Bike Sharing Demand Regression

Muhammad Agisni - DTIDS 0206

Introduction

- Bike-sharing systems are a new generation of traditional bike rentals where the whole process, from membership, rental, and return back, has become automatic.
- Unlike other transport services such as buses or subways, the duration of travel, departure, and arrival position is explicitly recorded in these systems. This feature turns the bike-sharing system into a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that the most important events in the city could be detected by monitoring these data.

FEATURES & TARGET

Short description of features and targets

Features	Description
dteday	Rental date
season	Season (1: Winter, 2: Spring, 3: Summer, 4: Fall)
hr	Hour of day (0 to 23)
holiday	Holiday or not
temp	Normalized temperature in Celsius using MinMaxScaler (min=-8, max=39)
atemp	Normalized feeling temperature in Celsius using MinMaxScaler (min=-16, max=50)
hum	Normalized humidity (divided by 100)
weathersit	Weather situation
	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
casual	Count of bikes rented to casual users
registered	Count of bikes rented to registered users

==========

Target	Description
cnt	Total number of bikes rented, including both casual and registered users

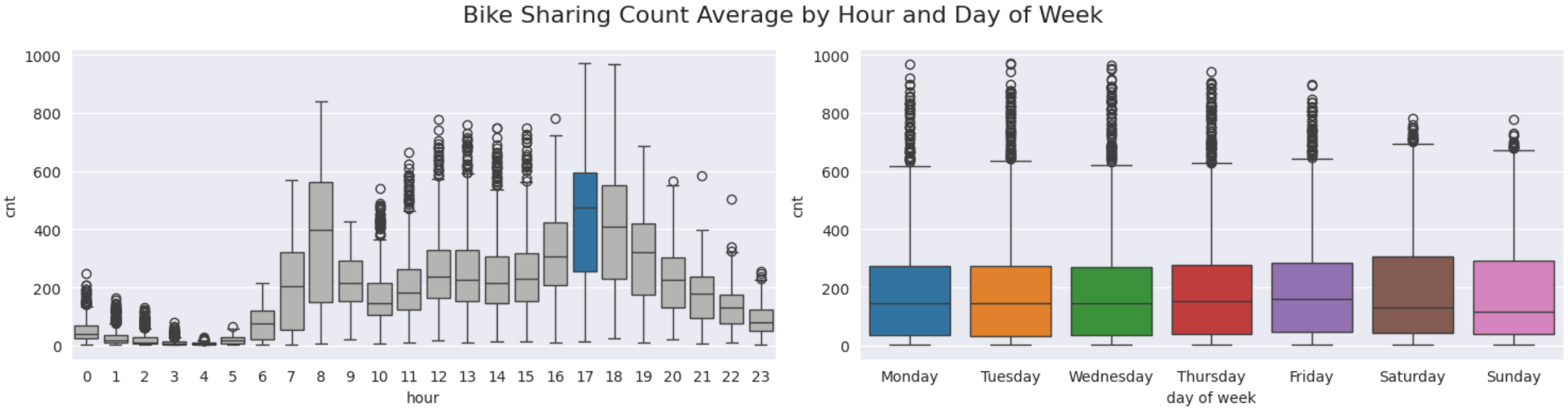
Data Understanding

- The obtained data consists of a two-year historical log corresponding to the years 2011 and 2012 from the Capital Bikeshare system in Washington D.C., USA.
- This data is publicly available at http://capitalbikeshare.com/system-data and includes hourly records with relevant weather and seasonal information, sourced from http://www.freemeteo.com.

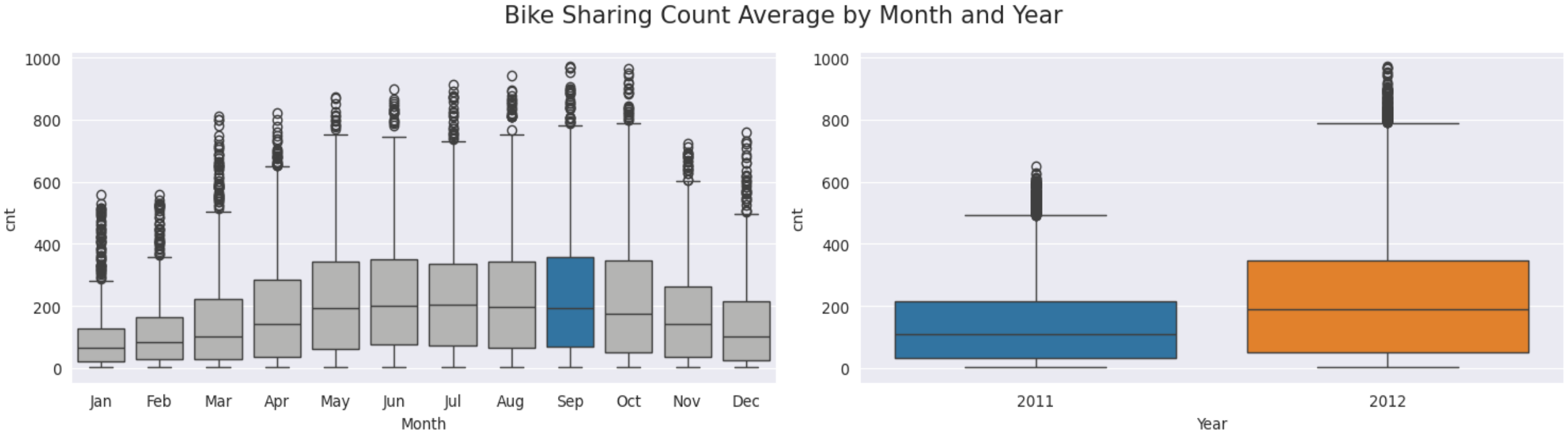
	feature	data_type	null_value(%)	neg_value(%)	0_value(%)	duplicate	n_unique	sample_unique
0	dteday	object	0.0	0.0	0.000	0	731	[2011-12-09, 2012-06-17, 2011-06-15, 2012-03-3...
1	hum	float64	0.0	0.0	0.115	0	89	[0.62, 0.64, 0.53, 0.87, 0.55, 0.72, 0.54, 0.9...
2	weathersit	int64	0.0	0.0	0.000	0	4	[1, 2, 3, 4]
3	holiday	int64	0.0	0.0	97.024	0	2	[0, 1]
4	season	int64	0.0	0.0	0.000	0	4	[4, 2, 3, 1]
5	atemp	float64	0.0	0.0	0.008	0	65	[0.3485, 0.5152, 0.6212, 0.697, 0.4545, 0.6515...
6	temp	float64	0.0	0.0	0.000	0	50	[0.36, 0.54, 0.62, 0.76, 0.46, 0.7, 0.26, 0.82...
7	hr	int64	0.0	0.0	4.349	0	24	[16, 4, 23, 8, 18, 0, 22, 9, 5, 7, 14, 15, 21,...
8	casual	int64	0.0	0.0	9.248	0	305	[24, 2, 17, 19, 99, 6, 20, 13, 219, 1, 11, 9, ...
9	registered	int64	0.0	0.0	0.115	0	742	[226, 16, 90, 126, 758, 39, 196, 27, 5, 315, 2...
10	cnt	int64	0.0	0.0	0.000	0	830	[250, 18, 107, 145, 857, 45, 216, 40, 7, 534, ...

- There are no missing, negative, or duplicate values in the dataset
- Some features have values equal to 0.
- Some features have values that have been normalized.
- The 'dteday' column is of datetime type.
- Except for one column, all other columns are of float or integer type.
- Some categorical fields are represented as integers, such as 'holiday', 'season', 'weathersit'.

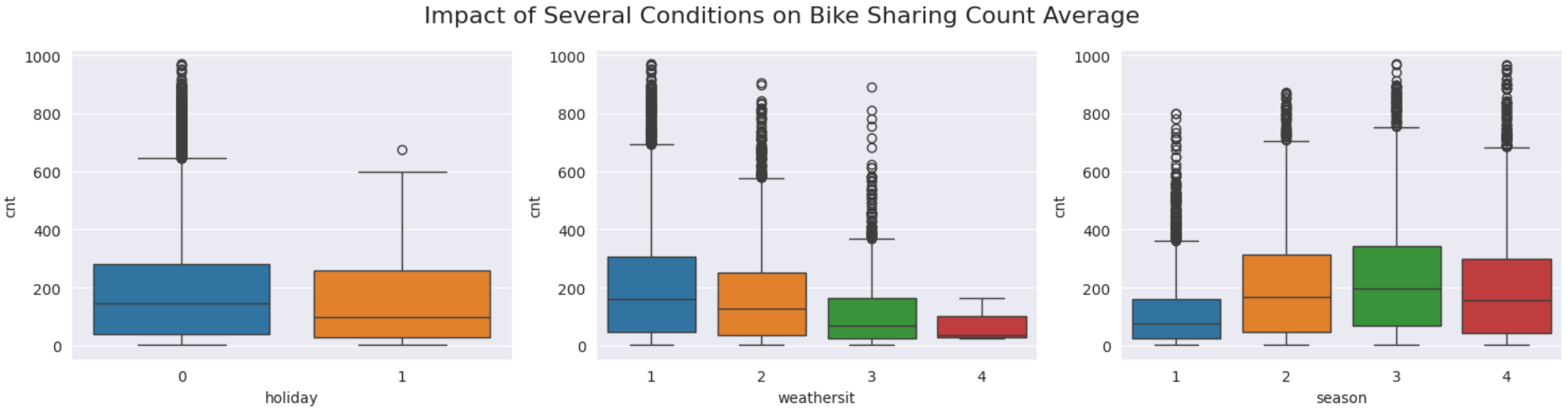
Explorary Data Analysis



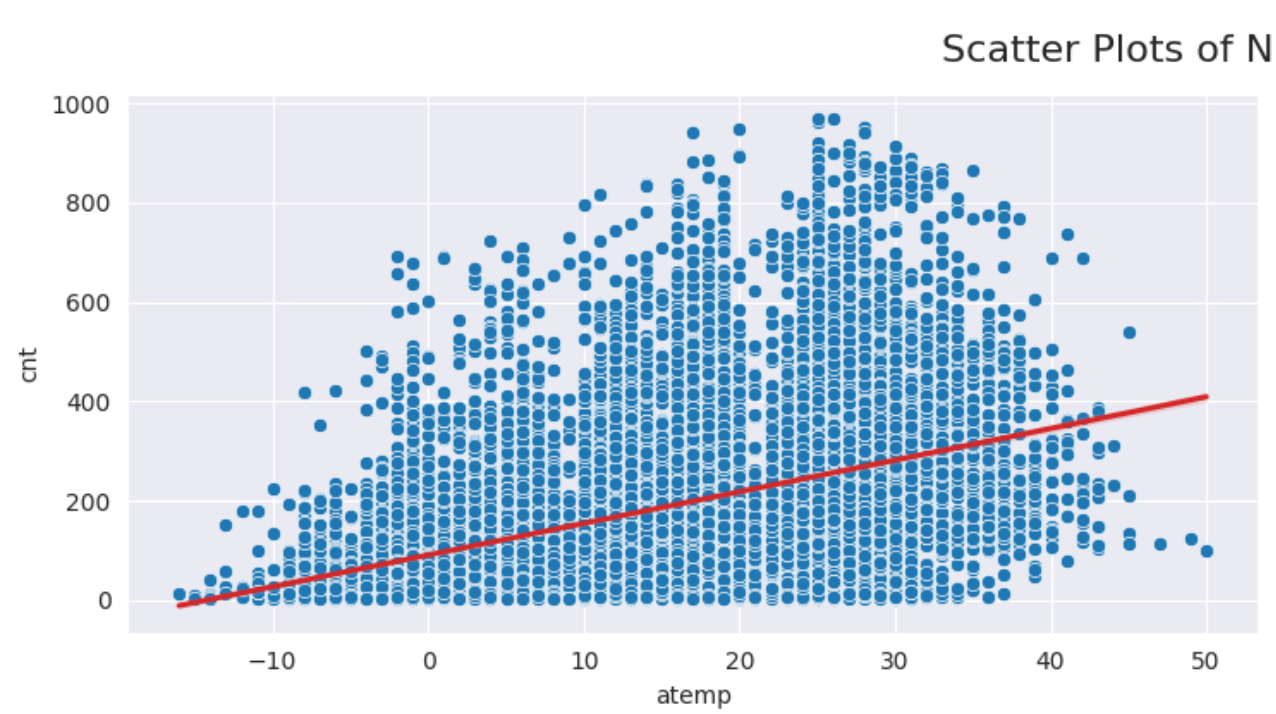
- Bike sharing start increasing from 6 AM and peak at 8 AM, reflecting a considerable number of people using bikes for commuting to work or school.
- The biggest increase occurred at 5 PM, possibly indicating people were returning from work.
- From midnight (0 AM) to early morning (4 AM), bike sharing are low, reflecting minimal activity during these hours.
- Bike sharing median show an increase on weekdays (Monday to Friday) and a decrease on weekends (Saturday and Sunday) but are not significant.
- There is a relatively high consistency in bike sharing on weekdays, indicating a similar bike usage routine for people on workdays.

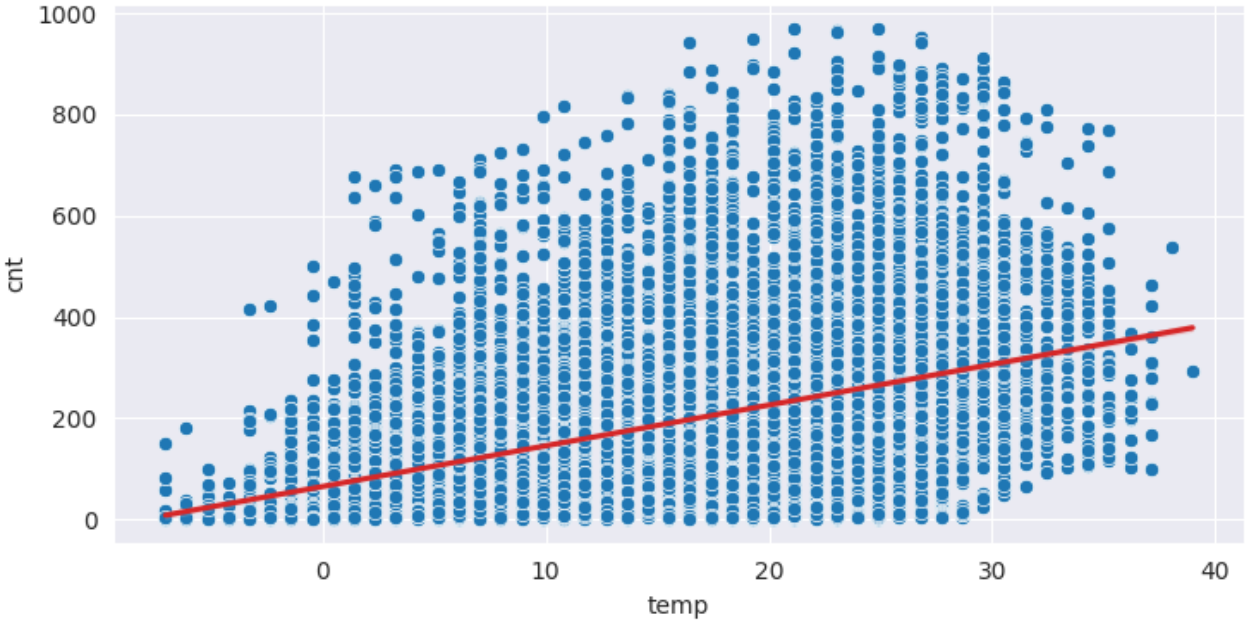


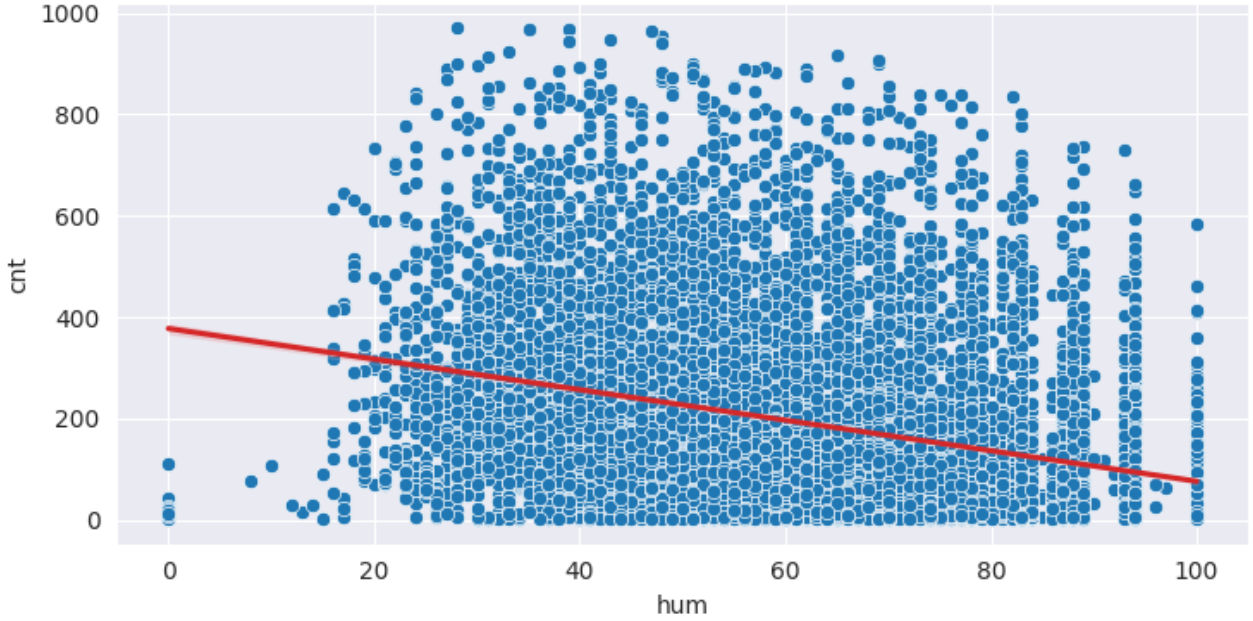
- Median bike sharing count start increasing in March and peak in September, reflecting a increase in bike usage during the summer and late summer seasons.
- Bike sharing significantly decrease during the winter months, especially from December to February, aligning with the expected decline due to colder weather.
- Apart from summer, bike sharing also show an increase during the spring months (March and April) and decrease during the fall months (October and November).
- The bike usage pattern underscores the significance of seasons in influencing people's decisions to bike.
- There is a significance increase in bike sharing from the year 2011 to 2012, indicating growth in the popularity of the bike-sharing system during that period.



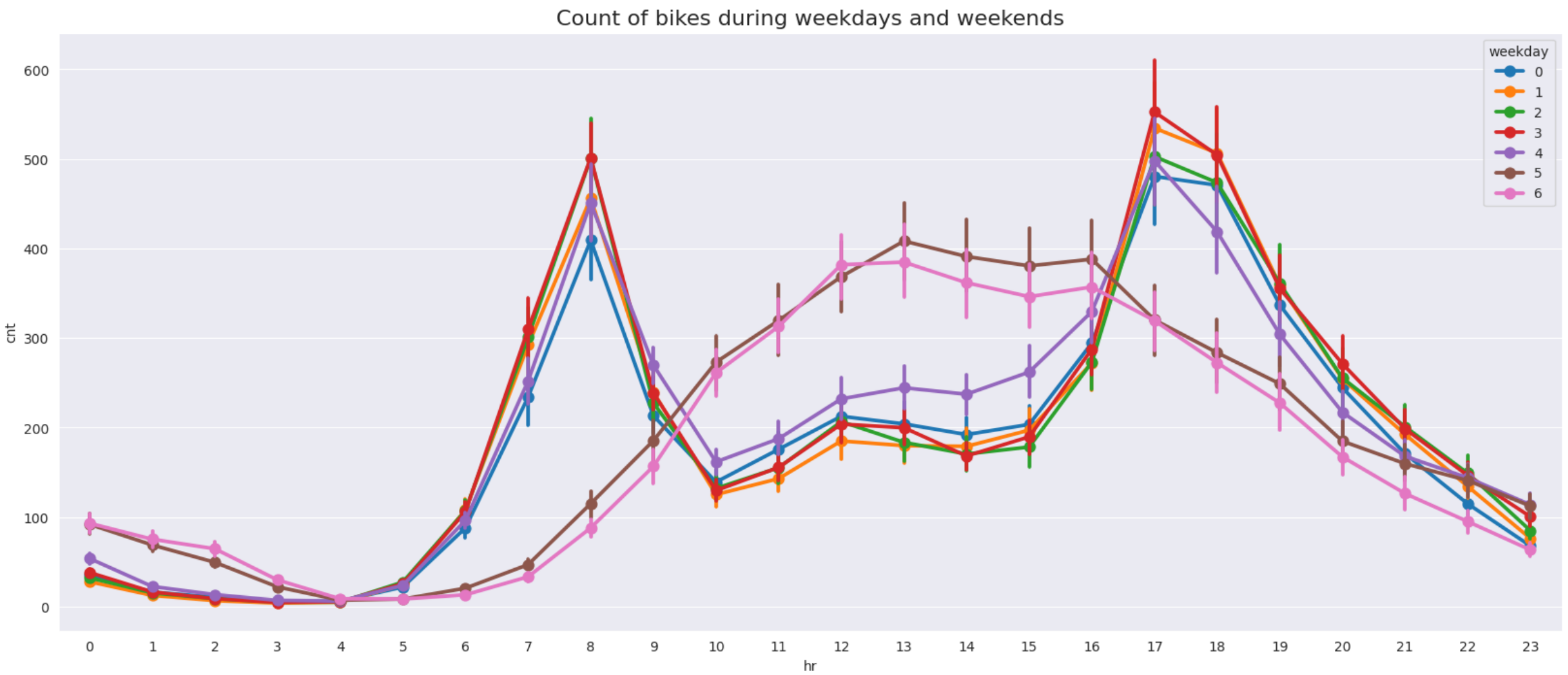
- Weather conditions greatly affect bicycle rental. Better weather conditions (code 1) correspond to the highest number of bicycle rentals.
- Poorer weather conditions, such as mist or cloudy weather (code 2), result in lower bike sharing.
- Extreme weather conditions, like heavy rain, ice pellets, thunderstorms, and thick fog (codes 3 and 4), have a significant impact on reducing bike sharing.
- This indicates that good and clear weather encourages people to use bikes more. This insight underscores the importance of weather analysis in the operational planning of bike-sharing systems.
- The number of bike sharing is lower on holidays (code 1) compared to regular days (code 0).
- Seasonal variations influence bike sharing. Summer (code 3) has the highest bike sharing, followed by fall (code 4), spring (code 2), and winter (code 1) with the lowest bike sharing.



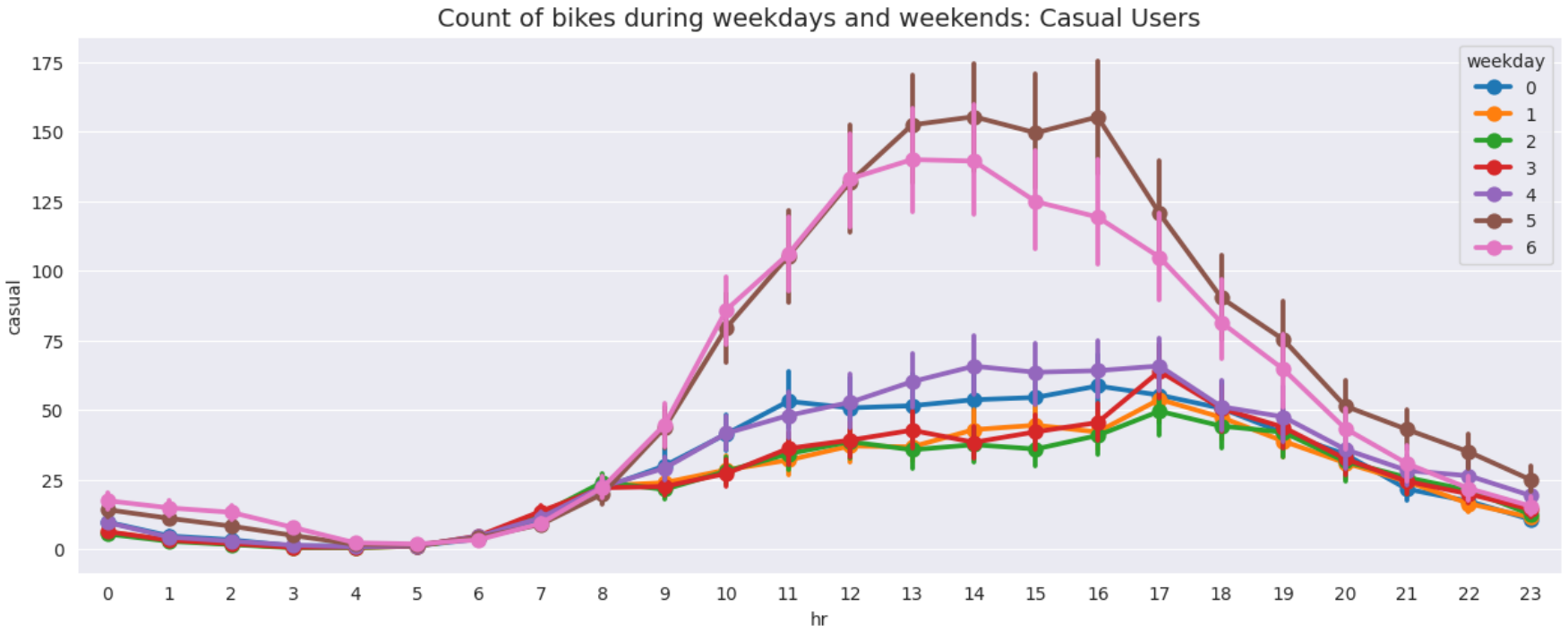


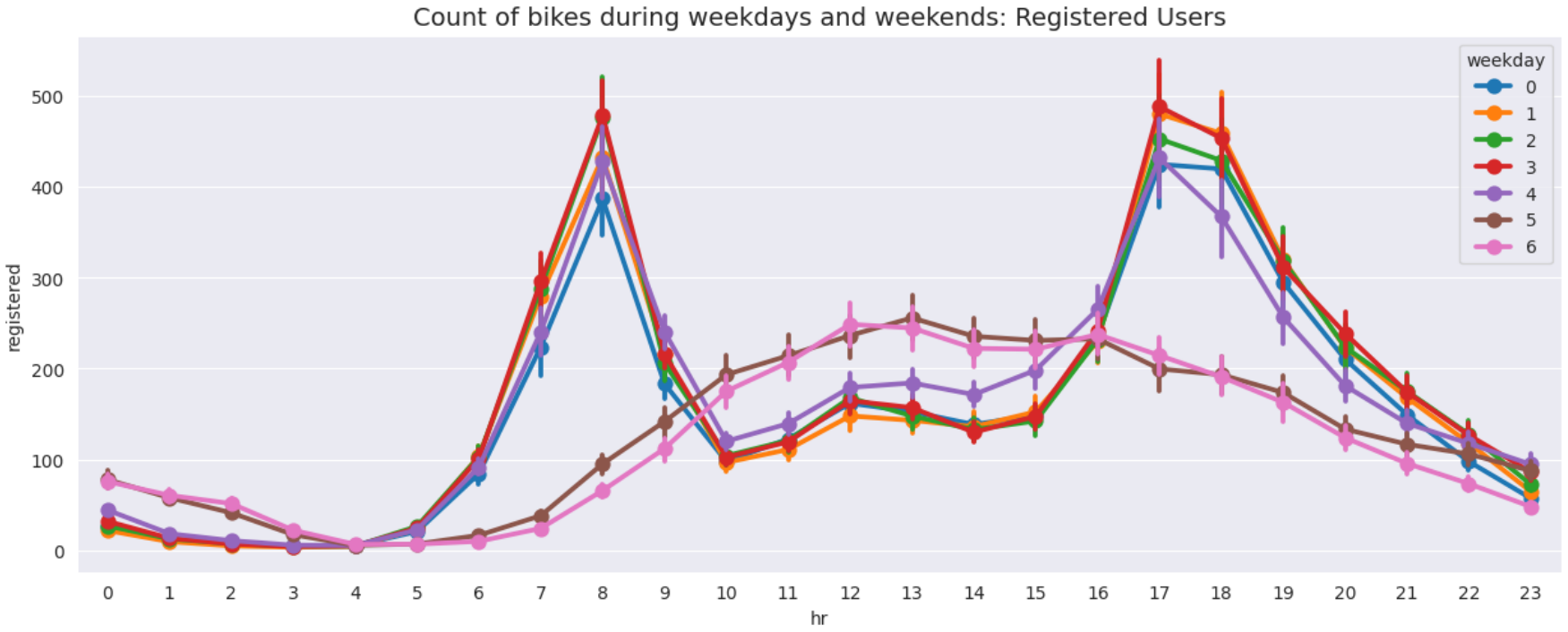


- The relationship 'temp' or 'atemp' features to 'cnt' has a positive linear pattern.
- Value 0 in 'hum' feature looks like an outlier because it is far from dominant.
- There is a negative linear relationship between 'hum' and 'cnt', where when the humidity level increases, the number of bicycle uses tends to decrease.

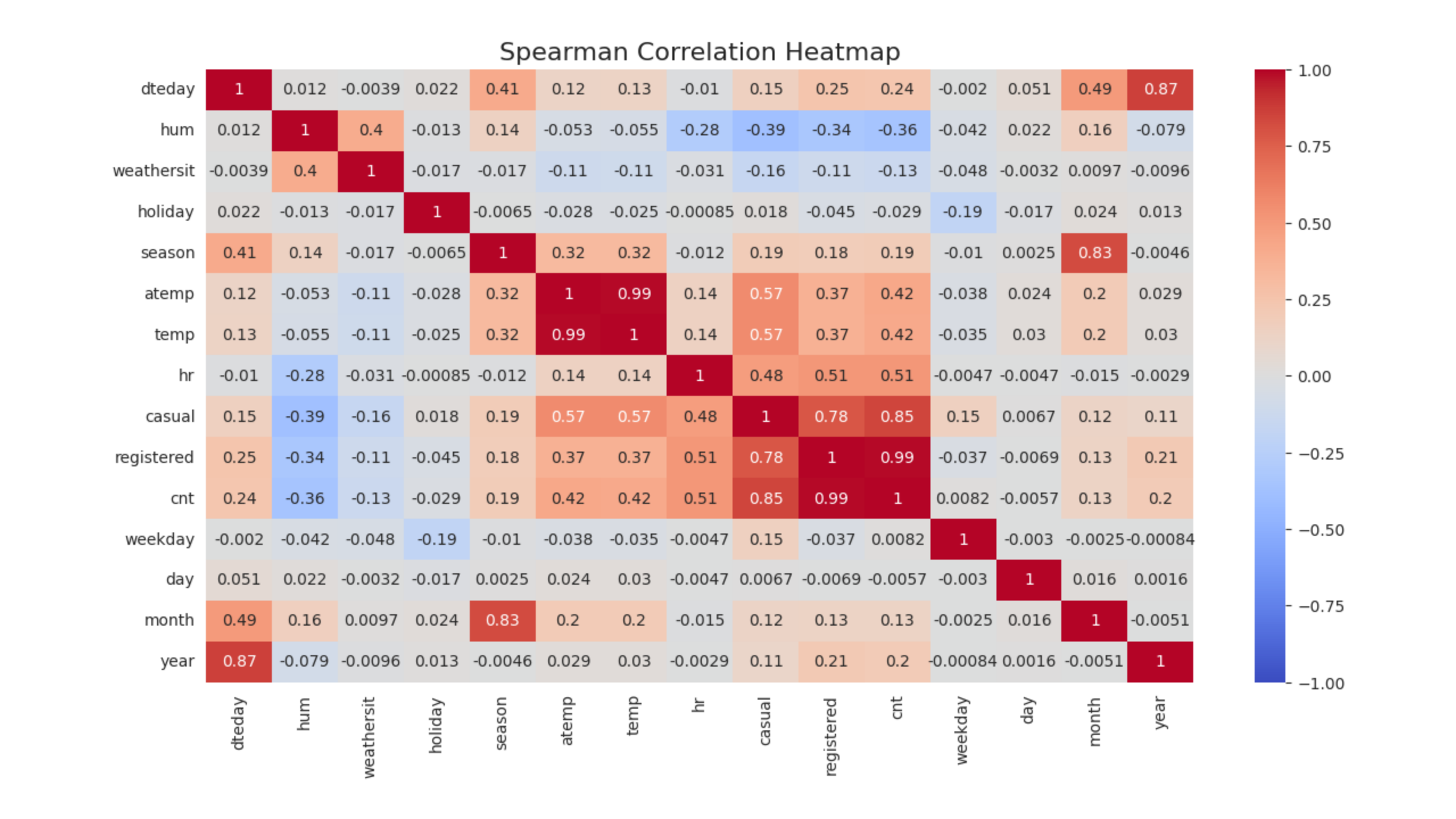


- A significant increase in bike sharing users during the hours of 6 - 8 am only occurs on weekdays Monday - Friday (Code 0 - 4) which indicates that people use bicycles to go to school or work.
- Likewise, at 16 - 18 pm there is a significant increase on weekdays (Monday - Friday) which is the time back from work.
- On weekends Saturday and Sunday (Code 5 and 6) there tends to be an increase during the day until it peaks at 13 noon.





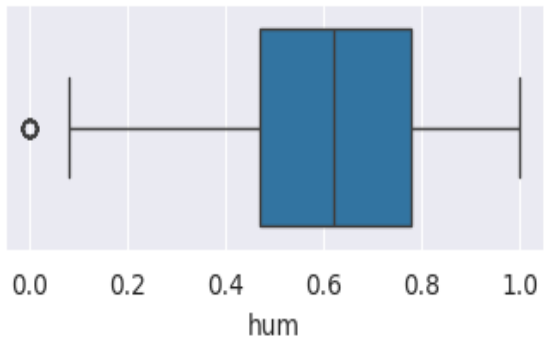
- Casual users use bicycles more on weekends (Saturday and Sunday) in the midday to afternoon.
- Registered users mostly use bicycles on weekdays (Monday - Friday) in the morning and evening, which indicates that users use bicycles for work.

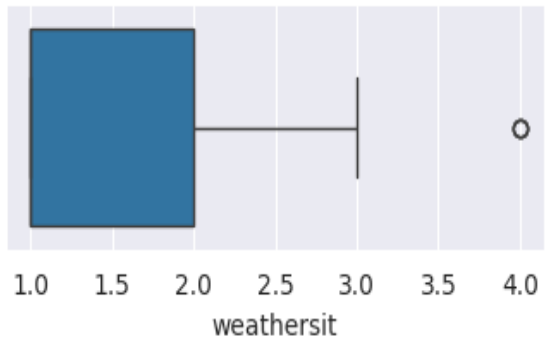

Spearman Correlation Heatmap
dteday 1 0.012 -0.0039 0.022 0.41 0.12 0.13 -0.01 0.15 0.25 0.24 -0.002 0.051 0.49 0.87
hum 0.012 1 0.4 -0.013 0.14 -0.053 -0.055 -0.28 -0.39 -0.34 -0.36 -0.042 0.022 0.16 -0.079
weathersit -0.0039 0.4 1 -0.017 -0.017 -0.11 -0.11 -0.031 -0.16 -0.11 -0.13 -0.048 -0.0032 0.0097 -0.0096
holiday 0.022 -0.013 -0.017 1 -0.0065 -0.028 -0.025 -0.00085 0.018 -0.045 -0.029 -0.19 -0.017 0.024 0.013
season 0.41 0.14 -0.017 -0.0065 1 0.32 0.32 -0.012 0.19 0.18 0.19 -0.01 0.0025 0.83 -0.0046
atemp 0.12 -0.053 -0.11 -0.028 0.32 1 0.99 0.14 0.57 0.37 0.42 -0.038 0.024 0.2 0.029
temp 0.13 -0.055 -0.11 -0.025 0.32 0.99 1 0.14 0.57 0.37 0.42 -0.035 0.03 0.2 0.03
hr -0.01 -0.28 -0.031 -0.00085 -0.012 0.14 0.14 1 0.48 0.51 0.51 -0.0047 -0.0047 -0.015 -0.0029
casual 0.15 -0.39 -0.16 0.018 0.19 0.57 0.57 0.48 1 0.78 0.85 0.15 0.0067 0.12 0.11
registered 0.25 -0.34 -0.11 -0.045 0.18 0.37 0.37 0.51 0.78 1 0.99 -0.037 -0.0069 0.13 0.21
cnt 0.24 -0.36 -0.13 -0.029 0.19 0.42 0.42 0.51 0.85 0.99 1 0.0082 -0.0057 0.13 0.2
weekday -0.002 -0.042 -0.048 -0.19 -0.01 -0.038 -0.035 -0.0047 0.15 -0.037 0.0082 1 -0.003 -0.0025 -0.00084
day 0.051 0.022 -0.0032 -0.017 0.0025 0.024 0.03 -0.0047 0.0067 -0.0069 -0.0057 -0.003 1 0.016 0.0016
month 0.49 0.16 0.0097 0.024 0.83 0.2 0.2 -0.015 0.12 0.13 0.13 -0.0025 0.016 1 -0.0051
year 0.87 -0.079 -0.0096 0.013 -0.0046 0.029 0.03 -0.0029 0.11 0.21 0.2 -0.00084 0.0016 -0.0051 1
dteday hum weathersit holiday season atemp temp hr casual registered cnt weekday day month year
1.00 0.75 0.50 0.25 0.00 −0.25 −0.50 −0.75 −1.00

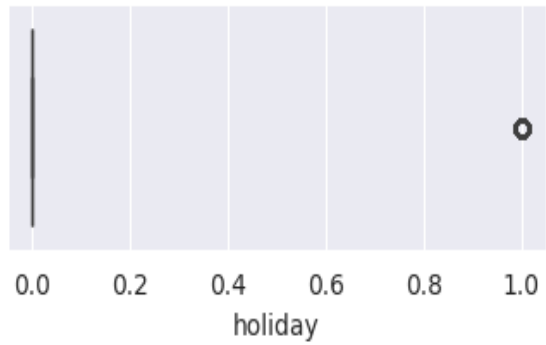
Data Correlation

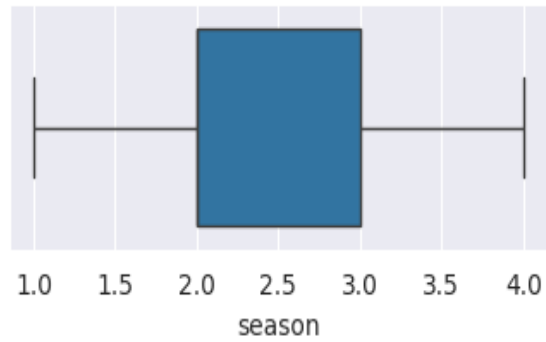
- The `'cnt'` feature is highly correlated with `'casual'` and `'registered'` since it is their sum, so these two features are dropped to avoid redundancy.
- The `'month'` and `'season'` features have a high correlation (0.83) but pass the multicollinearity threshold of 0.9.
- `'temp'` and `'atemp'` show strong multicollinearity (0.9); thus, one is dropped.
- The `'hr'` feature has a moderate positive correlation with `'cnt'`, indicating time-of-day influences bike rentals.
- Both `'temp'` and `'atemp'` moderately correlate with `'cnt'`, showing temperature impacts rentals, while `'hum'` negatively correlates, implying lower humidity favors biking.
- Features like `'year'`, `'season'`, `'month'`, and `'weathersit'` show weaker correlations, but weather conditions negatively impact counts.
- `'holiday'`, `'weekday'`, and `'day'` have minimal correlation, suggesting little influence on bike-sharing trends.

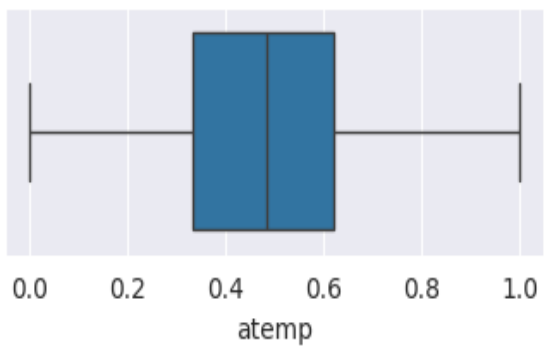
Distribution of Different Features

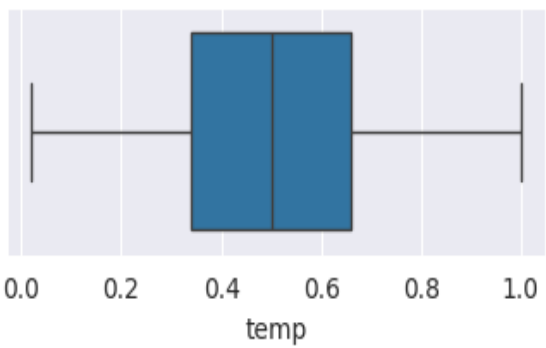


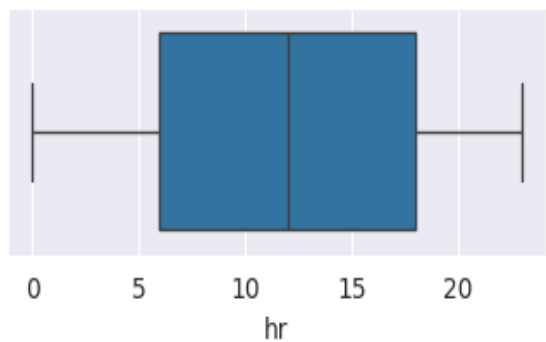


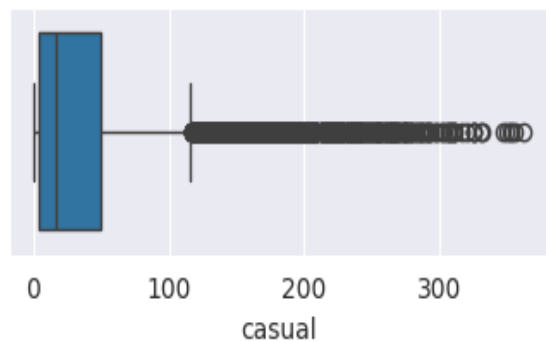


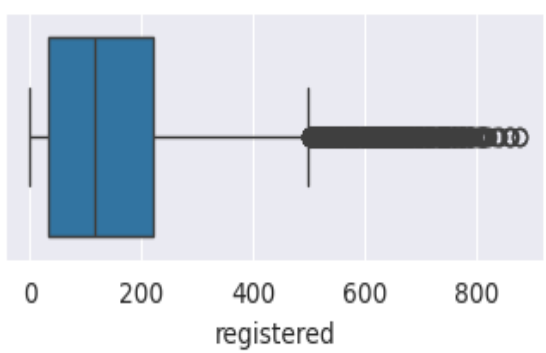


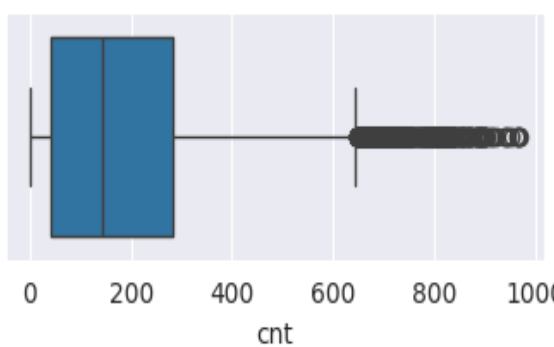


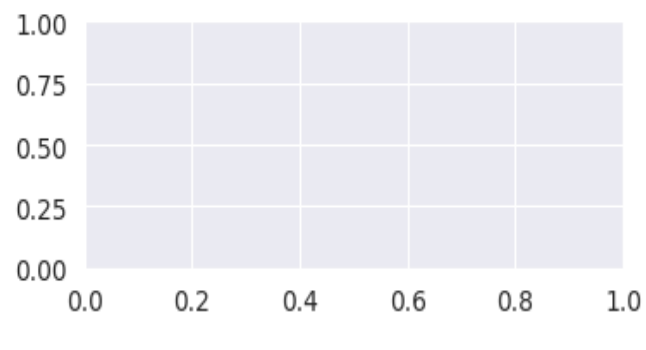


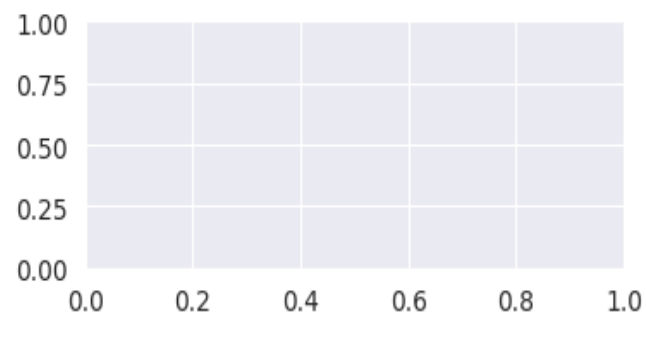












Outliers

	Column	Total Outliers	Percentage (%)	Lower Bound	Upper Bound
0	hum	14	0.12	0.00500	1.24500
1	weathersit	3	0.02	-0.50000	3.50000
2	season	0	0.00	0.50000	4.50000
3	atemp	0	0.00	-0.09855	1.05305
4	temp	0	0.00	-0.14000	1.14000
5	hr	0	0.00	-12.00000	36.00000
6	casual	847	6.96	-63.50000	116.50000
7	registered	470	3.86	-245.00000	499.00000
8	cnt	338	2.78	-323.00000	645.00000

- Based on the boxplots, some features like 'hr', 'season', 'atemp', and 'temp' visually display a normal distribution.
- Apart from that, the majority of features exhibit a positive or right-skewed distribution with some outliers.
- Furthermore, outliers are noticeable in several features, including 'weathersit', 'hum', 'casual', 'registered', and 'cnt'.

Handling Data Problems

Probelm	Feature	Action
Missing Value	None	None
Duplicated Value	None	None
Outlier	'weathersit'	Winsorize with upper and lower bound of 1%
	'hum'	Winsorize with upper and lower bound of 1%
	'cnt'	Logarithmic Function (on Modeling)
Multicollinearity	'atemp'	Drop the column because it can result in multicollinearity
Redudancy	'casual', 'registered'	Drop the column because these features are part of the target variable which can result in redundancy.

Technique	Features	Description
Winsorize	'weathersit', 'hum', 'windspeed'	Winsorize is used to overcome outlier problems. This technique replaces outlier values with previously determined upper bound or lower bound values. In this way, the data distribution becomes more controllable and less influenced by extreme values.
Feature Creation	‘dteday’	Feature creation is the process of creating new features from existing data, aimed at improving the quality of information in the dataset to support better analysis or predictions. Create new features such as day, date, month, year, by extracting from the 'dteday' feature.
Feature Drop	'atemp', 'casual', 'registered’	Feature selection by eliminating features that can cause multicollinearity and redundancy which can negatively affect model performance.
One Hot Encoding	'hr', 'dteday_day', 'dteday_month', 'dteday_year', 'dteday_weekday', 'season', 'weathersit'	One-Hot Encoder is a technique that converts categorical variables into binary vectors, allowing machine learning models to process categorical information effectively. This technique will be used in several distance-based models such as linear regression, ridge, lasso, SVR.

Experiment Setup

Parameter	Description
`data = df_seen`	Specifies the dataset to be used in the experiment, in this case, df_seen.
`target = 'cnt'`	Specifies the target variable, which is 'cnt' (count of bike rentals), to be predicted in the experiment.
`train_size = 0.8`	Determines the size of the training data, set to 80% of the total data, for building machine learning models.
`preprocess = False`	With the use of a custom preprocessing pipeline, this indicates that the prprocessing step will not be executed by Pycaret.
`session_id = RANDOM_SEED`	Sets the session ID to ensure reproducibility of the experiment, using the predefined variable RANDOM_SEED.
`fold = K_FOLDS`	Determines the number of folds for crosvalidation during model training, using the previously defined K_FOLDS.
`n_jobs = 4`	Specifies the number of workers to be used for parallel tasks during the model training process.
`custom_pipeline = base_pipe`	Utilizes the custom pipeline defined earlier (base_pipe) as part of the experiment.

Model Result

Searching for Best Models

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
5	CatBoost Regressor	23.8144	1546.6710	39.2750	0.9533	0.2854	0.2336
4	Extreme Gradient Boosting	26.4366	1962.2962	44.2088	0.9407	0.3103	0.2554
3	Light Gradient Boosting Machine	27.3697	2038.4666	45.0846	0.9385	0.3081	0.2565
1	Random Forest Regressor	28.5361	2262.7593	47.5051	0.9317	0.3356	0.2832
11	Support Vector Regression	33.7176	2931.2619	54.0783	0.9116	0.3745	0.3451
0	Decision Tree Regressor	38.3498	4277.4182	65.3516	0.8709	0.4743	0.4032
2	Decision Tree Regressor	46.0920	6233.3046	78.6746	0.8118	0.5222	0.4689
6	K Neighbors Regressor	56.1415	8173.2956	90.3701	0.7534	0.5248	0.5254
8	Linear Regression	63.7597	9546.9331	97.6503	0.7117	0.5892	0.6045
7	Bayesian Ridge	63.7944	9563.4448	97.7355	0.7112	0.5892	0.6053
9	Ridge Regression	63.8278	9577.6558	97.8088	0.7108	0.5893	0.6061
10	Lasso Regression	143.5248	42067.8161	205.0827	-0.2694	1.4247	3.9394

- Based on the benchmark results using transformed targets:
- - CatBoost Regressor achieves the best performance with the lowest MAE (23.81), MAPE (23.36%), and a high R^2 (0.95). This indicates its exceptional ability to handle transformed targets effectively, providing robust predictions with a training time of 2.51 seconds.
- - Extreme Gradient Boosting (XGBoost) also demonstrates strong performance with an MAE (26.44) and MAPE (25.54%) close to CatBoost. Its R^2 (0.94) indicates a high level of variance explanation, and its training time (0.21 seconds) makes it an efficient alternative to CatBoost for similar tasks.
- - Light Gradient Boosting Machine (LightGBM) follows closely, with an MAE (27.37) and MAPE (25.65%), coupled with a high R^2 (0.94). Its extremely fast training time (0.20 seconds) highlights its suitability for quick model training while maintaining competitive performance.

Searching for Best Models

- Based on the analysis and benchmarking results, regression models, particularly non-linear and gradient-based models, perform well on the bike-sharing dataset, effectively handling the complexity of predicting bike rental demand based on weather, time, and other contextual variables.
- Strong predictive power: Models like CatBoost, XGBoost, and LightGBM achieve high R^2 value (above 0.93) with low errors (MAE and MAPE), demonstrating their ability to explain variability in bike rental demand and make precise predictions.
- Gradient-boosted models excel: CatBoost leads in predictive accuracy, making it the most robust model for this problem. XGBoost and LightGBM closely follow, offering a strong balance between accuracy and training efficiency.
- Efficiency in model training: LightGBM is exceptionally fast during training (0.20 seconds) while maintaining competitive performance, making it ideal for real-time applications or iterative model tuning.
- Linear models lag behind: Models such as Linear Regression, Bayesian Ridge, and Ridge Regression struggle to capture the dataset's non-linear relationships, evidenced by relatively lower R^2 values (~0.71) and higher errors.
- Suboptimal models: Simpler models like Decision Tree and K-Nearest Neighbors perform moderately, while Lasso Regression shows the weakest results, indicating it is not suitable for this dataset.

Hyper parameter

Catboost Regression

Hyperparameter setup

```
hyperparam_catboost = {
    'regressor__depth': [7, 8, 9],
    'regressor__learning_rate': [0.06, 0.07, 0.08],
    'regressor__iterations': [1400, 1500, 1600]
}
```

Best Hyperparameter

```
{'iterations': 1400,
 'learning_rate': 0.06,
 'depth': 8,
 'loss_function': 'RMSE',
 'verbose': False}
```

Before Tuning:

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	24.9458	1671.4200	40.8830	0.9485	0.2851	0.2375
1	23.9340	1726.6373	41.5528	0.9492	0.2778	0.2256
2	22.8618	1315.3803	36.2682	0.9598	0.3051	0.2572
3	24.0120	1613.2702	40.1655	0.9504	0.2880	0.2317
4	23.3183	1406.6471	37.5053	0.9587	0.2710	0.2158
Mean	23.8144	1546.6710	39.2750	0.9533	0.2854	0.2336
Std	0.7053	158.4220	2.0367	0.0049	0.0115	0.0138

After Tuning:

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	23.6986	1554.3005	39.4246	0.9521	0.2822	0.2332
1	22.4356	1601.8629	40.0233	0.9529	0.2798	0.2217
2	21.0175	1148.0853	33.8834	0.9649	0.3017	0.2538
3	22.6629	1520.7538	38.9968	0.9533	0.2838	0.2253
4	22.1345	1285.8152	35.8583	0.9622	0.2632	0.2092
Mean	22.3898	1422.1635	37.6373	0.9571	0.2821	0.2287
Std	0.8646	175.0325	2.3661	0.0054	0.0123	0.0148

Catboost Regression

- After tuning, all evaluation metrics show improvement. MAE, RMSE, MSE, RMSLE, and MAPE decrease, indicating enhanced accuracy and better predictive performance.
- Mean MAE improves by approximately 1.42 (23.81 to 22.39), showing better precision in predictions.
- The R^2 value increases slightly from 0.9533 to 0.9571, highlighting improved variance explanation and generalization.
- Variability (std) in metrics remains consistent, reflecting stable performance across folds.

XGboost Regression

Hyperparameter setup

```
xgboost_tuned_params = {'learning_rate': 0.05,
                        'max_depth': 5,
                        'min_child_weight': 0,
                        'n_estimators': 1300,
                        'subsample': 0.9}
```

Best Hyperparameter

```
{'objective': 'reg:squarederror',
 'base_score': None,
 'booster': None,
 'callbacks': None,
 'colsample_bylevel': None,
 'colsample_bynode': None,
 'colsample_bytree': None,
 'device': None,
 'early_stopping_rounds': None,
 'enable_categorical': False,
 'eval_metric': None,
 'feature_types': None,
 'gamma': None,
 'grow_policy': None,
 'importance_type': None,
 'interaction_constraints': None,
 'learning_rate': 0.05,
 'max_bin': None,
 'max_cat_threshold': None,
 'max_cat_to_onehot': None,
 'max_delta_step': None,
 'max_depth': 5,
 'max_leaves': None,
 'min_child_weight': 2,
 'missing': nan,
 'monotone_constraints': None,
 'multi_strategy': None,
 'n_estimators': 1300,
 'n_jobs': None,
 'num_parallel_tree': None,
 'random_state': None,
 'reg_alpha': None,
 'reg_lambda': None,
 'sampling_method': None,
 'scale_pos_weight': None,
 'subsample': 0.9,
 'tree_method': None,
 'validate_parameters': None,
 'verbosity': None}
```

Before Tuning:

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	28.2005	2295.8252	47.9148	0.9292	0.3145	0.2635
1	26.6529	2192.6675	46.8259	0.9355	0.3121	0.2491
2	25.9210	1827.1891	42.7456	0.9441	0.3308	0.2861
3	26.3550	1877.9296	43.3351	0.9423	0.3129	0.2539
4	25.0536	1617.8695	40.2227	0.9525	0.2811	0.2244
Mean	26.4366	1962.2962	44.2088	0.9407	0.3103	0.2554
Std	1.0338	248.3199	2.8063	0.0079	0.0161	0.0200

After Tuning:

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	25.5007	1861.1484	43.1410	0.9426	0.2915	0.2402
1	23.9707	1775.0428	42.1313	0.9478	0.2892	0.2306
2	22.3252	1272.0266	35.6655	0.9611	0.3131	0.2628
3	23.7168	1592.9893	39.9123	0.9510	0.2964	0.2317
4	23.7920	1493.4649	38.6454	0.9561	0.2727	0.2177
Mean	23.8611	1598.9344	39.8991	0.9517	0.2926	0.2366
Std	1.0085	208.7095	2.6453	0.0064	0.0130	0.0150

XGboost Regression

- After tuning, there is a noticeable improvement in the model's performance across all metrics. The mean MAE decreases from 26.44 to 23.86, indicating improved prediction accuracy. Similarly, MSE and RMSE values decrease, suggesting better error minimization and a reduction in prediction discrepancies.
- R^2 improves slightly from 0.9407 to 0.9517, indicating a better fit to the data and a higher proportion of variance explained by the model.
- MAPE decreases from 0.2554 to 0.2366, showing enhanced model efficiency in predicting relative errors.
- RMSLE remains relatively stable, but the reduction in standard deviation for both MAE and RMSE reflects increased model stability after tuning.
- Overall, the tuning process significantly enhanced the model's generalization and stability, achieving better accuracy while maintaining reliability across folds.

LightGBM Regression

Hyperparameter setup

```
hyperparam_lgbm = {
    'regressor__n_estimators': [100, 200, 500],
    'regressor__learning_rate': [0.01, 0.05, 0.1],
    'regressor__max_depth': [-1, 5, 10],
    'regressor__num_leaves': [31, 50, 100],
    'regressor__min_child_samples': [10, 20, 30]
}
```

Best Hyperparameter

```
{'check_inverse': True,
 'func': <ufunc 'log1p'>,
 'inverse_func': <ufunc 'expm1'>,
 'regressor__boosting_type': 'gbdt',
 'regressor__class_weight': None,
 'regressor__colsample_bytree': 1.0,
 'regressor__importance_type': 'split',
 'regressor__learning_rate': 0.05,
 'regressor__max_depth': -1,
 'regressor__min_child_samples': 20,
 'regressor__min_child_weight': 0.001,
 'regressor__min_split_gain': 0.0,
 'regressor__n_estimators': 500,
 'regressor__n_jobs': None,
 'regressor__num_leaves': 50,
 'regressor__objective': None,
 'regressor__random_state': None,
 'regressor__reg_alpha': 0.0,
 'regressor__reg_lambda': 0.0,
 'regressor__subsample': 1.0,
 'regressor__subsample_for_bin': 200000,
 'regressor__subsample_freq': 0,
 'regressor': LGBMRegressor(learning_rate=0.05, n_estimators=500, num_leaves=50),
 'transformer': None}
```

Before Tuning:

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	28.7259	2273.9042	47.6855	0.9299	0.3062	0.2607
1	28.1195	2276.3615	47.7112	0.9330	0.3004	0.2452
2	25.6246	1726.7759	41.5545	0.9472	0.3323	0.2806
3	27.7131	2038.0536	45.1448	0.9374	0.3075	0.2588
4	26.6654	1877.2379	43.3271	0.9449	0.2942	0.2372
Mean	27.3697	2038.4666	45.0846	0.9385	0.3081	0.2565
Std	1.1010	216.8736	2.4173	0.0067	0.0130	0.0149

After Tuning:

Fold	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	26.5288	1781.3577	42.2061	0.9451	0.3887	0.3857
1	24.9130	1643.9947	40.5462	0.9516	0.3896	0.3770
2	24.3729	1575.2366	39.6893	0.9518	0.4294	0.4223
3	24.2648	1597.1256	39.9641	0.9509	0.4021	0.3735
4	24.5411	1656.7643	40.7034	0.9514	0.3815	0.3478
Mean	24.9241	1650.8958	40.6218	0.9502	0.3983	0.3812
Std	0.8319	71.7188	0.8746	0.0026	0.0169	0.0241

LightGBM Regression

- The mean MAE decreases from 27.37 to 24.92, indicating a significant enhancement in prediction accuracy.
- MSE and RMSE also show considerable reductions, from 2038.47 to 1681.87 and from 45.08 to 40.95, respectively, suggesting better error minimization and a decrease in large prediction discrepancies.
- R^2 improves slightly from 0.9385 to 0.9502, reflecting a better fit to the data and an increase in the proportion of variance explained by the model.
- MAPE decreases from 0.2565 to 0.3812, demonstrating enhanced model efficiency in predicting relative errors with greater consistency.
- RMSLE remains stable, but the reduced standard deviation in both MAE and RMSE indicates increased stability and reliability across folds.
- Overall, the tuning process has significantly enhanced the model's accuracy and generalization ability, improving its performance while ensuring stability across different data subsets.

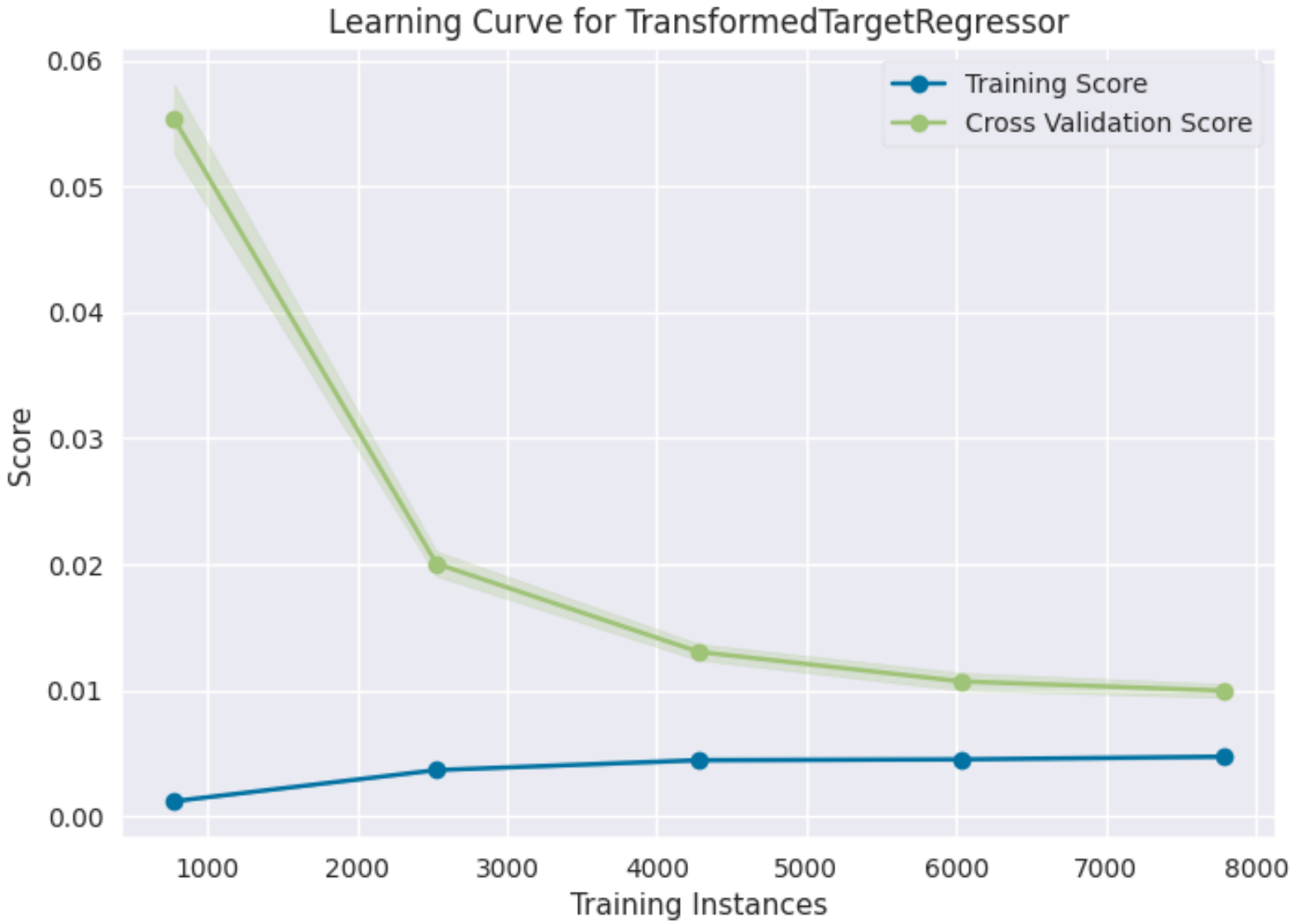
Final Result

Summary

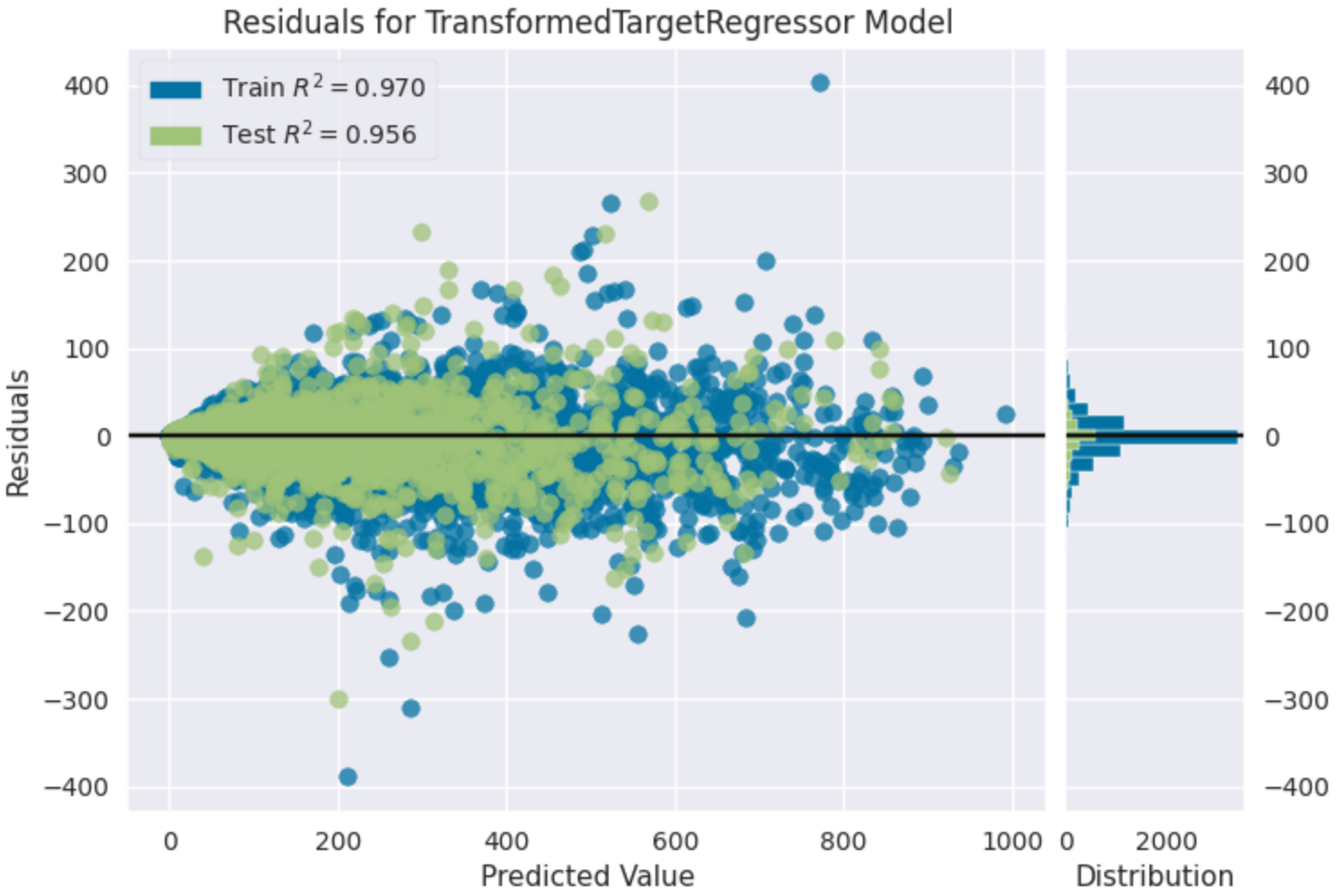
Model	MAE	MAPE	R2	TT (Sec)
CatBoost Regressor	22.89 (+0.00)	22.81% (+0.00)	0.95 (+0.00)	0.73 (+0.00)
Extreme Gradient Boosting	23.86 (+1.52)	23.63% (+0.82)	0.95 (+0.00)	0.19 (-0.54)
Light BGM	24.93 (+5.68)	38.15% (+4.52)	0.95 (-0.03)	1.02 (+5.27)

- Based on this analysis, CatBoost Regressor may be the best choice because it provides a good balance between high accuracy and reasonable training time. However, if training speed is a priority, Extreme Gradient Boosting can also be considered a good alternative.
- Considering business problems, CatBoost is the most appropriate choice because it predicts demand more accurately, thus avoiding losses and potential profit loss.

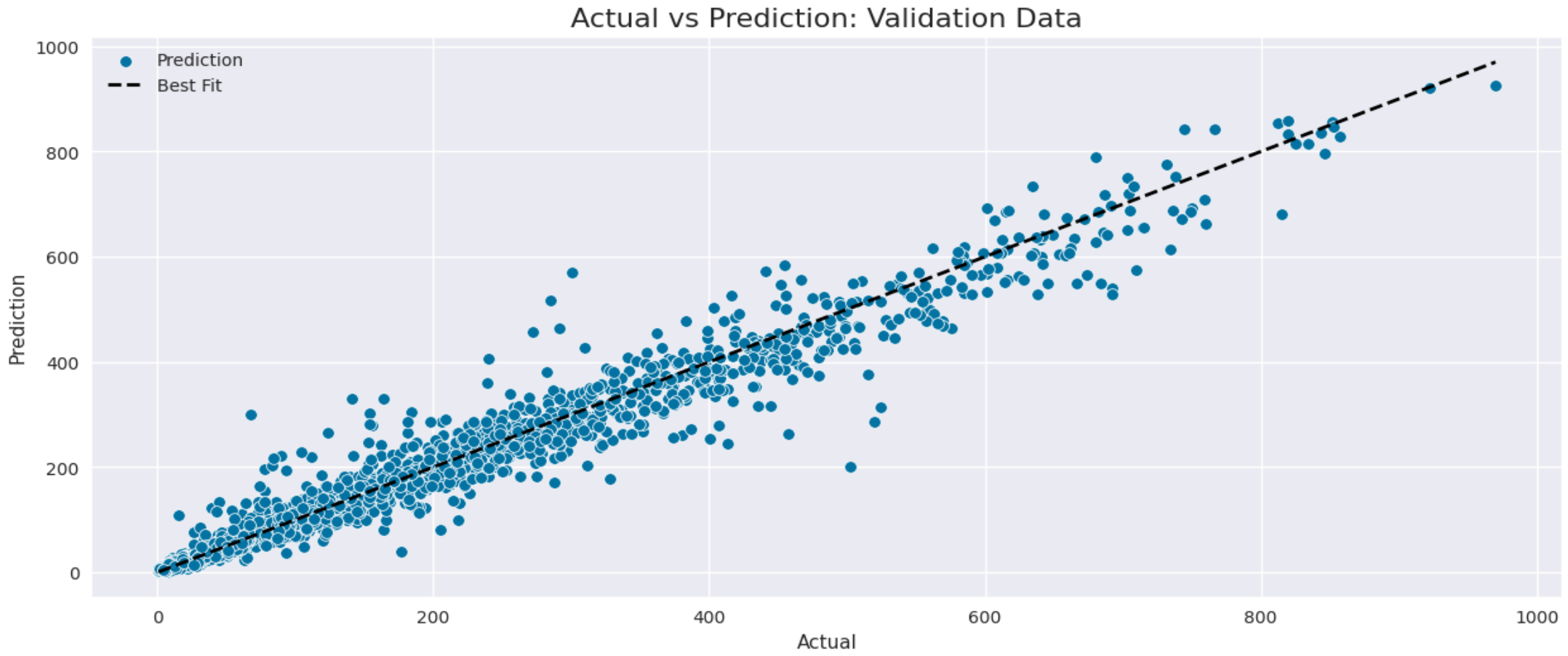
Evaluate Model



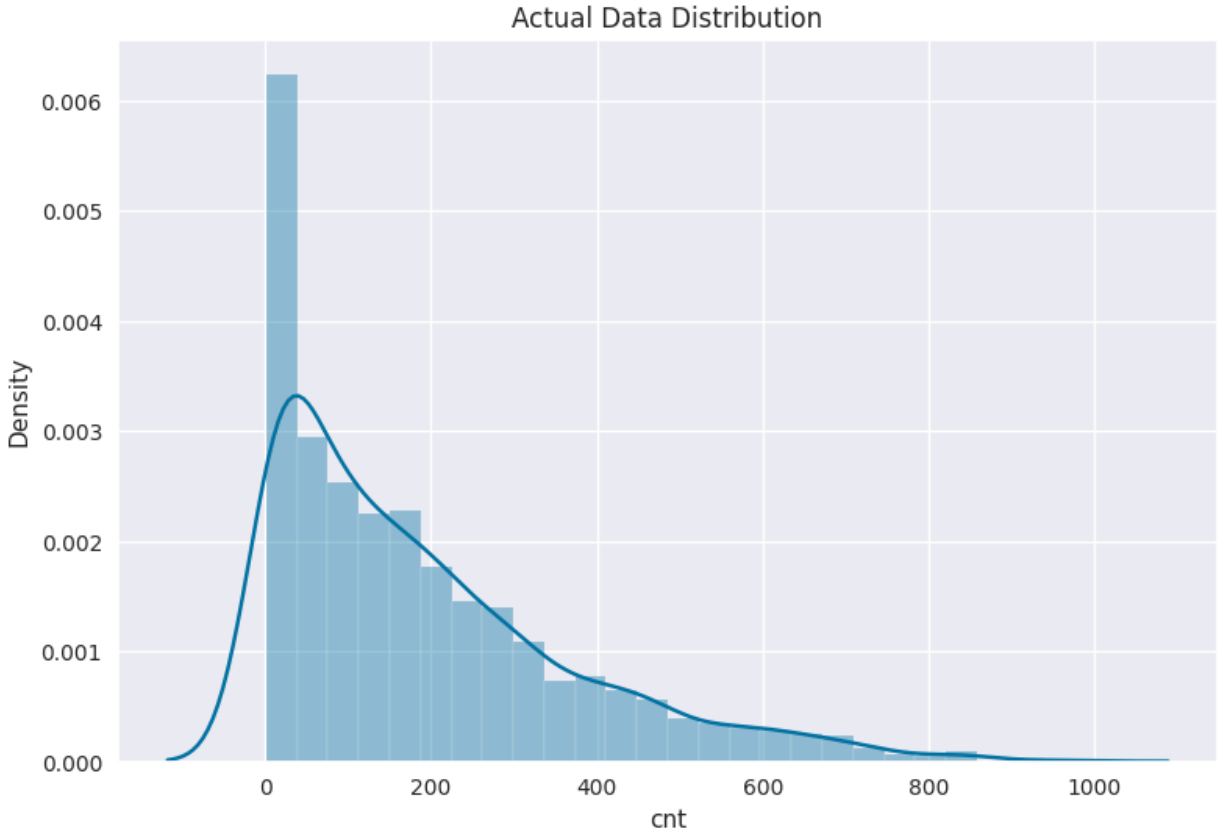
- The learning curve for the TransformedTargetRegressor shows steady improvement in cross-validation scores with more training data, indicating the model benefits from additional data.
- The convergence of validation and training scores suggests good generalization without underfitting or overfitting.
- After ~6000 instances, cross-validation improvement plateaus, implying limited gains from more data.
- Overall, the model demonstrates robustness and is well-suited for the current dataset size.

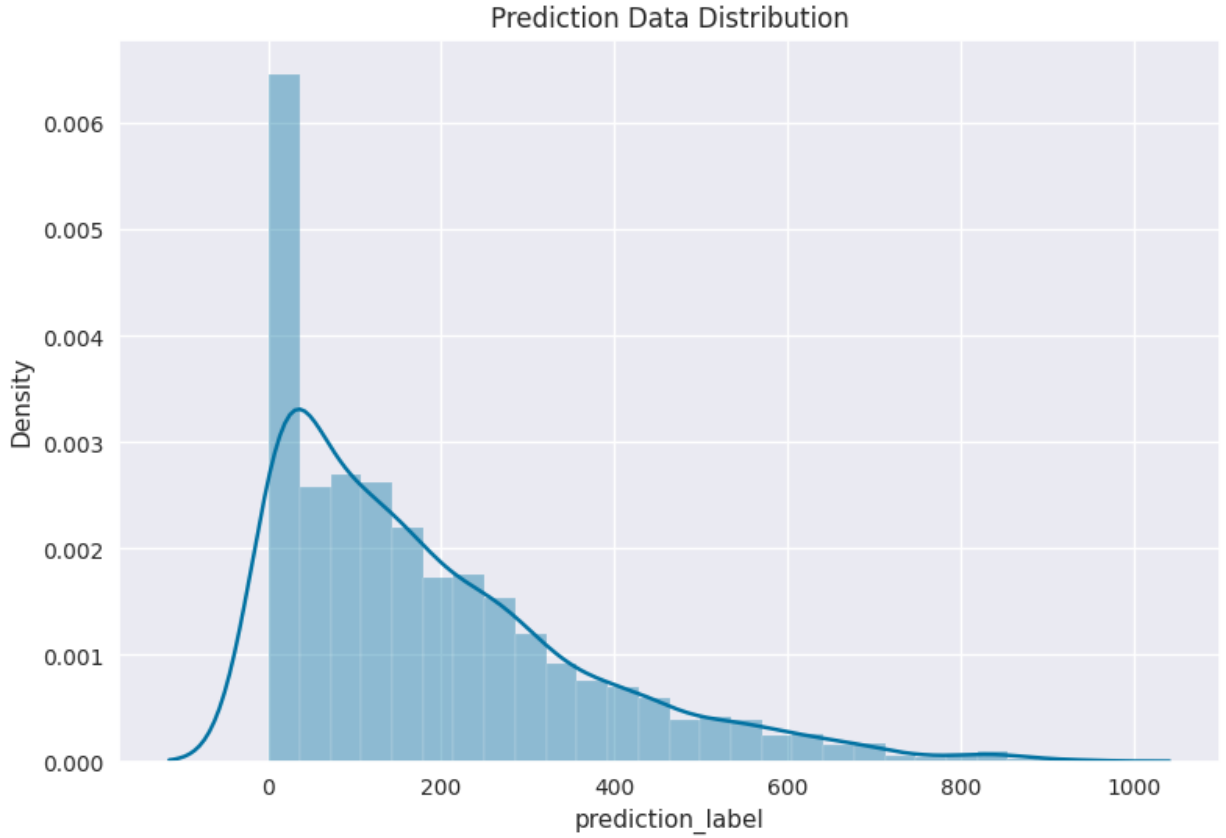


- The residuals near the zero line indicate that the TransformedTargetRegressor model has accurate predictions with low error.
- High R^2 values (0.970 for training and 0.956 for test) show strong predictive ability.
- The residual distribution is centered around zero with no clear patterns, suggesting random errors.
- The histogram confirms symmetric residuals, indicating minimal bias.
- Overall, the model is accurate, with well-behaved residuals and good generalization to unseen data.

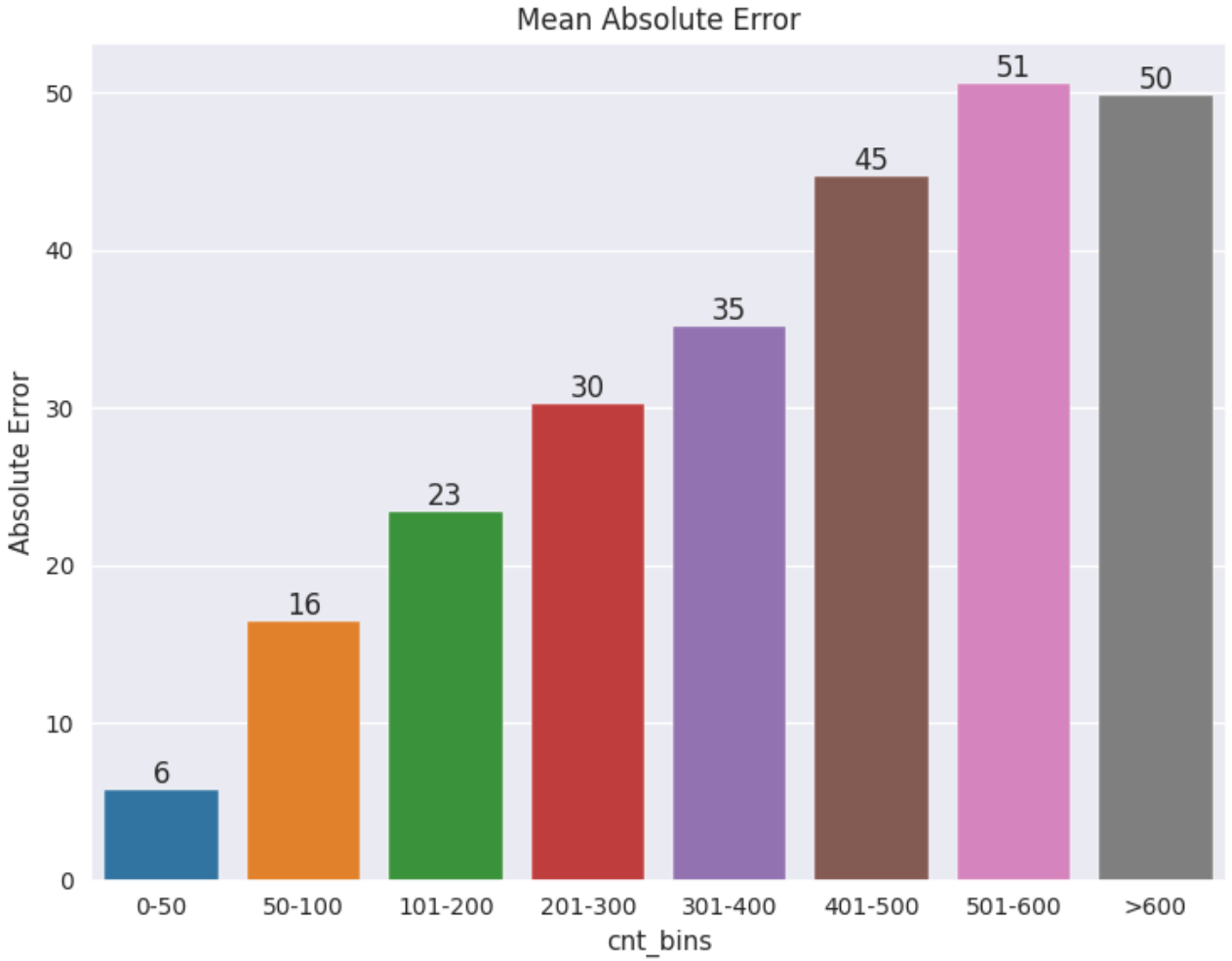


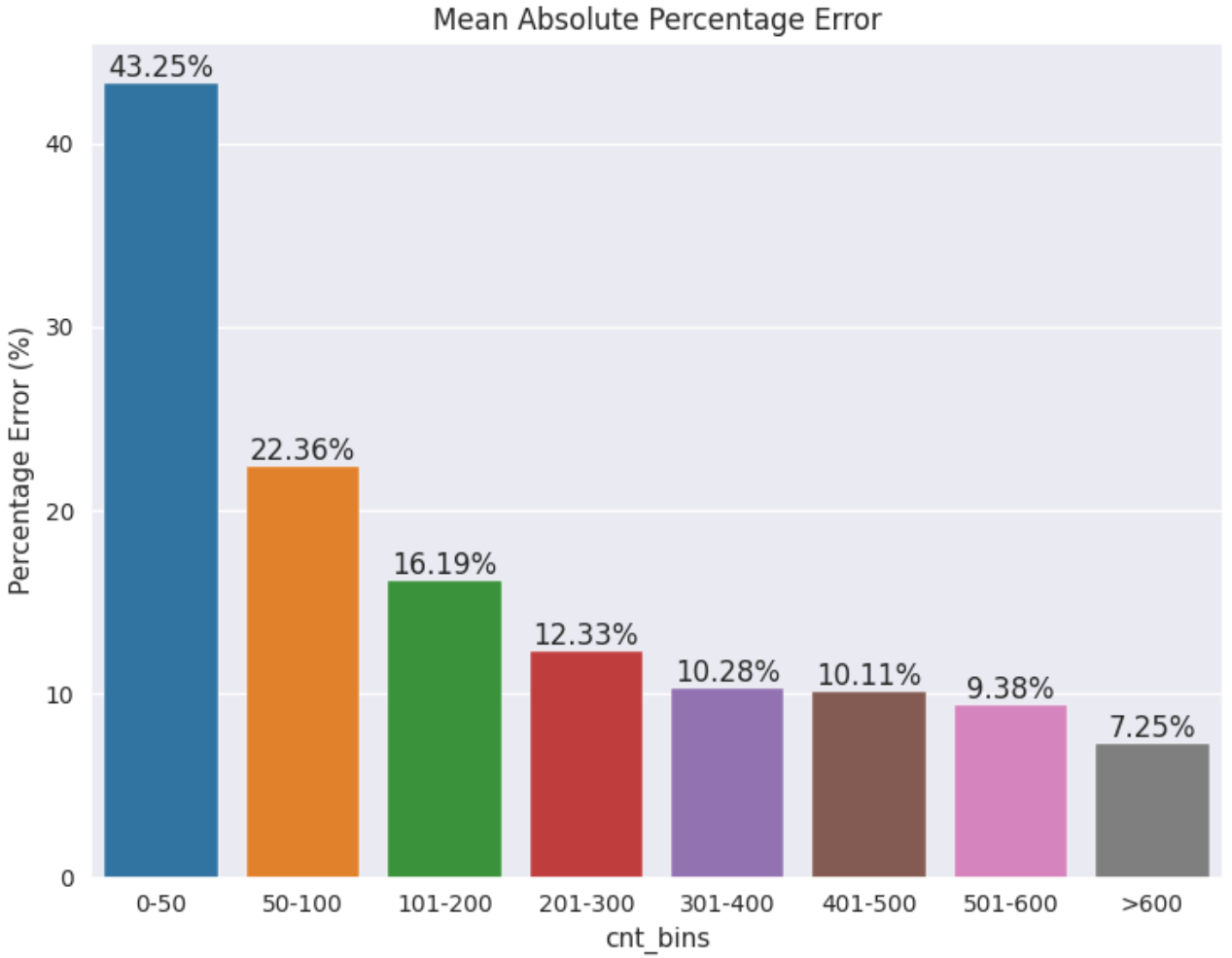
- The predictions are generally accurate, with most data points close to the best-fit line.
- Outliers indicate some cases the model struggles with, likely due to noise or unlearned patterns.
- With a MAPE of 22%, the errors are within acceptable limits.
- Overall, the model performs well, with minor deviations in a few cases.



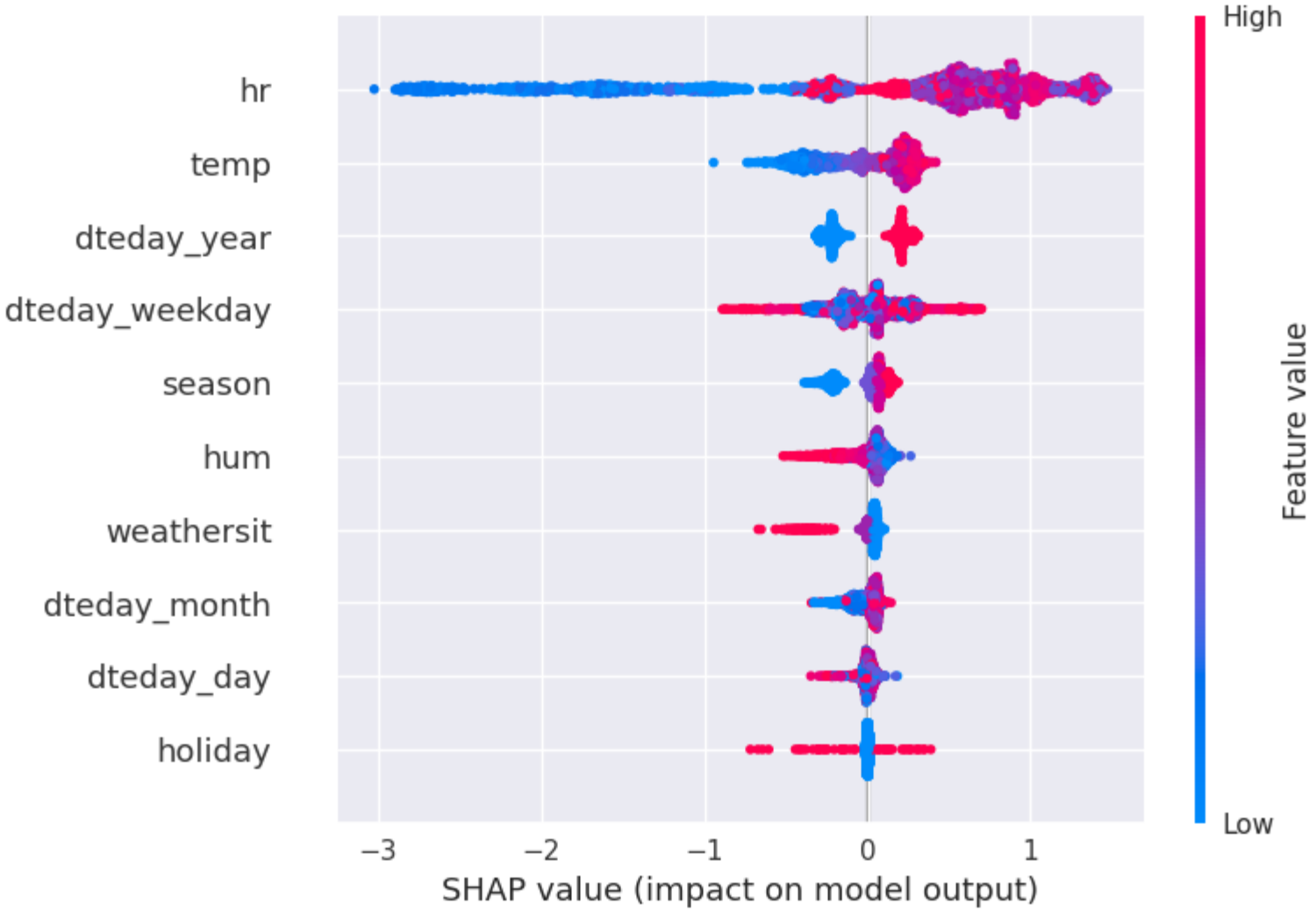


- The predicted distribution resembles the actual data but with a smoother shape and more spread-out values.
- The model underestimates low values and overestimates mid-to-high values, leading to higher variability.
- While the model captures the overall trend, it needs improvement to better align with the sharpness of low-value peaks and reduce overestimation for mid-to-high values.





- The model is less reliable for small values (0-50) with a high MAPE of 43.25%, though the MAE is low.
- Performance improves with higher value ranges, showing progressively lower MAPE (22.36% for 50-100, 16.19% for 101-200, and 12.33% for 201-300).
- For larger values (301-600+), the model performs best, with MAPE ranging from 7.25% to 10.28%.
- While accuracy for small values needs improvement, the model performs well for mid-to-high ranges.



- hr (hour):Peak hours positively impact predictions, while early hours have a negative effect.
- temp (temperature): Higher temperatures increase predictions, while lower temperatures reduce them.
- dteday_year:2012 has a positive impact, while 2011 has a negative one.
- season & hum (humidity): Season and humidity show mixed effects, with high humidity generally lowering predictions.
- weathersit: High values of `weathersit` have a strong negative impact.
- dteday_weekday & dteday_month: These have smaller, consistent impacts, with lower months showing a negative influence.
- Holiday: Minimal impact overall.

Data Testing

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	CatBoost Regressor	20.9802	1235.0449	35.1432	0.9616	0.2736	0.2229

	cnt	prediction_label	error	abs_error	square_error	percentage_error
10417	140	153.423592	13.423592	13.423592	180.192833	0.095883
12030	445	381.258822	-63.741178	63.741178	4062.937809	0.143239
8646	30	19.754169	-10.245831	10.245831	104.977059	0.341528
6596	89	111.785944	22.785944	22.785944	519.199259	0.256022
6691	27	20.592510	-6.407490	6.407490	41.055934	0.237314

- The CatBoost Regressor model shows strong performance with an R2 of 0.9616, explaining most of the variation in the target variable.
- With a MAE of 20.98 and MAPE of 22.29%, predictions are accurate and consistent.
- However, some outliers, like errors of -63.74 and 22.79, indicate areas where the model struggles.
- Overall, it provides reliable predictions with acceptable error levels, making it suitable for practical use.

Registered Members Prediction

Registered Member Prediction:

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	CatBoost Regressor	17.2709	856.7939	29.2710	0.9614	0.2865	0.2325

	registered	prediction_label
10417	107	127.102321
12030	344	299.304862
8646	19	15.325491
6596	62	80.426648
6691	25	20.734107

- For registered members, the final CatBoost Regressor model achieves strong predictive performance with low error metrics: MAE of 17.27, MAPE of 23.25%, and an R2 value of 96.14%.
- This demonstrates that the model effectively captures the variations in the data for registered members.

Casual Members Prediction

Casual Member Prediction:

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	CatBoost Regressor	7.7143	188.1530	13.7169	0.9226	0.4766	0.4089

	casual	prediction_label
10417	33	37.433615
12030	101	83.484538
8646	11	4.480155
6596	27	25.052152
6691	2	1.195057

- For casual members, the model performs well with an MAE of 7.71 and an R2of 92.26%. However, the MAPE of 40.89% reflects higher relative errors due to the small value range (1-50) and predictions near zero. Despite this, the MAE suggests reasonable predictions.
- For example, the model predicts an average demand of 35 casual members per hour with a 22.28% error rate, leading to an allocation of around 43 bicycles.
- Overall, while there are some inaccuracies, the model is sufficient for practical applications like bicycle allocation and revenue estimation, with a positive profit margin.

Conclusion

Model Processing

- Primary Model: CatBoost is the top choice for bike rental demand prediction due to its high accuracy, ability to handle non-linear relationships, and stable performance.
- Hyperparameter Tuning: Continue using the tuned parameters (iterations=1400, learning_rate=0.06, depth=8) for optimal results.
- Model Efficiency: LightGBM is a strong alternative for faster training, ideal for real-time applications and iterative tuning.
- Avoid Linear Models: Linear models should be avoided as they can't capture the dataset's non-linear relationships, except for simple baseline comparisons.
- Further Refinements: Explore ensemble methods and additional hyperparameter tuning to further improve model performance.

Model Evaluation

- Learning Curve: The TransformedTargetRegressor shows steady improvements with more data, but gains plateau after ~6000 instances, indicating good performance for the current dataset size.
- Residuals & Generalization: Residuals are centered around zero with no patterns, indicating accurate predictions and minimal bias. High R^2 values (0.970 for training, 0.956 for test) show strong predictive ability and good generalization.
- Prediction Accuracy: The model is generally accurate with a MAPE of 22%, though outliers, likely due to noise, need improvement.
- Performance by Range: The model struggles with small values (MAPE 43.25%) but performs well for larger values, achieving MAPE as low as 7.25% for values over 301.
- Feature Impacts:
- - Hour: Peak hours positively influence predictions, early hours negatively.
- - Temperature: Higher temperatures increase predictions.
- - Year: 2012 positively affects predictions, 2011 negatively.
- - Humidity: Higher humidity reduces predictions.
- - Weather: Higher values of weathersit negatively affect predictions.
- - Weekday & Month: Lower months show a negative effect.
- - Holiday: Minimal impact.

Business Impact

- Accurate demand predictions enable companies to optimize bicycle allocation, ensuring availability matches user demand. This helps reduce the risk of over or under-allocating resources, which can lead to unnecessary operating costs or lost revenue opportunities.
- For instance, consistent allocation based on static forecasts can result in excess bikes during low-demand periods (overestimating demand), leading to increased operational costs. Alternatively, underestimating demand during peak times can result in missed revenue opportunities as bikes become unavailable.
- Demand prediction insights can be leveraged to design targeted marketing campaigns. Promotions or special offers can be scheduled for periods of high demand, such as weekends or holidays, improving customer satisfaction and increasing sales.

Recommendation

- To address limitations in predicting small demand values (1-50), consider fine-tuning the model specifically for this range. This could include targeted feature engineering, adjusting hyperparameters, or experimenting with alternative models better suited for low-demand scenarios.
- Reevaluate the metrics used for model evaluation to ensure they align with business objectives. While MAPE is useful for larger demand values, combining it with MAE for smaller ranges (1-50) may offer a better balance in accuracy across the entire range of demand.
- Conduct further analysis to identify the causes of significant prediction errors. Investigate whether external factors, changes in user behavior, or policy adjustments are influencing demand patterns, and ensure these factors are accounted for in the model.

End.