Bike Sharing Rental P 417

Syed Aftab Ahmed

(Group 2)

Bike Rental Sharing

The business problem is to ensure a stable supply of rental bikes in urban cities by predicting the demand for bikes. By providing a stable supply of rental bikes, the system can enhance mobility comfort for the public and reduce waiting time, leading to greater customer satisfaction and accurately predicting bike demand can help bike sharing companies optimize operations including bike availability, pricing, strategies, and marketing efforts by considering demand Based on various external factors such as weather, season, holiday etc..,

Data Set Over View

instant	dteday	season	yr	mnth	hr	holiday	weekday	workingda	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	1/1/2011	springer	2011	1	0	No	6	No work	Clear	0.24	0.2879	0.81	0	3	13	16
2	1/1/2011	springer	2011	1	1	No	6	No work	Clear	0.22	0.2727	0.8	0	8	32	40
3	1/1/2011	springer	2011	1	2	No	6	No work	Clear	0.22	0.2727	?	0	5	27	32
4	1/1/2011	springer	2011	1	3	No	6	No work	Clear	0.24	0.2879	0.75	0	3	10	13
5	1/1/2011	springer	2011	1	4	No	6	No work	Clear	0.24	0.2879	0.75	0	0	1	1
6	1/1/2011	springer	2011	1	5	No	6	No work	Mist	0.24	0.2576	0.75	0.0896	0	1	1
7	1/1/2011	springer	2011	1	6	No	6	No work	?	0.22	0.2727	0.8	0	2	0	2
8	1/1/2011	springer	2011	1	7	No	6	No work	Clear	0.2	0.2576	0.86	0	1	2	3
9	1/1/2011	springer	2011	1	8	No	6	No work	Clear	0.24	0.2879	0.75	0	1	7	8
10	1/1/2011	springer	2011	1	9	No	6	No work	Clear	?	0.3485	0.76	0	8	6	14
11	1/1/2011	springer	2011	1	10	No	6	No work	Clear	0.38	0.3939	0.76	0.2537	12	24	36
12	1/1/2011	springer	2011	1	11	No	6	No work	Clear	0.36	?	0.81	0.2836	26	30	56
13	1/1/2011	springer	2011	1	12	No	6	No work	Clear	0.42	0.4242	0.77	?	?	?	84
14	1/1/2011	springer	2011	1	13	No	6	No work	Mist	0.46	0.4545	0.72	0.2985	47	47	94
15	1/1/2011	springer	2011	1	14	No	6	No work	Mist	0.46	0.4545	0.72	0.2836	35	71	106
16	1/1/2011	?	2011	1	15	No	6	No work	Mist	0.44	0.4394	0.77	0.2985	40	70	110
17	1/1/2011	springer	2011	1	16	No	6	No work	Mist	0.42	0.4242	0.82	0.2985	41	52	93
18	1/1/2011	springer	2011	1	17	No	6	No work	Mist	0.44	0.4394	0.82	0.2836	15	52	67
19	1/1/2011	springer	2011	1	18	No	6	No work	Light Snow	?	0.4242	0.88	0.2537	9	26	35

Libraries Used

- Pandas
- NumPy
- Matplotlib
- Seaborn
- Scikit learn
- TensorFlow
- Streamlit

EDA

EDA – Descriptive Statistics

	instant	hr	weekday	temp	atemp	hum	windspeed	casual	registered	cnt
count	17379.0000	17379.000000	17379.000000	17368.000000	17373.000000	17373.000000	17374.000000	17378.000000	17378.000000	17379.000000
mean	8690.0000	11.546752	3.003683	0.497132	0.475851	0.627208	0.190080	35.676603	153.792554	189.463088
std	5017.0295	6.914405	2.005771	0.192525	0.171829	0.192939	0.122321	49.306423	151.359786	181.387599
min	1.0000	0.000000	0.000000	0.020000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000
25%	4345.5000	6.000000	1.000000	0.340000	0.333300	0.480000	0.104500	4.000000	34.000000	40.000000
50%	8690.0000	12.000000	3.000000	0.500000	0.484800	0.630000	0.194000	17.000000	115.000000	142.000000
75%	13034.5000	18.000000	5.000000	0.660000	0.621200	0.780000	0.253700	48.000000	220.000000	281.000000
max	17379.0000	23.000000	6.000000	1.000000	1.000000	1.000000	0.850700	367.000000	886.000000	977.000000

- we can see that casual ,registered and cnt have high Std which tells us that data is spread out and outliers can be there

EDA – Data cleaning

- No null values were found in the dataset.
- No duplicate entries were present
- Some columns were in object type and needed conversion to appropriate data types (e.g., date columns, categorical variables).
- Some entries had special values (e.g., '?'). These were handled appropriately (e.g., replacing with a default value, imputing, or removing).

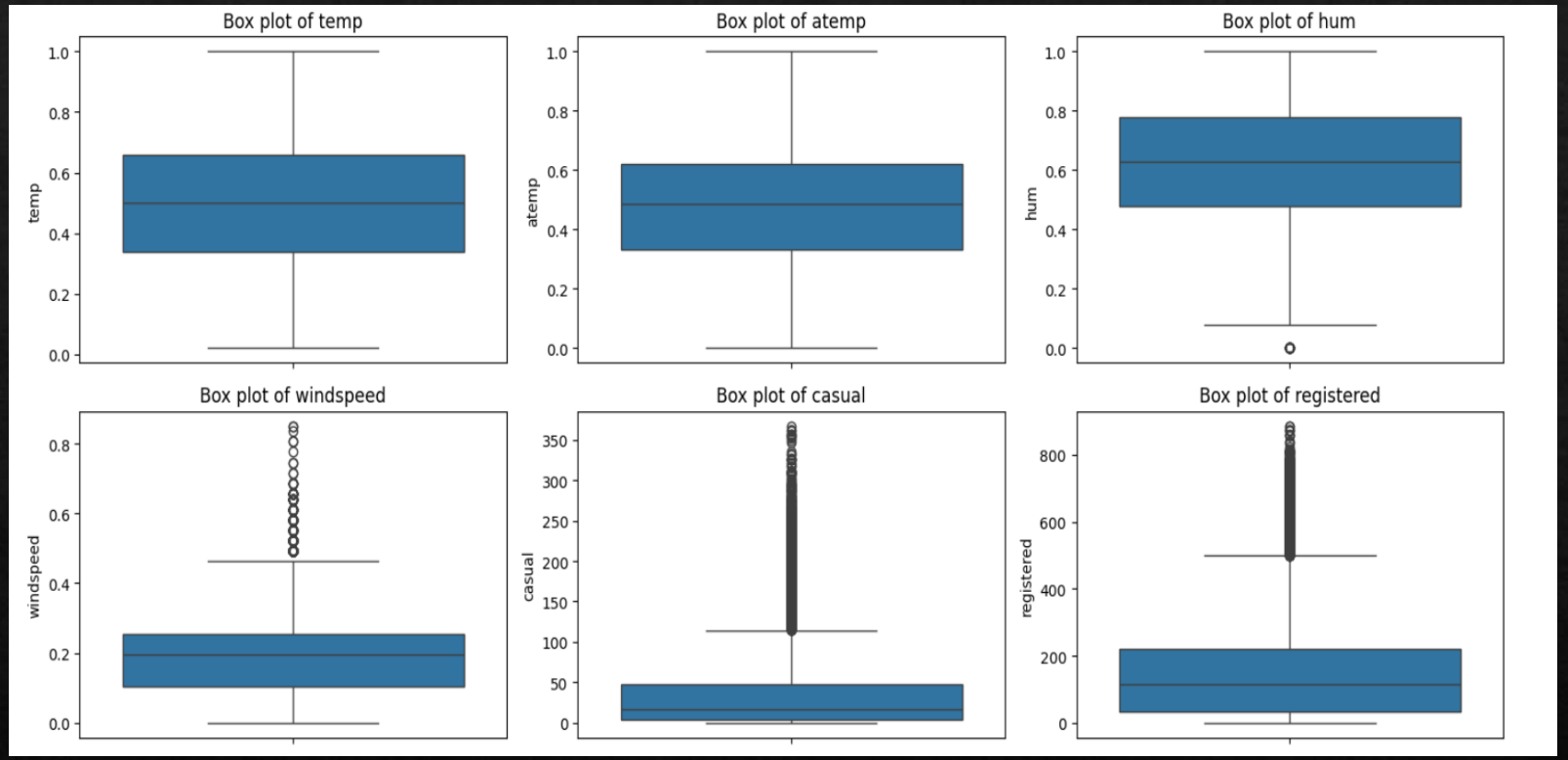
EDA Data Transformation

```
# Check for invalid records and convert object columns to appropriate numeric types
for column in ['temp', 'atemp', 'hum', 'windspeed', 'casual', 'registered']:
    data[column] = pd.to_numeric(data[column], errors='coerce')
```

- Converting the data from object type to numeric for a suitable format

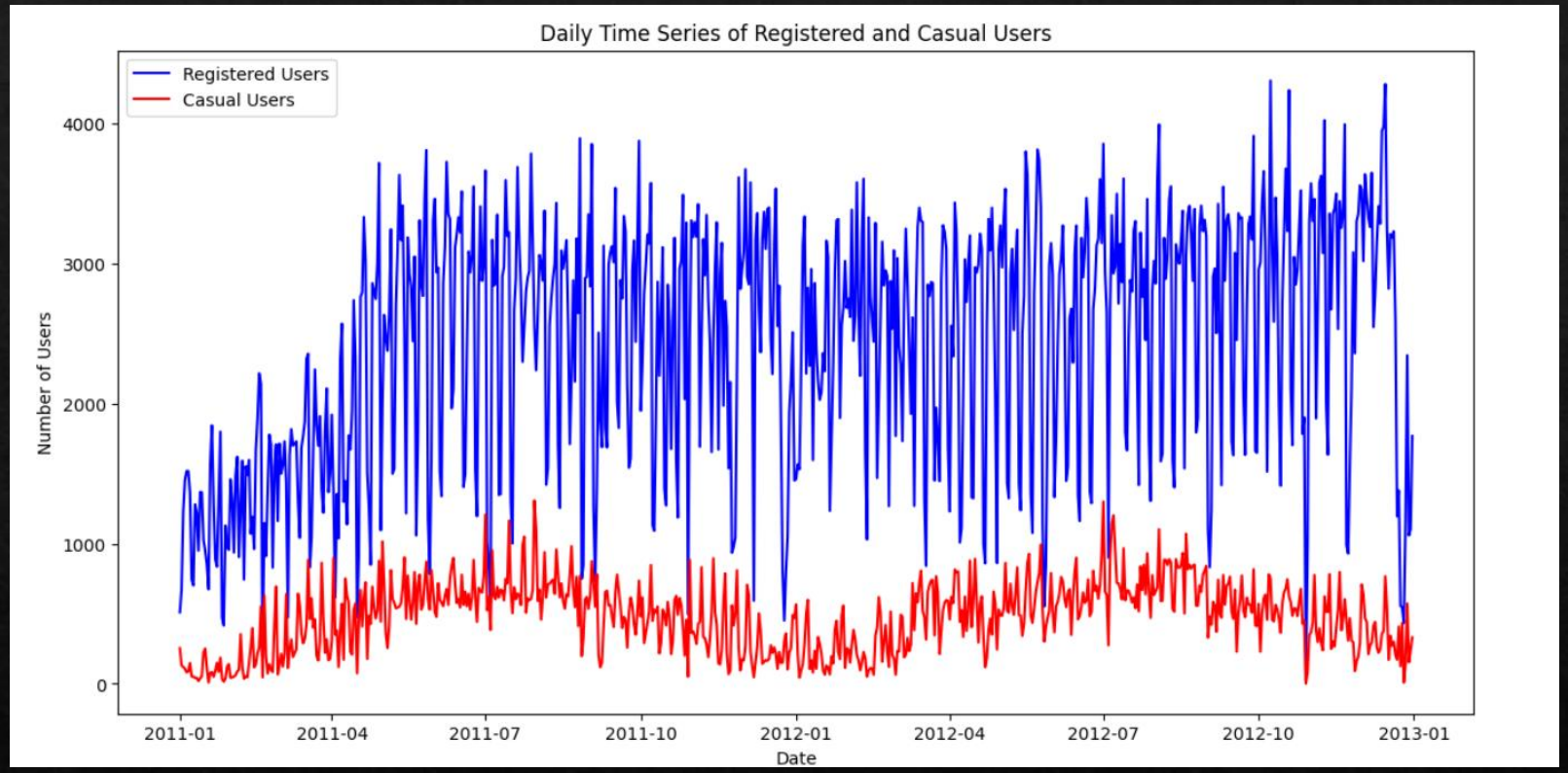
Data Visualization

Data Visualization Boxplot



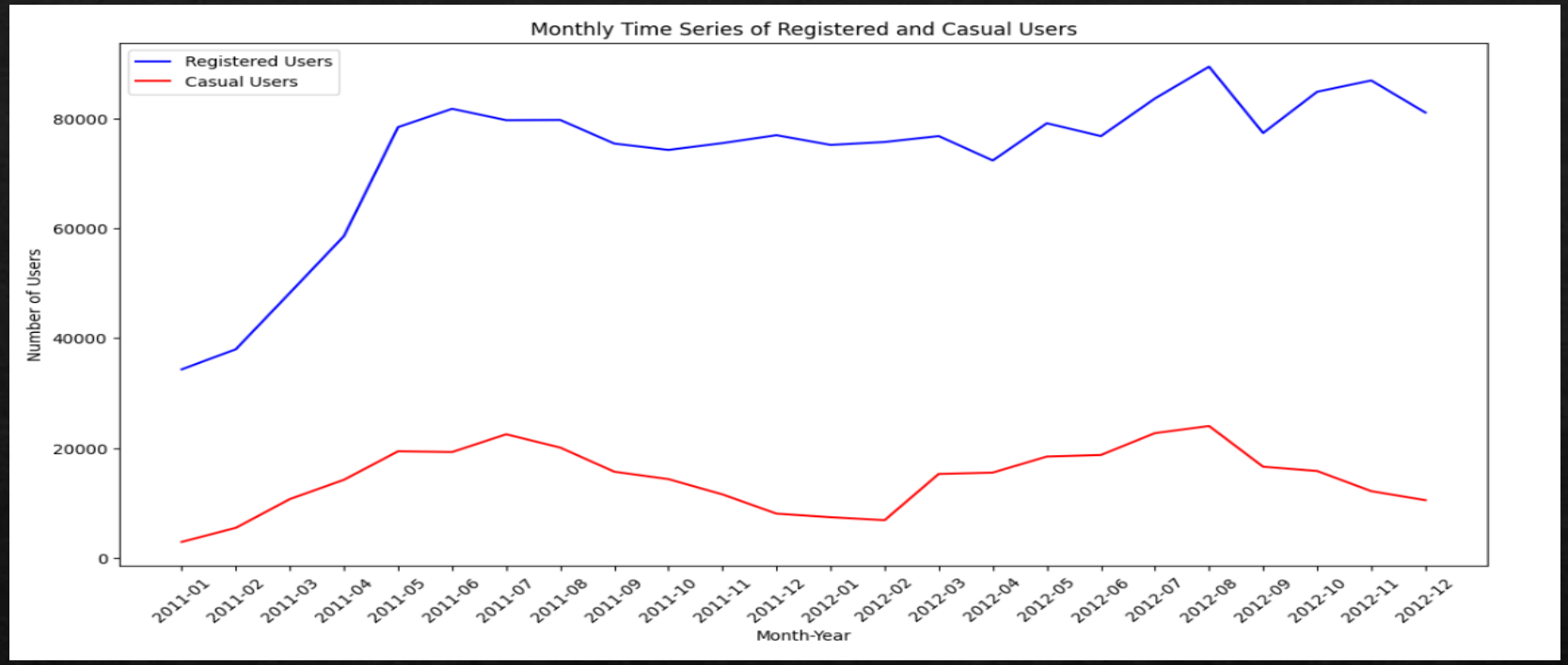
- The boxplot confirmed our insights drawn from earlier statistics , there are high number of outliers in casual, cnt and registered . Additionally we found out there are some outliers in windspeed

Data Visualization Timeseries



- The registered users peak at October and the casual users peak at July

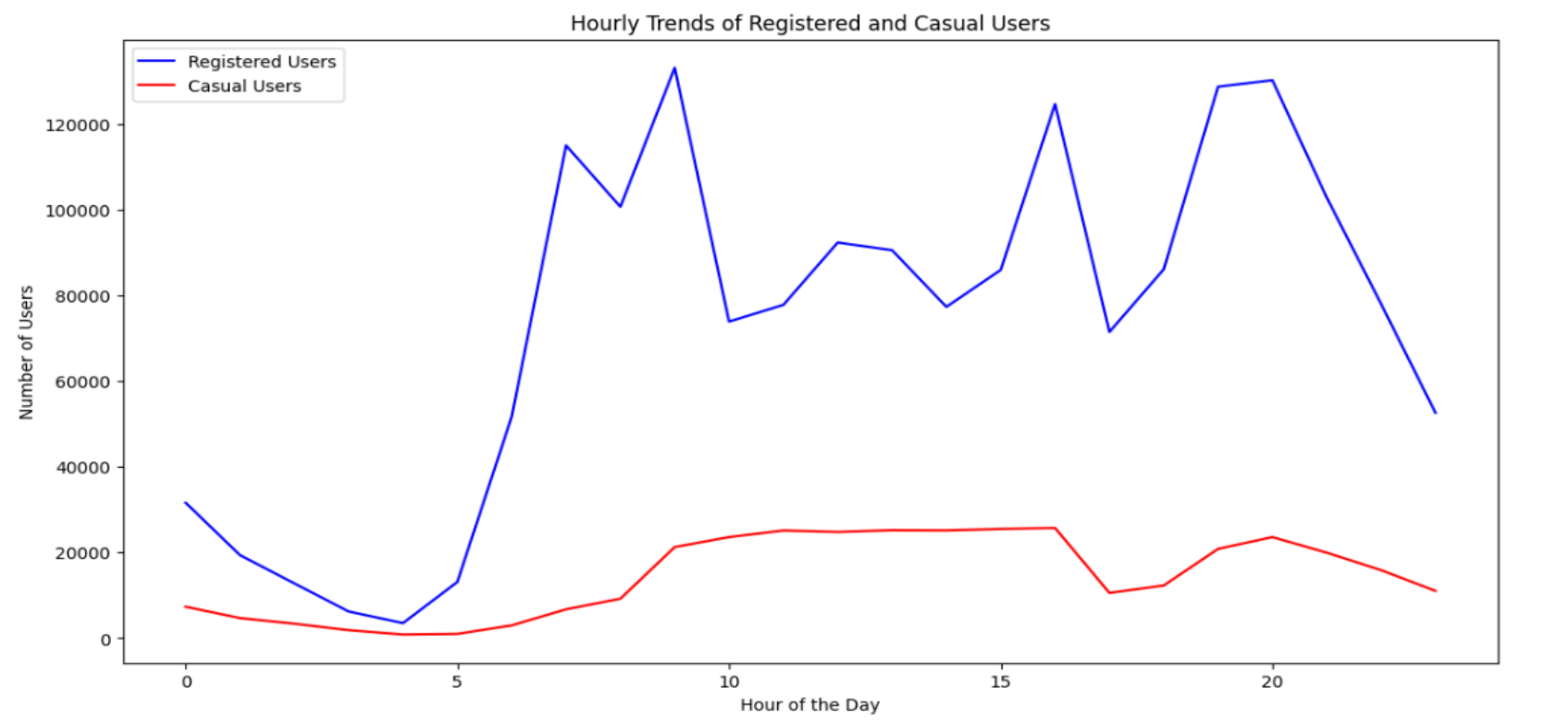
Data Visualization Timeseries



- Casual Users: Casual users peak every 2nd quarter and experience a drop every 4th quarter of the year.

- Registered Users: Registered users peaked in the 2nd quarter of 2011 and the 3rd quarter of 2012, showing a general upward trend over time with minor variations.

Data Visualization Timeseries



- Registered Users peak during early hours in day and during evening
- Casual users peak during the afternoon

Model Development

Model Development - Data Preparation

- Categorical Encoding:
  Columns to Encode: `season`, `weathersit`, `holiday`, `workingday`
  Method: Label Encoding
  Converts categorical values into numeric codes
  Example: `Spring` → 1, `Summer` → 2, etc.

- Data Cleaning:
  Drop Unnecessary Columns:
  Removed columns that are not needed for modeling: `dteday`, `instant`

- Feature and Target Variable Definition:
  Features (X): Exclude target and other non relevant columns
  Target (y): `cnt` (total bike rentals)

Model Development - Data Preparation

- Ensure Numeric Data:
  Select Numeric Columns: Retain only numeric data types
  Convert Non Numeric Values: Handle any remaining non numeric values

- Feature Scaling:
  Method: StandardScaler
  Standardizes features by removing the mean and scaling to unit variance

Data Splitting for Model Training

- Objective: Split the dataset into training and test sets to evaluate model performance.
- Method: `train_test_split` from Scikit-learn
- Process:
  - Features (X) and Target Variable (y) are split
  - Indices are preserved to keep track of original data rows
- Details:
  - Training Set Size: 80% of the data
  - Test Set Size: 20% of the data
  - Random State: 42 (for reproducibility)
- Outcome: The dataset is now divided into training and test subsets, with indices retained for reference.

Model Comparison

Model	Root Mean Squared Error (RMSE)	Mean Squared Error (MSE)	R^2 Score
Decision Tree Regressor	47.22	2229.94	0.85
Gradient Boosting Regressor	50.42	2542.22	0.83
Random Forest Regressor	34.17	1167.91	0.92
AdaBoost Regressor	73.97	5472.22	0.63

Key Insights:

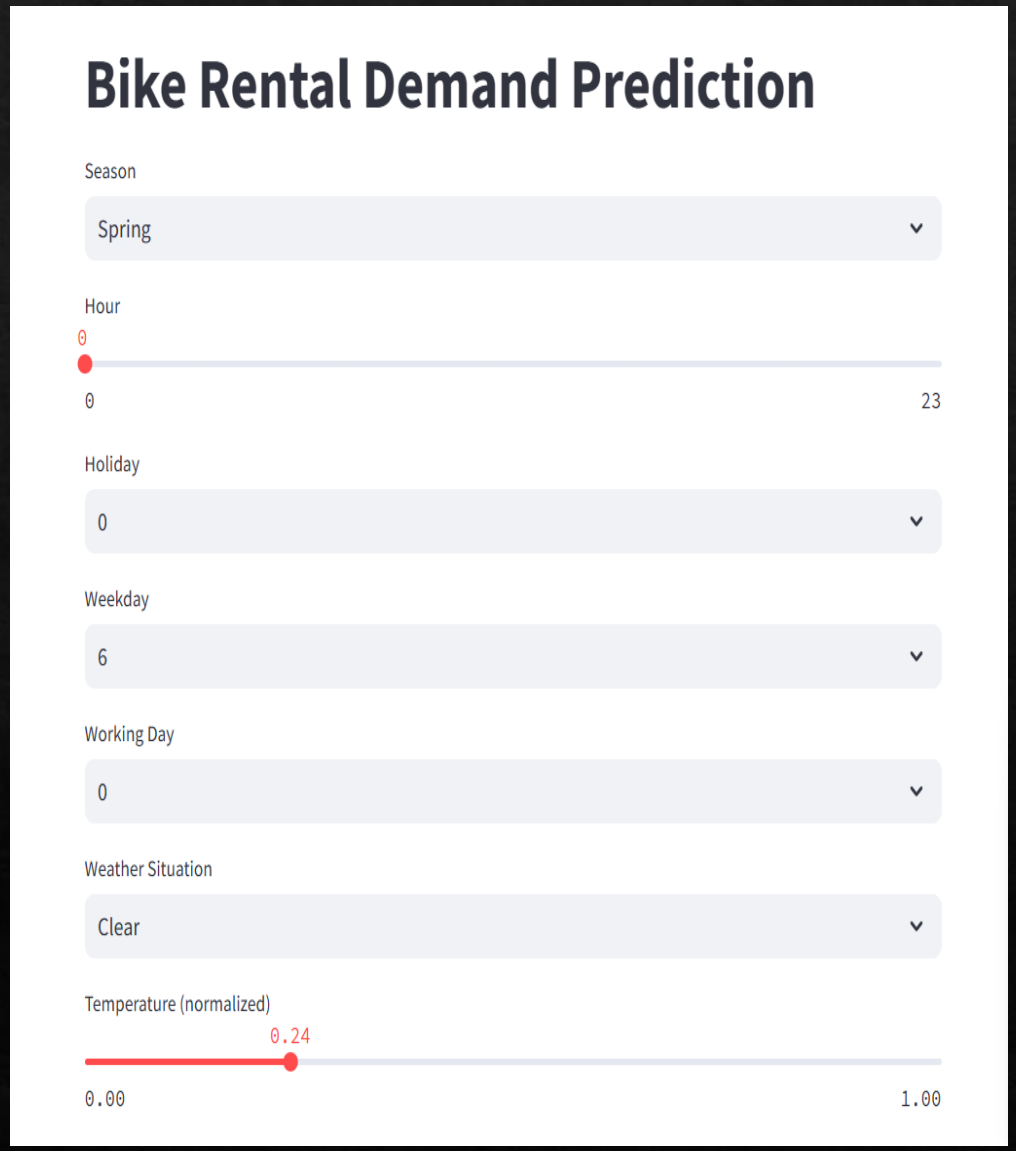
- Best Performance: Random Forest Regressor
  - Lowest RMSE: 34.17
  - Lowest MSE: 1167.91
  - Highest R^2 Score: 0.92

- Worst Performance: AdaBoost Regressor
  - Highest RMSE: 73.97
  - Highest MSE: 5472.22
  - Lowest R^2 Score: 0.63

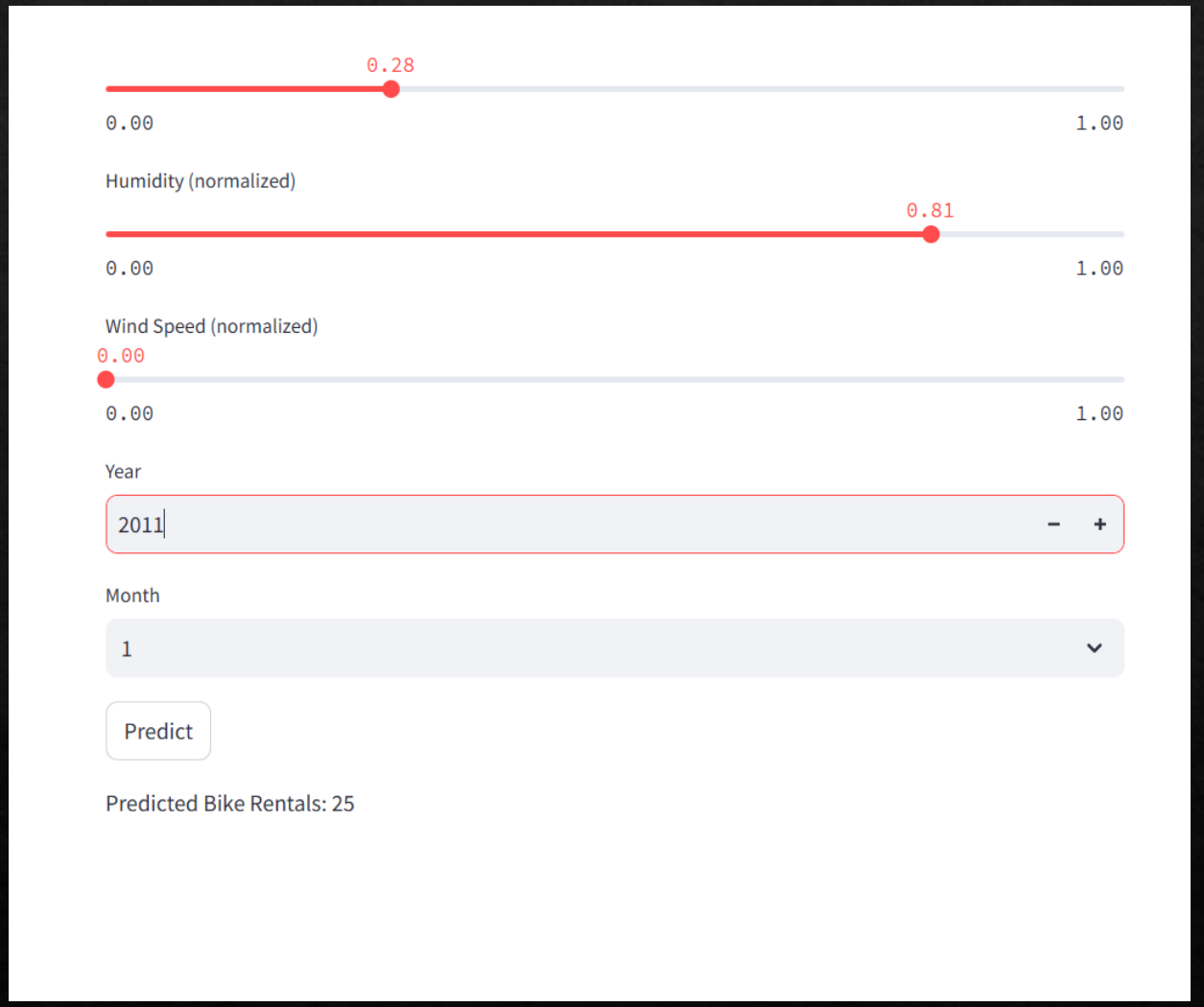
Conclusion: The Random Forest Regressor outperforms other models with the lowest error metrics and highest R^2 score, indicating the best overall fit for the data.

Model Deployment

Model Deployment



Model Deployment



Challenges Faced

- Model Compatibility Issues:
  - Version Mismatch: Encountered issues due to version differences between the model training environment and the deployment environment. For instance, models trained with older versions of libraries (e.g., XGBoost) might not load correctly with newer versions.
  - Serialization Problems: Models saved in one version might not be compatible with other versions, causing difficulties in loading and using these models in different environments.
- Solutions Implemented:
  - Version Management: Ensured consistent library versions across different environments and updated models to ensure compatibility.
  - Model Re-training: Re-trained models using the latest versions of libraries to avoid serialization issues and ensure smooth deployment.
- Lessons Learned:
  - Importance of Compatibility: Ensuring version consistency is crucial for seamless model deployment and integration.
  - Future Proofing: Regularly updating and testing models in the target environment can help mitigate compatibility issues.

Thank you...