

# Linear Regression Analysis Report

## Dataset

We worked with the Bike Sharing Dataset, which contains records of bike rentals in Washington D.C. across 2011–2012.

**Target Variable:** Count (cnt) → predicting the number of bicycles rented based on conditions like weather and time of the year.

**Features used:** time attributes (season, yr, mnth, weekday, workingday), weather (weathersit, temp, hum, windspeed)

## Preprocessing

Before training models, the dataset was preprocessed to ensure consistency, and make features suitable for linear regression methods.

- Data Cleaning
  - Dropped irrelevant identifiers such as instant (record index) and dateday (string date).
  - Dropped casual and registered because they directly sum to the target count and would cause data leakage.
- Categorical Variables
  - Converted categorical variables into numerical representations using one-hot encoding:
    - season (spring, summer, fall, winter)
    - weathersit (clear, mist, light rain/snow, heavy rain/snow)
    - weekday (0–6)
- Feature Scaling
  - We didn't have to do scaling since the numeric features were already scaled in the dataset with a mean of 0 and standard deviation of 1.
- Train-Test Split
  - The dataset was divided in a 80/20 split for training and testing.
- Correlation Analysis
  - Generated a correlation heatmap and a table in relation to count to see which features are affecting bicycle count the most.
  - Some features like day of the week did not matter as much so they were dropped from the dataframe.
  - At the end we only used 14 feature columns after having a total of 27 columns after encoding the categorical variables.



## SGDRegressor - Sklearn

## Hyperparameter Tuning

We tuned a wide range of hyperparameters using GridSearchCV. The following image shows the various combinations of hyperparameters used in the current model.

```
#hyperparameters to test
param_grid = {
    "loss": ["squared_error", "huber", "epsilon_insensitive", "squared_epsilon_insensitive"],
    "penalty": ["l2", "l1", "elasticnet"],
    "alpha": [1e-5, 1e-4, 1e-3, 1e-2],
    "max_iter": [1000, 2000, 3000, 5000],
    "tol": [1e-3, 1e-4, 1e-5],
    "learning_rate": ["optimal", "invscaling", "constant"],
    "eta0": [0.01, 0.05]
}
```

We also also tried testing with more values for the current parameters and other parameters like fit\_intercept, shuffle, l1\_ratio, and a few others but the model kept learning for almost 2 hours since it was testing a variety of combinations to find the best.

 **Executing (1h 48m 47s)**  **Python 3**

The following values turned out to be the best performance:

Best parameters: {'alpha': 1e-05, 'eta0': 0.05, 'learning\_rate': 'invscaling', 'loss': 'squared\_error', 'max\_iter': 3000, 'penalty': 'l1', 'tol': 1e-05}

## Evaluation Metrics

- Train RMSE: 861.910, R-Squared: 0.797
- Test RMSE: 867.526, R-Squared: 0.812
- Baseline RMSE: 2022.173
- Improvement over Baseline: 99.979

The baseline model, which predicts the average number of bike rentals for every instance, produces an RMSE of over 2000. While, our tuned SGDRegressor reduces this error to ~868 rentals, an improvement of nearly 100%. Also, the close alignment between training and testing RMSE values shows that the model generalizes well without overfitting. In this context, an RMSE of ~868 means that on average, the model's predictions are off by about 868 bikes per day.

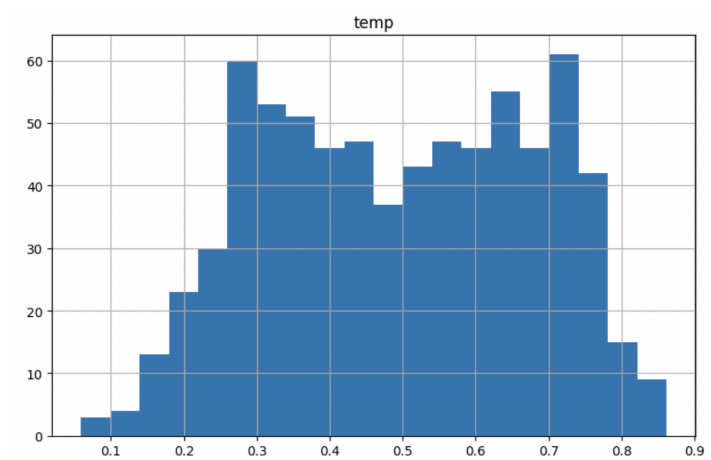
## Plots

Correlation Table with Bicycle Count

	cnt
cnt	1.00
registered	0.95
casual	0.67
atemp	0.63

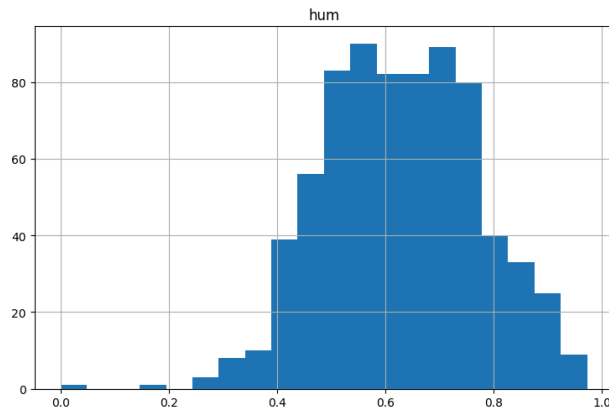
weekday_4	0.03
weekday_6	0.01
weekday_3	0.01
weekday_2	0.00
weekday_1	-0.04
weekday_0	-0.06
holiday	-0.07
hum	-0.10
weathersit_2	-0.17
windspeed	-0.23
weathersit_3	-0.24
season_1	-0.56

- registered (0.95) and casual (0.67): these dominate the correlation since they directly sum to cnt. They were excluded from modeling to avoid leakage.
- The weekdays were also removed because they did not contribute much to predicting the count of the bike rentals.
- weathersit\_2 (mist/clouds, -0.17) and weathersit\_3 (rain/snow, -0.24): adverse weather significantly reduces rentals. season\_1 (spring, -0.56): spring is strongly negatively correlated, likely due to cooler, rainier conditions.



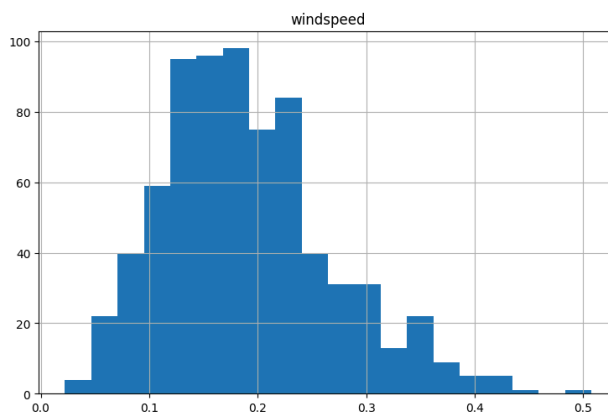
**Graph 1 (temp)**

Spread fairly evenly from 0.1–0.8 (normalized), with peaks at moderate and warmer ranges. There is a correlation of  $r = 0.63$ , which signals rentals rise as temperatures increase — more people bike in warmer conditions.



**Graph 2 (hum)**

The graph is bell-shaped, centered around 0.6–0.7. Few cases of very low or very high humidity. With a correlation  $r = -0.10$ , it suggests that rentals are weak and negative and that humidity doesn't influence bike demand. It also suggests that humid conditions can negatively impact the rentals.



**Graph 3 (windspeed)**

Strongly right-skewed with most values  $< 0.25$ . Very few high wind cases. With a correlation:  $r = -0.23$ , it shows that higher wind speeds reduce bike rentals, but since most hours have low wind, this effect is less frequent. Still, wind contributes negatively in some cases.

## OLS - Library of statsmodels

OLS Regression Results						
=====						
Dep. Variable:	cnt	R-squared:	0.817			
Model:	OLS	Adj. R-squared:	0.813			
Method:	Least Squares	F-statistic:	212.1			
Date:	Mon, 22 Sep 2025	Prob (F-statistic):	2.49e-201			
Time:	19:23:10	Log-Likelihood:	-4746.6			
No. Observations:	584	AIC:	9519.			
Df Residuals:	571	BIC:	9576.			
Df Model:	12					
Covariance Type:	nonrobust					
=====						
	coef	std err	t	P> t	[0.025	0.975]
const	999.9078	214.838	4.654	0.000	577.938	1421.877
yr	1993.3211	69.957	28.494	0.000	1855.917	2130.725
mnth	-14.5179	19.417	-0.748	0.455	-52.655	23.619
holiday	-535.0211	223.934	-2.389	0.017	-974.856	-95.187
workingday	151.2542	75.525	2.003	0.046	2.913	299.595
temp	5240.5330	353.229	14.836	0.000	4546.745	5934.321
hum	-1130.9463	332.510	-3.401	0.001	-1784.039	-477.853
windspeed	-2525.0448	482.396	-5.234	0.000	-3472.533	-1577.557
season_1	-675.3035	102.511	-6.588	0.000	-876.649	-473.958
season_2	511.1625	88.183	5.797	0.000	337.959	684.366
season_3	142.6425	127.197	1.121	0.263	-107.189	392.474
season_4	1021.4062	124.482	8.205	0.000	776.907	1265.905
weathersit_1	1078.9905	81.077	13.308	0.000	919.746	1238.235
weathersit_2	589.0186	99.995	5.890	0.000	392.615	785.422
weathersit_3	-668.1013	182.864	-3.654	0.000	-1027.270	-308.932
=====						
Omnibus:	60.681	Durbin-Watson:	1.968			
Prob(Omnibus):	0.000	Jarque-Bera (JB):	159.166			
Skew:	-0.528	Prob(JB):	2.74e-35			
Kurtosis:	5.329	Cond. No.	5.28e+16			
=====						

Above are the model diagnostics when trained with the exact data used to train the SGD model

### Explanation & Interpretation of the output diagnostics

- coef
  - The coefficient metric shows the expected change in the bike rentals per predictor. For example, temperature has coef of 5240.5 which is quite high and it shows that it is one of the strongest positive predictors of bike rentals. On the opposite end, windspeed has a coef of -2525 which means it has a big impact in reducing rentals.
- standard error
  - Most of the predictor variables have relatively lower std error meaning they have stable effects on daily bike usage patterns. However, higher std errors such as the one for temperature(353.23) varies when it comes to its impact on bike rentals.
- t-value
  - The t-value represents whether a predictor variable is statistically significant. Most of our predictor variables have absolute values of the t-value that are well away from 0 meaning they are significant. Month and season\_3 are the 2 variables closest to 0, meaning they are the least significant.
- p-value

- The p-values are correlated with the t-value and show significance if the p-value is less than 0.05. In our case, all p-values are less than 0.05 except month and season\_3 meaning they are not significant.
- R-squared
  - The R-squared metric for our model is 0.817 which means the OLS model explains 81.7% of variance in bike rentals. This R-squared metric is close to 1 which shows that our model is quite strong and accurate.
- R-squared adjusted
  - The R-squared adjusted for our model is 0.813 which is very close to the R-squared metric(0.817). What this means is that our model is quite good and also that all our features are important/useful.
- F-statistic
  - The F-statistic for our model is 212.1, and since this number is quite high, it shows that the variance explained by our model is much higher than the variance in the errors. What this means is that the model is meaningful/significant and the features we used to train it are all important/useful.

### **Evaluation Metrics for OLS model**

- Test RMSE: 812.193, R-Squared: 0.817
- Baseline RMSE: 2022.173
- Improvement over Baseline: 99.9801

### **Model Comparison**

The OLS model performs slightly better than the SGD model. The OLS model has a test RMSE of approximately 812.19, while the SGD model's test RMSE is around 867.53. The lower RMSE shows that the OLS model's predictions are closer to the actual bike counts compared to the SGD model. Additionally the R-squared values for both models are also extremely similar (0.812 vs 0.817) but since they are both close to 1, this means both models are strong in prediction. Definitely, both models are a lot better than the simple baseline of predicting the average bike count.

## **How to run**

The .ipynb notebook file is very easy to run. You just need to download the file and upload to Google Colab and everything should be populated and be able to run with the “Run all” button.