24	3	10	13
4	2011-01-01	4	0.75	clear	0	winter	0.2879	0.24	0	1	1

DATA PREPROCESSING

2. Change datatypes and separate 'date' columns

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12165 entries, 0 to 12164
Data columns (total 14 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   date        12165 non-null  object
 1   hour        12165 non-null  int64
 2   humidity    12165 non-null  float64
 3   weather     12165 non-null  object
 4   holiday     12165 non-null  category
 5   season      12165 non-null  object
 6   atemp       12165 non-null  float64
 7   temp        12165 non-null  float64
 8   casual      12165 non-null  int64
 9   registered  12165 non-null  int64
 10  count       12165 non-null  int64
 11  month       12165 non-null  category
 12  year        12165 non-null  category
 13  dayname     12165 non-null  category
```

DATA PREPROCESSING

3. Check missing values and duplicate data

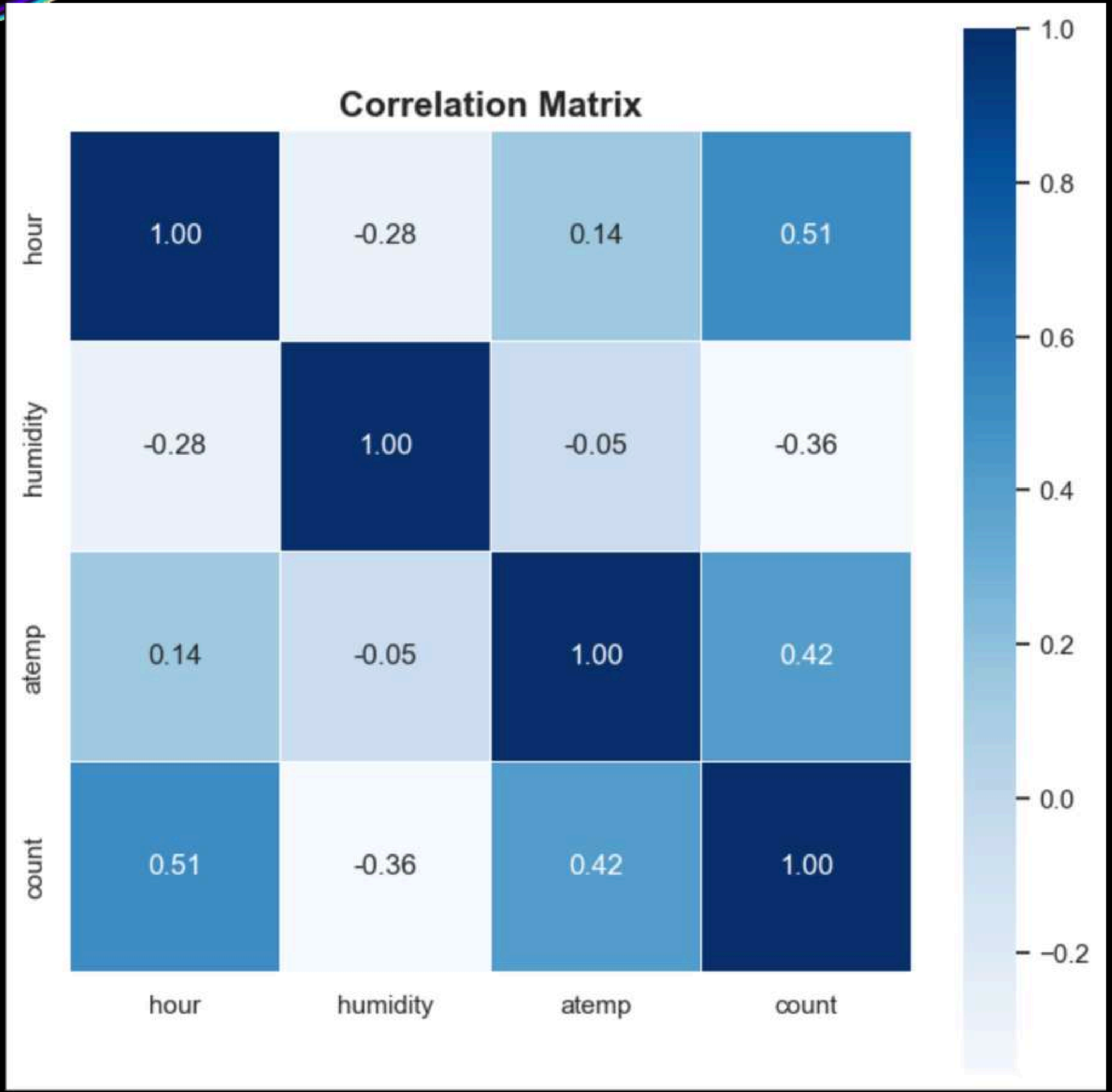
Missing Values = 0
Duplicate Data = False (0)

4. Drop column (Feature Selection)

Based on domain knowledge, **Casual** and **Registered** are directly related to the target. Additionally, the **date** column is no longer needed since its values are already represented by the `year`, `month`, and `day` columns.

DATA PREPROCESSING

5. Data Correlation (Matrix and VIF Score)



	variables	VIF
0	humidity	6.623596
1	atemp	339.222594
2	temp	306.416661
3	hour	3.918037
4	count	3.065671
5	dayname	2.928169

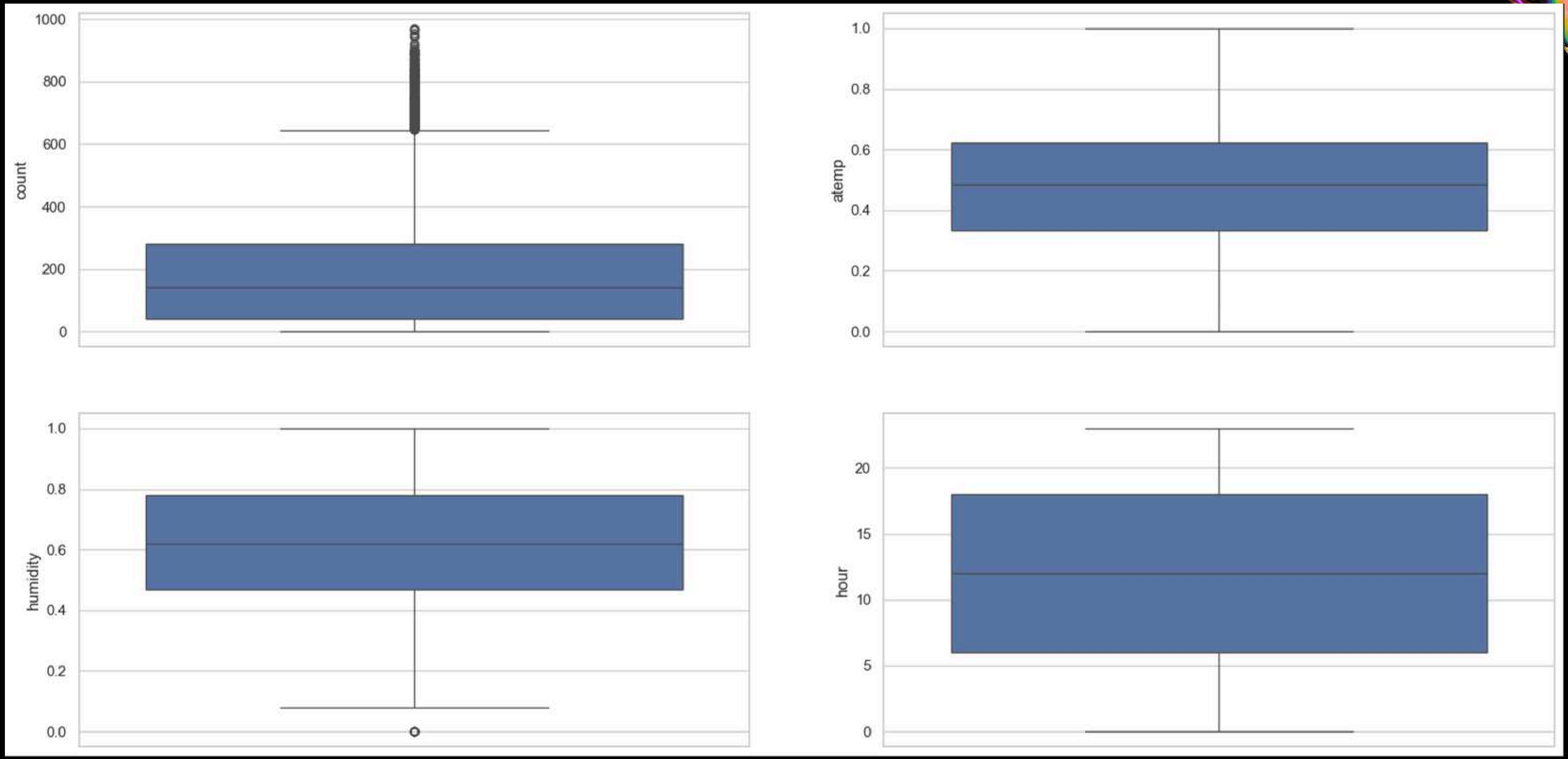
	variables	VIF
0	humidity	5.571588
1	atemp	8.360153
2	hour	3.837270
3	count	3.063059

DATA PREPROCESSING

6. Checking Outliers

Count (Right-Skewed):
upper bound : {645.0}
There is more than 300 is >645

Humidity :
There is 14 rows that has **humidity** value = 0



DATA PREPROCESSING

7. Clean Dataset

```
<class 'pandas.core.frame.DataFrame'>
Index: 12151 entries, 0 to 12164
Data columns (total 10 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   hour      12151 non-null  int64
 1   humidity  12151 non-null  float64
 2   weather   12151 non-null  object
 3   holiday   12151 non-null  category
 4   season    12151 non-null  object
 5   atemp     12151 non-null  float64
 6   count     12151 non-null  int64
 7   month     12151 non-null  category
 8   year      12151 non-null  category
 9   dayname   12151 non-null  category
dtypes: category(4), float64(2), int64(2), object(2)
memory usage: 712.7+ KB
```

MODELING

- ENCODING
- TRAIN AND TEST SPLITTING
- CHOOSE BENCHMARK MODEL
- PREDICT TEST USING BENCHMARK
- HYPERPARAMETER TUNING
- PERFORMANCE COMPARISON
- MODEL LIMITATION
- FEATURE IMPORTANCES

MODELING

ENCODING

Target: Count
Passthrough: Humidity, Temperature, Hour, Month, Year, Daynamne
OneHotEncoding: Season, Weather, Holiday

TRAIN AND TEST SPLITTING

Train : 70
Test: 30

MODELING

CHOOSE BENCHMARK MODEL

Base Model :

1. Linear Regression
2. K-Nearest Neighbors Regressor
3. Decision Tree Regressor

Ensemble Model :

1. Random Forest Regressor
2. Gradient Boosting Regressor
3. Extreme Gradient Boost Regressor

	Model	MAE	MAPE	R-squared
5	XGBoost Regressor	-26.964132	-0.257352	0.937008
3	RandomForest Regressor	-29.332776	-0.290825	0.925625
4	Gradient Boosting	-47.529187	-0.379345	0.817066
1	KNN Regressor	-40.993247	-0.394992	0.864377
2	DecisionTree Regressor	-41.030232	-0.439500	0.845962
0	Linear Regression	-106.474695	-1.361903	0.205071

From the results above, we know that XGBoost is the best model with the highest performance. The MAE score is 26.96, the MAPE is 0.25, and the R-Squared is approximately 0.94, which is higher than the other five models. Therefore, we can use the test set for prediction and benchmarking using the XGBoost model.

MODELING

CHOOSE BENCHMARK MODEL

What is XGBoost?

- Highly Efficient & Scalable: Delivers superior performance and speed.
- Sequential Decision Trees: Corrects errors iteratively for enhanced accuracy.

Key Features:

- Handling Missing Values: Learns paths for missing data.
- Regularization Techniques: Prevents overfitting using L1 and L2 regularization.
- Parallel Processing: Utilizes multiple CPU cores for faster training.
- Tree Pruning: Uses max depth pruning for optimal splits.
- Sparsity Awareness: Efficient with sparse data structures.
- Cross-validation: Accurate performance metrics and overfitting prevention.
- Non-interpretable Model: Difficult to determine incorrect variable predictions.
- Robust Performance: Reliable for various data types and use cases.

Source :
XGBoost: A Scalable Tree Boosting System
XGBoost Documentation
GeeksforGeeks XGBoost Article

MODELING

PREDICT TEST USING BENCHMARK

	MAE	MAPE	R-squared
XGB	25.315776	0.248794	0.950021

From score above, we now that predict using test set XGBoost still have best performance. There is MAE and MAPE is decreasing and the R2 score is increasing.

MODELING

HYPERPARAMETER TUNING

PARAMETER	VALUE
max_depth	2, 3, 4, 5, 6, 7, 8, 9
learning_rate	0.1, 0.001, 0.0001, 0.2, 0.3, 0.5, 0.7
n_estimator	200, 220, 240, 260, 280, 300

GridSearch: Systematically explores combinations of hyperparameter values to find the best performance.

Key Hyperparameters:

1. Tree Depth (max_depth):
   - Controls maximum tree depth: Captures complex patterns; risk of overfitting.
2. Learning Rate (learning_rate):
   - Step size shrinkage: Prevents overfitting; smaller values yield more accurate models but require more trees.
3. Number of Trees (n_estimators):
   - Sets number of trees: More trees can improve performance but increase computation time and risk of overfitting.

Goal: Balance model complexity and generalization to new data.

MODELING

HYPERPARAMETER TUNING

PARAMETER	VALUE
max_depth	8
learning_rate	0.1
n_estimator	200

Test score after using best parameter

	MAE	MAPE	R-squared
XGB	23.970842	0.243486	0.953526

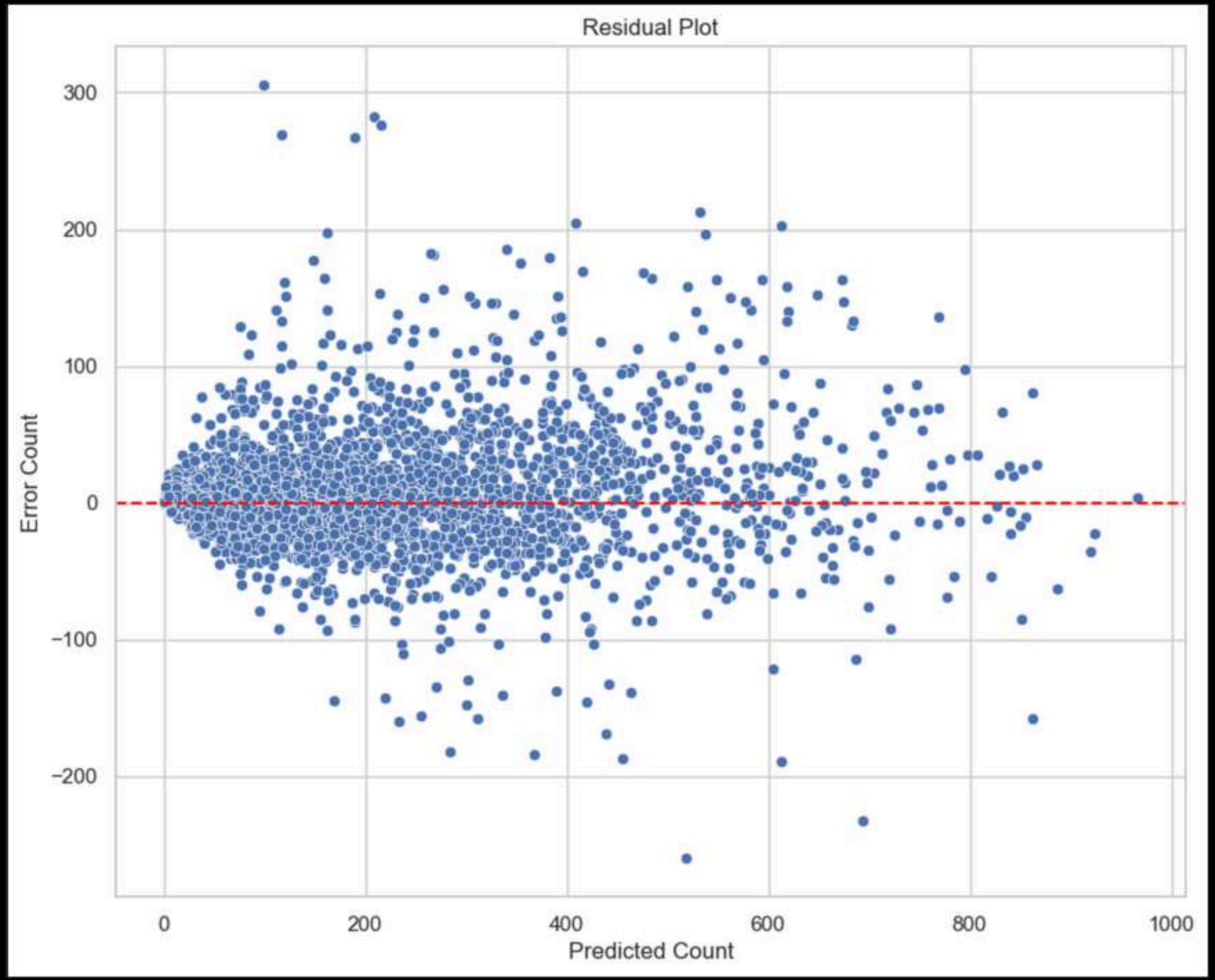
MODELING

PERFORMANCE COMPARISON

MODEL	MAE	MAPE	R-SQUARED
XGBoost Test Before Tuning	25.970842	0.248794	0.950021
XGBoost Test After Tuning	23.970842	0.243486	0.953526

MODELING

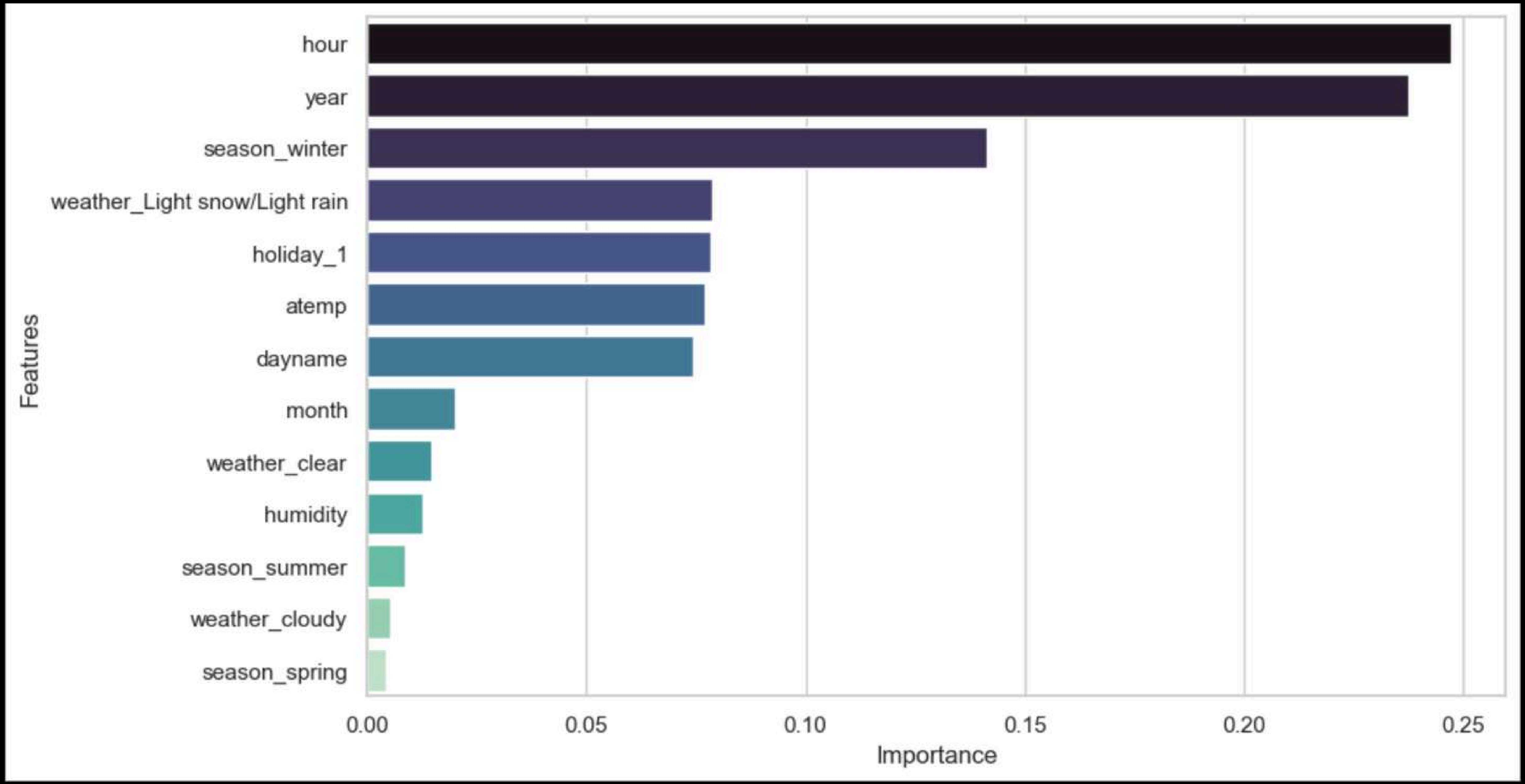
MODEL LIMITATION



	Score MAE	Score MAPE
<=50	6.625687	0.503021
51-100	16.658716	0.225928
101-150	21.468126	0.172045
151-200	25.440823	0.146815
201-250	25.862781	0.116267
251-300	32.055356	0.116432
301-350	34.896643	0.108170
351-400	39.623156	0.105902
401-450	43.183431	0.101038
451-500	55.399033	0.116599
501-550	48.238999	0.091597
551-600	45.043717	0.078734
>600	61.259912	0.085578
All Count Range (Max 953)	23.970842	0.243486

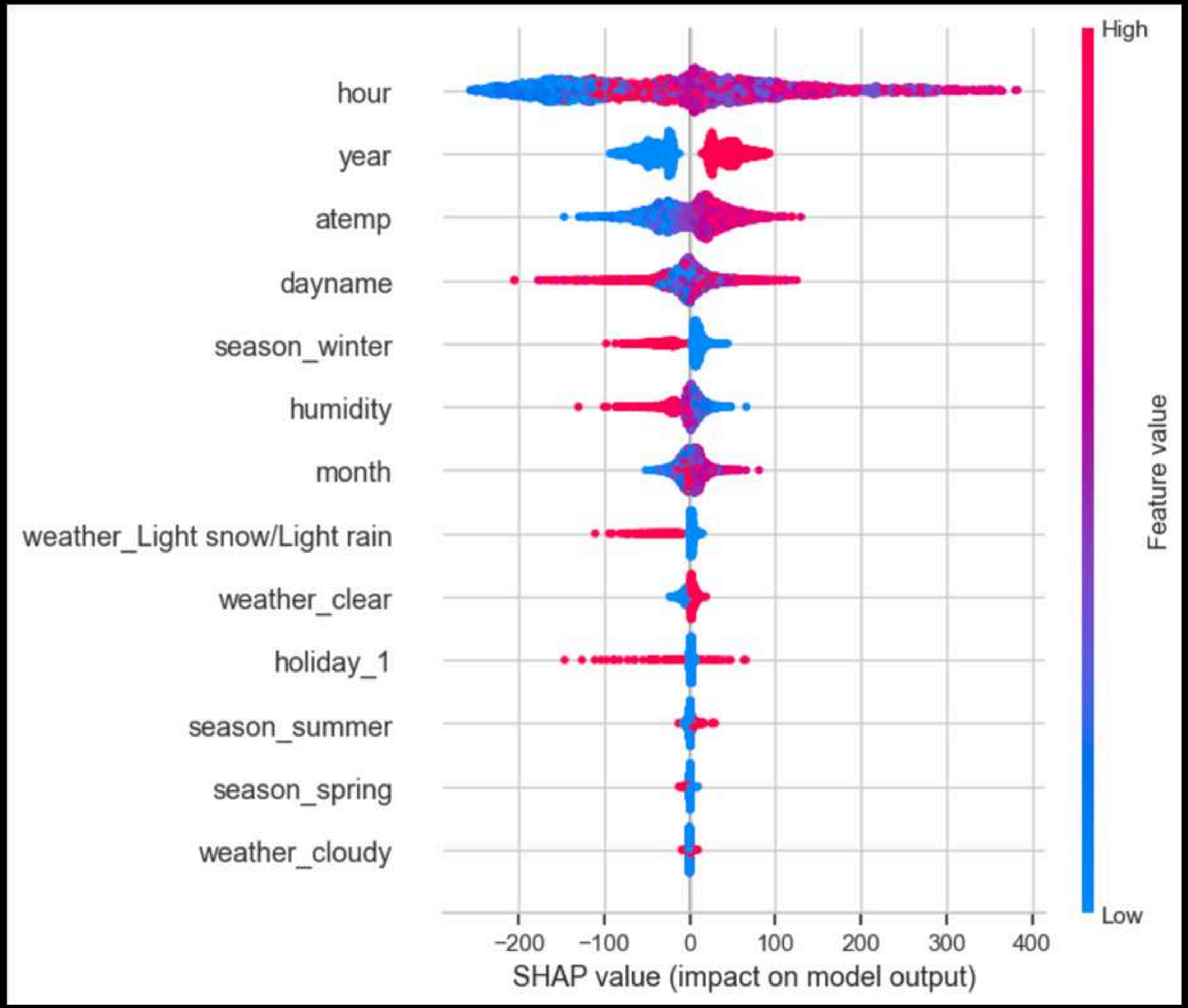
MODELING

FEATURE IMPORTANCES



MODELING

EXPLAINABLE AI



CONCLUSION

- Best model is XGBoost Regressor.
- Optimal hyperparameters: n_estimators 200, max_depth 8, learning_rate 0.1.
- Evaluation metrics: MAE 23.97, MAPE 0.24, R-squared 0.9535 .
- Total Bike prediction up to 980 bikes, the prediction is about 24%.
- Model limitations: low range (<=50) MAE 6.63, MAPE 50.30%; medium to high range (51-600) MAE 16.66-45.04, decreasing MAPE; struggles with very low and very high values.
- Feature importance: most influential are hour, year, season_winter; SHAP highlights hour, year, atemp.

RECOMMENDATION

- Addition of Relevant Features
- Expand Data Range
- Develop Additional Models
- Leverage Data Characteristics
- Continuos Model Improvement
- Optimize Bike Distribution

CAPITAL BIKESHARE

THANK YOU

Present by Luqman Ilman M

[linkedin.com/https://www.linkedin.com/in/luqmanilman/](https://www.linkedin.com/in/luqmanilman/)

luqman.ilman@gmail.com