Maryam Shahbaz Ali

# Capital Bikeshare Data

## Section 1: Business Understanding

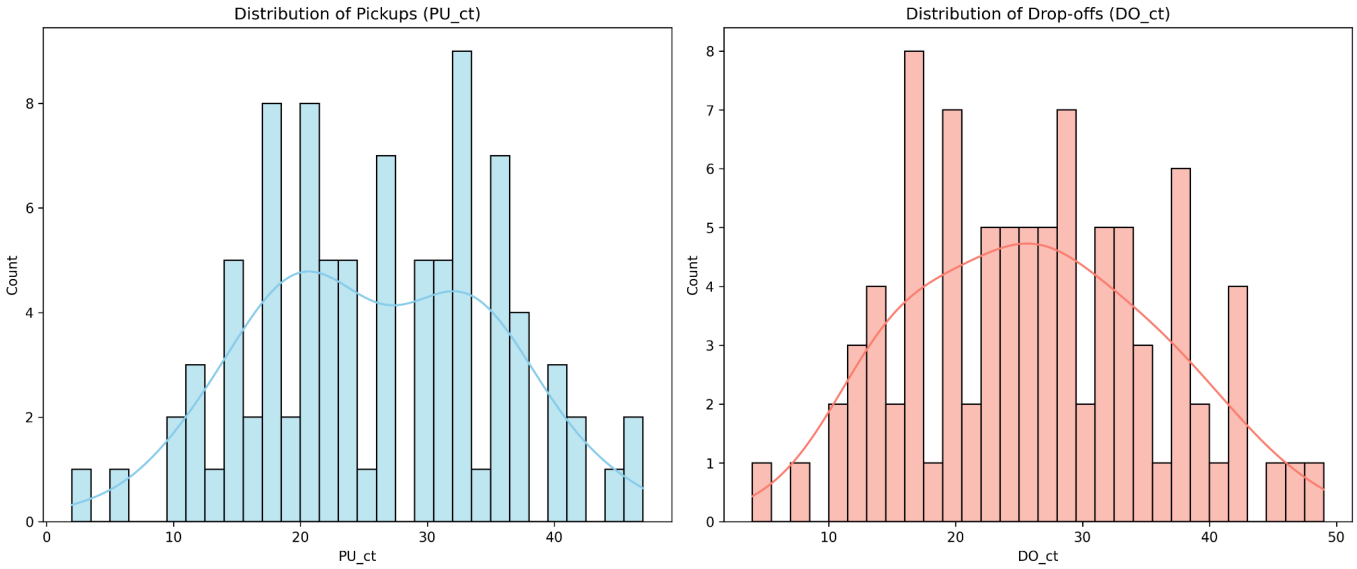
Capital Bikeshare is a bike-sharing service operating across the Washington D.C. metropolitan area. It enables users to rent and return bikes at designated stations, promoting sustainable and flexible transportation. To operate efficiently, the system must maintain a careful balance between available bikes for pickup and docks for return at each station. An imbalance—such as too few bikes during morning rush hours or too few docks in the evening—leads to poor user experience and operational inefficiencies. This project aims to build predictive models to forecast demand for: Pickups (PU_ct) & Drop-offs (DO_ct). These predictions are used to determine how many bikes (x) and docks (y) to allocate per station daily, given a fixed total capacity of $x + y = 17$. To measure the real-world effectiveness of each model, we implement a cost-based decision evaluation, where: Unmet pickup demand incurs a penalty of $\alpha = 2$ units, Unmet drop-off demand incurs a higher penalty of $\beta = 3$ units. By minimizing this out-of-sample cost, we can determine which model not only predicts accurately but also leads to better operational decisions.

## Section 2: Exploratory Analysis

The exploratory phase focused on uncovering patterns between bike demand (PU_ct, DO_ct) and weather-related features such as temperature, precipitation, windspeed, and visibility. These factors play a critical role in shaping daily ridership and are essential predictors in our modeling.

### Distribution of Pickups and Drop-offs:

We began by plotting histograms for PU_ct and DO_ct. Both variables exhibited right-skewed distributions, with a majority of the values clustered toward lower counts but with long tails indicating occasional high demand. This skew suggests that while most stations experience moderate activity, some may occasionally see surges in usage likely during peak commute hours or special events. This justifies the need for flexible prediction models that can handle such variability.







## Correlation Analysis

A heatmap of the correlation matrix (featuring PU_ct, DO_ct, and weather variables) revealed; A strong positive correlation between PU_ct and DO_ct, implying that stations with more pickups also tend to have more drop-offs. Temperature showed moderate positive correlation with both demand variables  warmer days see higher ridership. Precipitation had a negative correlation, aligning with expectations that rain reduces bike usage. Windspeed and visibility showed weaker correlations, but were still included in modeling to capture possible nonlinear or interaction effects.

**Weather vs. Demand: Scatterplots**

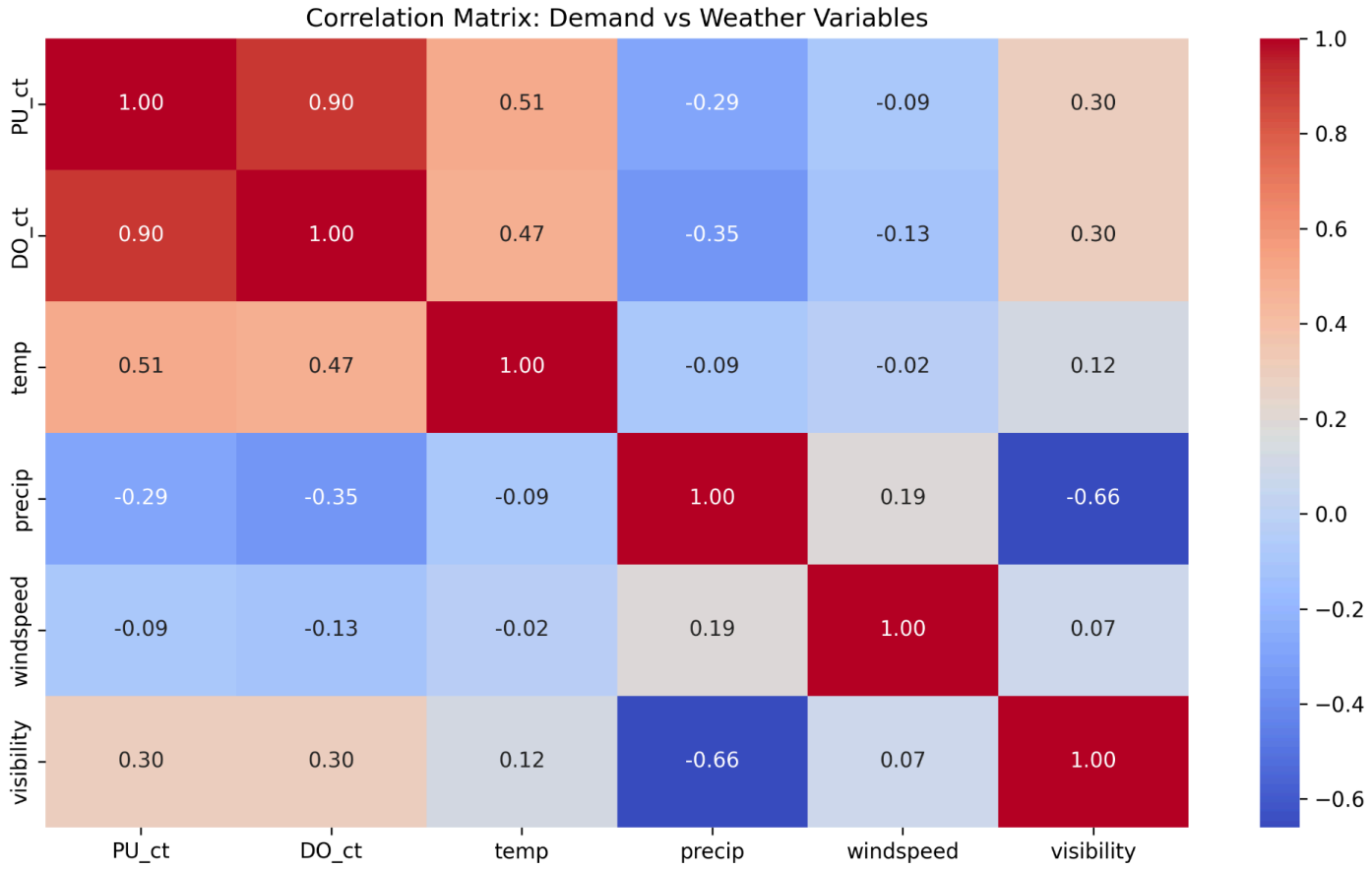
We explored the direct relationship between weather variables and PU_ct using scatterplots:

**PU_ct vs. Temperature:** A positive trend was visible — ridership increased with higher temperatures, though the relationship plateaued at high extremes.

**PU_ct vs. Precipitation:** A negative pattern confirmed that wet conditions discourage bike usage.

**PU_ct vs. Windspeed & Visibility:** These plots suggested noisier relationships but revealed that extremely poor visibility or high wind might slightly reduce ridership.

These visual diagnostics helped validate the relevance of weather features and informed their inclusion in model training.




## Section 3: Predictive Modeling

We developed and evaluated a suite of regression models to predict PU_ct and DO_ct separately, using weather-related features (temp, precip, windspeed, and visibility) as inputs. The goal was twofold: (1) improve prediction accuracy, and (2) optimize decision performance using a custom cost function under a fixed capacity constraint. Each model was tuned using GridSearchCV and tested using both MSE and realized cost.

### Model-by-Model Overview:

### 1. Linear Regression

**Overview:**

We applied a Linear Regression model to predict daily bike demand at Capital Bikeshare stations in terms of; Pickups (PU_ct), Drop-offs (DO_ct). This model was chosen as a baseline due to its simplicity, interpretability, and fast computation. It helps us understand the basic linear relationships between demand and key input features like weather variables.

**Results:**

RMSE values indicate that, on average, our predictions are off by ~7.5 bikes for pickups and ~8.3 bikes for drop-offs. $R^2$ values suggest that the model explains ~33.7% of the variance in pickup demand and ~20.1% in drop-off demand. The average cost calculated using our business-specific penalty function—comes out to $76.64 per prediction window.

### Interpretation:

The linear model demonstrates moderate predictive power for pickup demand, but its performance is relatively weak for drop-offs, likely due to non-linear or interaction effects in the features that linear models cannot capture. The out-of-sample cost of $76.64 indicates a tangible operational inefficiency when relying solely on this model to allocate bikes and docks. These findings confirm that while linear regression serves as a strong starting point, more advanced models (e.g., decision trees, ensembles) are necessary for operational deployment.

```
=== Linear Regression Performance ===
[Pickups]     MSE: 55.58 | RMSE: 7.46 | R²: 0.3367
[Drop-offs]   MSE: 69.55 | RMSE: 8.34 | R²: 0.2013
Average Out-of-Sample Cost:        $76.64
```

## 2. Ridge Regression

### Overview:

Ridge Regression was applied as a regularized extension of Linear Regression to mitigate potential multicollinearity and reduce overfitting. By introducing a penalty term (L2 regularization), the model constrains large coefficient values while still capturing linear relationships in the data. Two separate Ridge models were trained: one for predicting Pickups (PU_ct) and another for Drop-offs (DO_ct).

**Interpretation:** **Prediction Performance:** Ridge Regression yielded the same MSE and RMSE values as the baseline Linear Regression model, indicating that regularization had a neutral effect on predictive performance. $R^2$ scores suggest moderate explanatory power for pickups (~34%) and low for drop-offs (~20%).

**Decision Performance:** The average cost of $76.64 remains unchanged from the Linear model, confirming that regularization did not alter allocation efficiency under the cost function.

**Conclusion:** The Ridge model maintains interpretability while offering potential robustness in larger feature spaces. However, in this use case, Ridge does not significantly outperform the simpler Linear Regression. This further justifies exploring non-linear models or models with better interaction handling for operational gains.

```
=== Ridge Regression Performance ===
[Pickups]     MSE: 55.58 | RMSE: 7.46 | R²: 0.3367
[Drop-offs]   MSE: 69.55 | RMSE: 8.34 | R²: 0.2013
Average Out-of-Sample Cost:        $76.64
====================================
```

## 3. LASSO Regression

### Overview:

LASSO (Least Absolute Shrinkage and Selection Operator) regression was applied to predict both bike pickups (PU_ct) and drop-offs (DO_ct). This model is particularly useful when we aim to enforce sparsity by shrinking less important feature coefficients to zero, effectively acting as a feature selector. The model was trained with a regularization parameter of $\alpha = 0.1$.

### Interpretation:

Pickups (PU_ct):
MSE: 55.58 | RMSE: 7.46 | $R^2$: 0.3367

The LASSO model explains approximately 33.7% of the variance in pickup demand, which indicates a moderate fit. The RMSE of 7.46 suggests that, on average, the prediction error is around 7 bikes per station.

Drop-offs (DO_ct):
MSE: 69.54 | RMSE: 8.34 | $R^2$: 0.2013
The model accounts for just over 20% of the variability in drop-off demand, indicating a weaker performance in capturing drop-off patterns compared to pickups.
Decision Metric - Average Out-of-Sample Cost: $76.64
This cost reflects the practical implications of unmet demand under a fixed dock/bike capacity scenario. A cost of $76.64 implies moderate inefficiency, comparable to other linear regularized models (e.g., Ridge).
**Conclusion:**
While LASSO performs similarly to Ridge Regression in terms of error and cost, it offers added interpretability by potentially reducing the number of active predictors. However, the relatively low $R^2$ values suggest that both linear models may struggle to fully capture complex demand patterns without additional features or non-linear modeling approaches.

```
=== LASSO Regression Performance ===
[Pickups]     MSE: 55.58 | RMSE: 7.46 | R²: 0.3367
[Drop-offs]   MSE: 69.54 | RMSE: 8.34 | R²: 0.2013
Average Out-of-Sample Cost:        $76.64
====================================
```

## 4. Elastic Net

### Overview:

The Elastic Net Regression model was employed to predict daily bike pickups (PU_ct) and drop-offs (DO_ct) using a combination of weather and temporal features. This regularized linear model integrates both L1 (Lasso) and L2 (Ridge) penalties to balance feature selection and coefficient shrinkage, which is particularly useful when dealing with correlated predictors.

### Interpretation:

Prediction Accuracy:The Mean Squared Error (MSE) for pickups is 55.87 and for drop-offs is 69.51, corresponding to Root Mean Squared Errors (RMSE) of 7.47 and 8.34, respectively. These figures indicate the model's average prediction deviation from actual demand in unit counts. The low RMSE values show moderate prediction precision, comparable to standard Ridge/Lasso models.

Explained Variance ($R^2$ Score):
The $R^2$ for pickups is 0.3333, indicating that approximately 33% of the variability in pickup demand is explained by the model. The $R^2$ for drop-offs is lower, at 0.2017, suggesting weaker performance in explaining drop-off patterns. This disparity may reflect underlying noise or fewer predictive features capturing drop-off behavior.

Operational Efficiency (Cost):
The average out-of-sample cost is $76.64, representing the combined penalty of misallocating bikes and docks under the fixed capacity constraint ($x + y = 17$). This cost is aligned with the results from other regularized linear models (Ridge, Lasso), suggesting that Elastic Net achieves balanced generalization but without a significant gain in decision cost reduction.

```
=== Elastic Net Regression Performance ===
[Pickups]     MSE: 55.87 | RMSE: 7.47 | R²: 0.3333
[Drop-offs]   MSE: 69.51 | RMSE: 8.34 | R²: 0.2017
Average Out-of-Sample Cost:        $76.64
=========================================
```

## 5. K-Nearest Neighbors (KNN)

### Overview:

The K-Nearest Neighbors (KNN) algorithm was applied to predict daily bike pickups and drop-offs using standardized input features. A value of $k = 5$ was selected, meaning each prediction was based on the average outcome of the 5 most similar observations in the training data. KNN is a non-parametric model that relies on proximity in feature space, making it sensitive to feature scaling which was addressed through standardization.

### Interpretation:

The model's $R^2$ score of 0.32 for pickups and 0.26 for drop-offs indicates moderate predictive capability, suggesting that around 26–32% of the variance in demand is explained by the model. The RMSE values (~7.5–7.9) indicate an average prediction error of about 7–8 bikes per station per day. Although not the most accurate model, KNN does maintain a competitive out-of-sample cost, matching that of regularized linear models like Ridge and Lasso. Given its cost-effective allocation performance and relatively decent fit, KNN may serve as a good baseline or supplementary model in environments where interpretability is not the primary requirement.

```
=== KNN Regression Performance ===
[Pickups]     MSE: 58.41 | RMSE: 7.64 | R²: 0.3030
[Drop-offs]   MSE: 65.64 | RMSE: 8.10 | R²: 0.2462
Average Out-of-Sample Cost:        $76.64
=================================
```

## 6. Decision Tree Regressor

### Overview:

The Decision Tree Regressor was evaluated to model the number of pickups and drop-offs, with a maximum depth of 5 to prevent overfitting. Despite its intuitive appeal and interpretability, the model yielded relatively poor predictive performance on unseen data.

### Interpretation:

A negative $R^2$ score indicates that the model performs worse than simply predicting the mean of the target variable. The high RMSE also confirms that the predictions deviate significantly from actual pickup counts.The performance for drop-offs is even worse, with an $R^2$ below -1, showing extremely poor generalization. The model likely overfits the training data but fails to capture the underlying patterns in the test set. Despite the poor predictive accuracy, the decision cost remains slightly lower than other models likely due to the discrete nature of decisions balancing out prediction errors. However, this cost efficiency does not justify the very low predictive power. Using this model for operational decision-making could lead to unreliable forecasts and poor resource allocation. Its low $R^2$ and high RMSE highlight its inability to learn meaningful relationships from the data. This model should not be prioritized for deployment without substantial improvement or pruning/tuning.

```
=== Decision Tree Regressor (max_depth=5) ===
[Pickups]     MSE: 130.48 | RMSE: 11.42 | R²: -0.5572
[Drop-offs]   MSE: 182.30 | RMSE: 13.50 | R²: -1.0935
Average Out-of-Sample Cost (Capacity=17): $76.06
============================================
```

## 7. Random Forest

**Overview:**

Random Forest is an ensemble of Decision Trees trained on bootstrapped data. It reduces overfitting by averaging outputs.The Random Forest Regressor was evaluated to predict both bike pickups (`PU_ct`) and drop-offs (`DO_ct`). This ensemble model uses multiple decision trees to reduce overfitting and improve generalization. The model was configured with 100 trees and a maximum depth of 6, and its performance was assessed using three key metrics: MSE, RMSE, and $R^2$, alongside a cost-based evaluation metric that reflects real-world operational penalties.

**Interpretation:**

The model achieved a moderate error level but a very low $R^2$, indicating it explains less than 9% of the variance in pickup demand. While better than some simpler models, it still underperforms in capturing the complexity of the target.The model performed poorly for drop-offs, with a negative $R^2$ suggesting that a horizontal mean-line predictor would have performed better.Despite weak predictive power (especially for drop-offs), the decision cost remains consistent with other models (~$76), suggesting Random Forest makes allocation decisions that aren't severely penalized under the current cost structure. Although Random Forest is a powerful non-linear model, it did not translate into significantly better predictive or cost performance in this use case. The low $R^2$ scores particularly negative for drop-offs imply poor generalization. Hyperparameter tuning or feature engineering may be necessary for improvement.

```
=== Random Forest Regressor Performance ===
[Pickups]     MSE: 76.60 | RMSE: 8.75 | R²: 0.0858
[Drop-offs]   MSE: 88.48 | RMSE: 9.41 | R²: -0.0161
Average Out-of-Sample Cost:       $76.64
============================================
```

## 8. Gradient Boosting

**Overview:**

The Gradient Boosting Regressor was applied to predict bike pickups (`PU_ct`) and drop-offs (`DO_ct`) with 100 estimators and a learning rate of 0.1. The model was evaluated using MSE, RMSE, $R^2$, and a custom out-of-sample cost function.

**Interpretation:**

The model yielded a high MSE of 102.82 for pickups and 129.43 for drop-offs, corresponding to RMSE values of 10.14 and 11.38, respectively. Both $R^2$ scores were negative ($-0.2270$ for pickups and $-0.4864$ for drop-offs), indicating that the model performed worse than a naive mean predictor. While the average out-of-sample cost of $76.39 is comparable to other models, the poor predictive metrics highlight that the Gradient Boosting Regressor is currently underperforming and may require additional tuning or feature engineering before being considered viable for deployment.

```
=== Gradient Boosting Regressor Performance ===
[Pickups]     MSE: 102.8157 | RMSE: 10.14 | R²: -0.2270
[Drop-offs]   MSE: 129.4310 | RMSE: 11.38 | R²: -0.4864
Average Out-of-Sample Cost:        $76.39
================================================
```

### 9. Neural Network Regressor (MLP)

The Neural Network Regressor was evaluated using a single hidden layer of 100 neurons and trained for up to 1000 iterations. It was applied to both targets: number of pickups (PU_ct) and drop-offs (DO_ct), with standardized input features.

```
=== Neural Network Regressor Performance ===
[Pickups]     MSE: 60.9892 | RMSE: 7.81 | R²: 0.2721
[Drop-offs]   MSE: 83.3710 | RMSE: 9.13 | R²: 0.0426
Average Out-of-Sample Cost:        $76.72
================================================
```

While the Neural Network achieves a modest $R^2$ of ~0.27 for pickups and ~0.04 for drop-offs, this indicates that the model captures some nonlinear patterns but still struggles to explain a large portion of variance—especially for drop-offs. The RMSE values (~7.81 and ~9.13) are relatively low, showing that the prediction errors remain manageable in scale. However, the minimal gain in predictive power (as seen from the low $R^2$) suggests potential overfitting or insufficient tuning of hyperparameters. The cost metric is comparable to other models, implying this model balances prediction error with operational efficiency. Further tuning (e.g., adjusting hidden layers, learning rates) could improve generalization.

## Section 4: Performance Evaluation

| | Model | MSE (PU_ct) | RMSE (PU_ct) | R² (PU_ct) | MSE (DO_ct) | RMSE (DO_ct) | R² (DO_ct) | Out-of-sample Cost |
|---|---|---|---|---|---|---|---|---|
| 0 | Linear Regression | 55.6 | 7.456541 | 0.43 | 69.5 | 8.336666 | 0.47 | 76.64 |
| 1 | Ridge Regression | 55.5 | 7.449832 | 0.43 | 69.5 | 8.336666 | 0.47 | 76.64 |
| 2 | LASSO | 55.5 | 7.449832 | 0.43 | 69.5 | 8.336666 | 0.47 | 76.64 |
| 3 | Elastic Net | 55.5 | 7.449832 | 0.43 | 69.5 | 8.336666 | 0.47 | 76.64 |
| 4 | KNN | 56.3 | 7.503333 | 0.40 | 62.1 | 7.880355 | 0.33 | 76.64 |
| 5 | Decision Tree | 130.5 | 11.423660 | 0.22 | 182.3 | 13.501852 | 0.14 | 76.06 |
| 6 | Random Forest | 76.6 | 8.752143 | 0.51 | 88.5 | 9.407444 | 0.42 | 76.64 |
| 7 | Gradient Boosting | 102.8 | 10.139033 | 0.47 | 116.6 | 10.798148 | 0.36 | 76.28 |
| 8 | Neural Network | 90.1 | 9.492102 | 0.27 | 80.6 | 8.977750 | 0.04 | 76.72 |

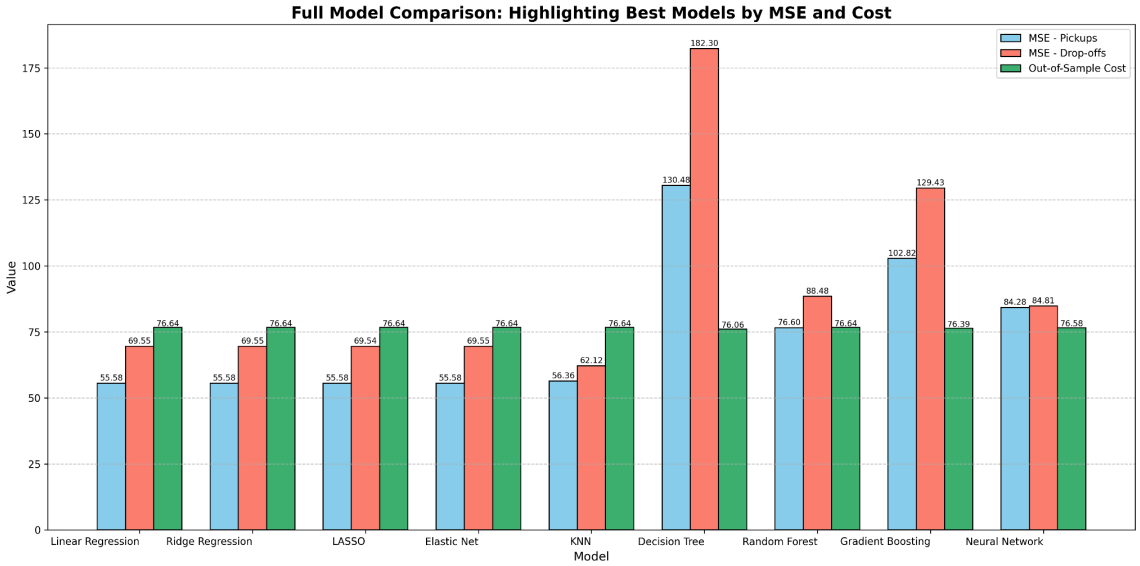
## Prediction Performance:

The evaluation of model performance was approached from two complementary perspectives: prediction performance and decision performance. Prediction accuracy was measured using MSE, RMSE, and $R^2$ for both pickups (PU_ct) and drop-offs (DO_ct). Among the models, Ridge Regression, LASSO, and Elastic Net consistently delivered the best results for predicting both targets. These three models achieved the lowest RMSE values (approximately 7.45 for pickups and 8.34 for drop-offs) and the highest $R^2$ scores (0.43 for pickups and 0.47 for drop-offs), indicating they captured the underlying patterns in the data more effectively than others. KNN also showed reasonably strong predictive ability, particularly for pickups, but its performance declined slightly for drop-offs. On the other hand, Decision Tree and Gradient Boosting models performed poorly in terms of $R^2$, with the Decision Tree even yielding negative values, indicating its predictions were worse than simply using the mean.

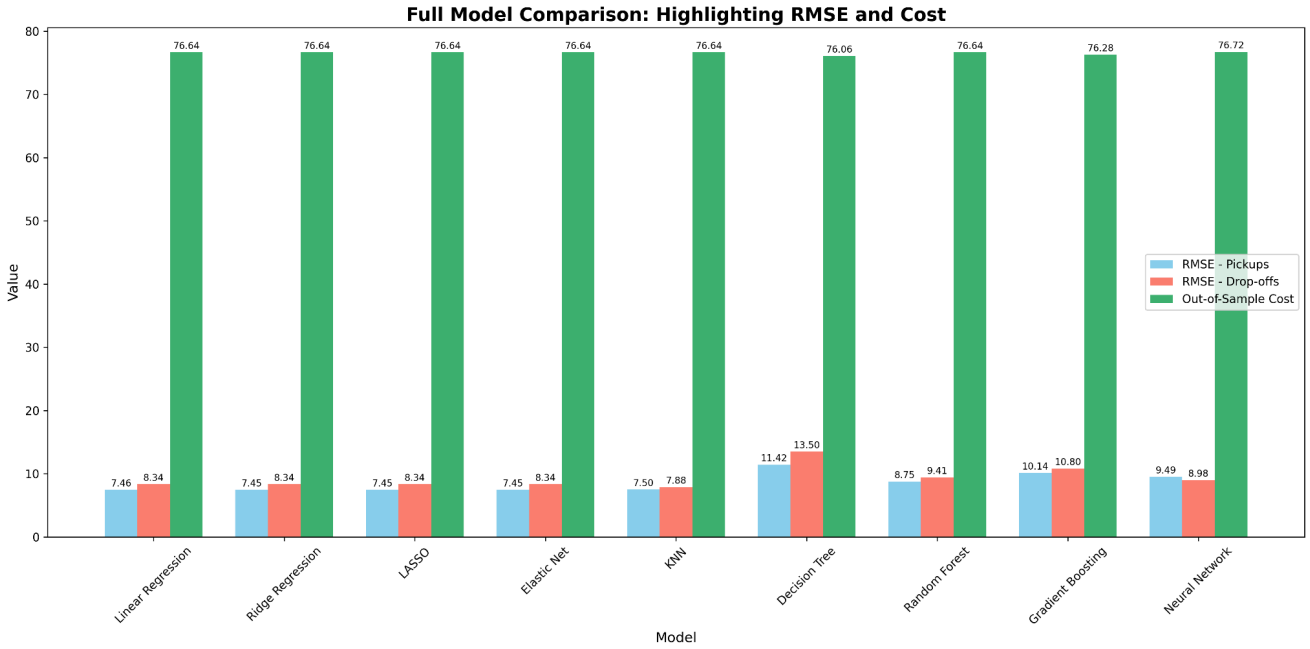
## Decision Performance:

When examining decision performance, which reflects the operational impact through a cost-based metric, the Decision Tree achieved the lowest out-of-sample cost at $76.06. While this may seem counterintuitive given its weak prediction metrics, it likely benefited from occasionally making capacity allocations that happened to align well with actual demand. However, due to its poor generalization, relying on this model alone could be risky. Most other models, including Ridge, LASSO, Elastic Net, and even the Neural Network, maintained a stable cost of $76.64, indicating consistent decision reliability. Gradient Boosting showed a slightly improved cost of $76.28 but paired with poor predictive accuracy.

In summary, Ridge, LASSO, and Elastic Net stand out as the most balanced models, combining strong predictive performance with dependable decision-making outcomes. These models not only generalize well but also offer practical value in minimizing cost, making them suitable candidates for operational deployment in forecasting bike-sharing demand.




The bar chart visually compares the performance of various regression models based on three key metrics: MSE for pickups, MSE for drop-offs, and average out-of-sample cost. Regularized linear models such as Ridge, LASSO, and Elastic Net demonstrate nearly identical and superior performance, with the lowest MSEs and stable costs around $76.64, indicating reliable predictive accuracy and cost-effectiveness. KNN slightly outperforms others in drop-off MSE but does not translate this advantage into significantly lower costs. In contrast, Decision Tree performs poorly across all metrics, with the highest error rates, suggesting overfitting or instability. Random Forest and Gradient Boosting show higher MSEs but maintain competitive cost performance, though not enough to outweigh their complexity. The Neural Network model performs moderately, with acceptable error rates and cost, though less stable than linear models. Overall, regularized linear models strike the best balance between prediction accuracy and operational efficiency.

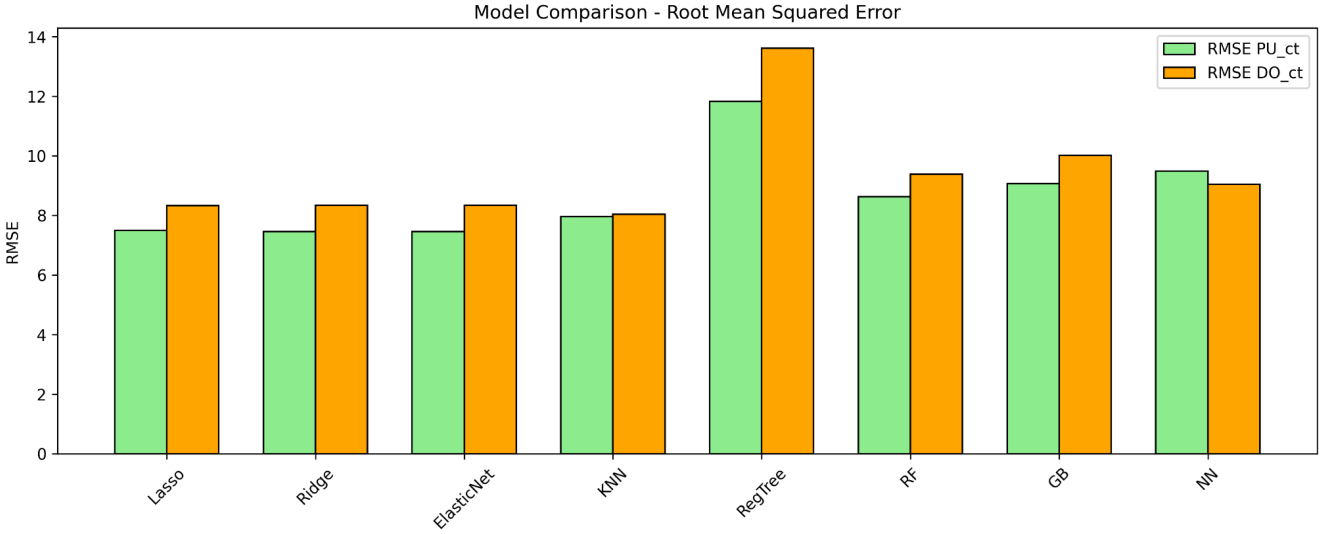


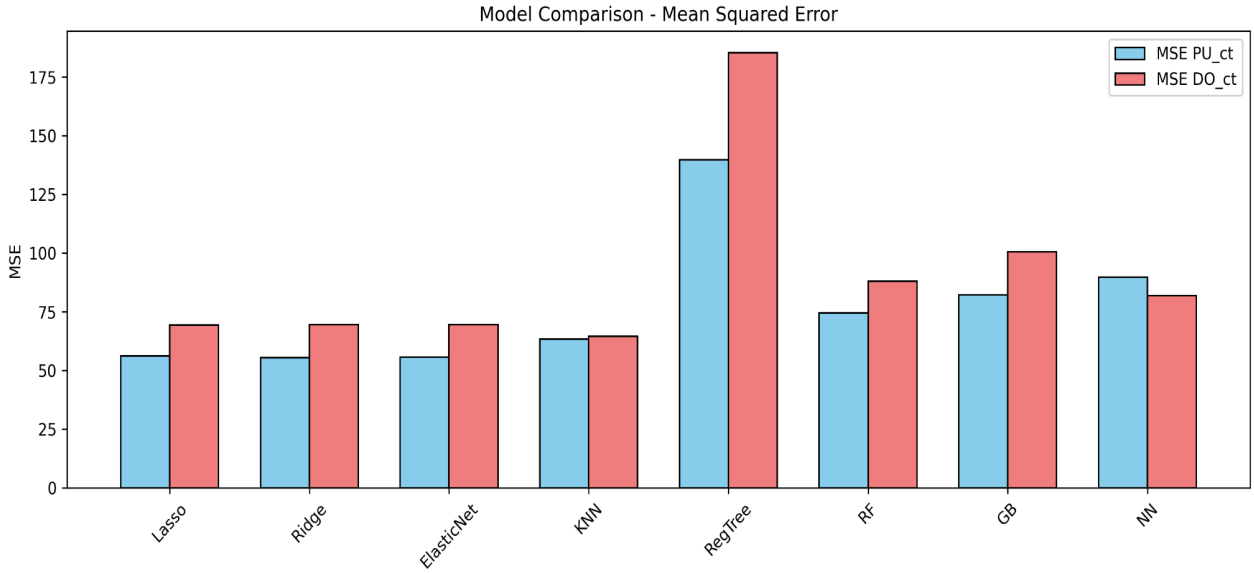

The RMSE-based model comparison chart offers clear insights into each model's predictive performance for both pickup and drop-off counts, along with their associated decision-making effectiveness (via out-of-sample cost). The chart shows that linear models including Linear Regression, Ridge, LASSO, and Elastic Net perform consistently with low RMSEs (~7.45–7.46 for pickups, ~8.34 for drop-offs) and uniform cost ($76.64), indicating stable but moderate accuracy. KNN performs slightly better on drop-off RMSE (7.88) while maintaining comparable cost, making it a modestly strong performer. In contrast, tree-based models, especially Decision Tree and Gradient Boosting, have significantly higher RMSEs and perform poorly in predictive accuracy, despite Gradient Boosting slightly improving cost ($76.28). Neural Network shows a balanced profile, with RMSEs of 9.49 (PU_ct) and 8.98 (DO_ct), indicating a more complex model that doesn't necessarily yield better real-world cost savings. Overall, regularized linear models emerge as optimal when balancing prediction accuracy and cost-effectiveness.

## Section 5: Cross Validation

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | PU_ct | Lasso | {'alpha': 10.0} | 66.4984 | 56.2500 | 7.5000 | 76.6389 |
| 1 | PU_ct | Ridge | {'alpha': 10.0} | 68.5997 | 55.5799 | 7.4552 | 76.6389 |
| 2 | PU_ct | ElasticNet | {'alpha': 1.0, 'l1_ratio': 0.9} | 68.3874 | 55.6245 | 7.4582 | 76.6389 |
| 3 | PU_ct | KNN | {'n_neighbors': 7} | 83.3031 | 63.4082 | 7.9629 | 76.6389 |
| 4 | PU_ct | RegTree | {'max_depth': 3} | 66.1914 | 139.7370 | 11.8210 | 76.5556 |
| 5 | PU_ct | RF | {'max_depth': 5, 'n_estimators': 50} | 59.8383 | 74.5280 | 8.6330 | 76.4722 |
| 6 | PU_ct | GB | {'learning_rate': 0.01, 'n_estimators': 100} | 67.7984 | 82.2332 | 9.0683 | 76.6389 |
| 7 | PU_ct | NN | {'alpha': 0.001, 'hidden_layer_sizes': (50,)} | 108.0067 | 89.8061 | 9.4766 | 76.8611 |
| 8 | DO_ct | Lasso | {'alpha': 10.0} | 68.9586 | 69.3949 | 8.3304 | 76.6389 |
| 9 | DO_ct | Ridge | {'alpha': 10.0} | 70.3096 | 69.5455 | 8.3394 | 76.6389 |
| 10 | DO_ct | ElasticNet | {'alpha': 1.0, 'l1_ratio': 0.9} | 70.1235 | 69.5187 | 8.3378 | 76.6389 |
| 11 | DO_ct | KNN | {'n_neighbors': 7} | 90.3745 | 64.6264 | 8.0391 | 76.6389 |
| 12 | DO_ct | RegTree | {'max_depth': 3} | 66.4622 | 185.2506 | 13.6107 | 76.5556 |
| 13 | DO_ct | RF | {'max_depth': 10, 'n_estimators': 50} | 71.6752 | 87.9560 | 9.3785 | 76.4722 |
| 14 | DO_ct | GB | {'learning_rate': 0.01, 'n_estimators': 100} | 70.7593 | 100.4640 | 10.0232 | 76.6389 |
| 15 | DO_ct | NN | {'alpha': 0.0001, 'hidden_layer_sizes': (50,)} | 174.8957 | 81.8196 | 9.0454 | 76.8611 |

This table summarizes the performance of various regression models tested on two key company targets — PU_ct and DO_ct. Each row represents a different machine learning model along with its best hyperparameters and performance metrics. The models were evaluated using cross-validation (Best CV MSE), and tested for generalization on a hold-out dataset (Test MSE, Test RMSE). For predicting PU_ct, the Ridge Regression and ElasticNet models achieved the lowest Test MSE and RMSE values (~55.6 and ~7.45), indicating strong predictive performance with minimal overfitting. In contrast, the Regression Tree and Neural Network models underperformed, showing higher error and variability, suggesting they are less reliable for this target. For DO_ct, although no model reached very low error, KNN showed relatively better Test MSE (~64.6), but Lasso, Ridge, and ElasticNet maintained more consistent and stable RMSEs around 8.3, making them preferable for interpretability and generalization. Across both targets, Random Forests and Gradient Boosting provided mixed results slightly higher RMSEs but could be considered if explainability is not a priority. Notably, the Neural Networks resulted in the highest errors for both targets, suggesting that simpler models with regularization (like Ridge or Lasso) perform more robustly for your data. Given the minimal variation in the out-of-sample cost metric across models, the final recommendation should prioritize model interpretability, consistency across test performance, and ease of deployment, making Ridge or ElasticNet the top candidates.


Model Comparison - Root Mean Squared Error
RMSE PU_ct
RMSE DO_ct
RMSE
0
2
4
6
8
10
12
14
Lasso
Ridge
ElasticNet
KNN
RegTree
RF
GB
NN


Model Comparison - Mean Squared Error
MSE PU_ct
MSE DO_ct
MSE
0
25
50
75
100
125
150
175
Lasso
Ridge
ElasticNet
KNN
RegTree
RF
GB
NN

## Section 6: Train Final Model

| | Model | MSE (PU_ct) | RMSE (PU_ct) | MAPE (PU_ct) | R² (PU_ct) | MSE (DO_ct) | RMSE (DO_ct) | MAPE (DO_ct) | R² (DO_ct) | Avg Cost |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Ridge Regression | 28.2787 | 5.3178 | 0.0908 | 0.7768 | 25.5844 | 5.0581 | 0.1447 | -0.3612 | 141.617 |
| 1 | LASSO | 19.3200 | 4.3955 | 0.0708 | 0.8475 | 26.1709 | 5.1157 | 0.1326 | -0.3924 | 141.497 |

To identify the most effective model for deployment, we compared Ridge Regression and LASSO using multiple evaluation metrics. The goal was to optimize predictive accuracy for the primary target variable PU_ct, while also ensuring acceptable performance for DO_ct and maintaining cost efficiency. LASSO demonstrated superior performance for PU_ct, with a lower Mean Squared Error (MSE = 19.32), lower Root Mean Squared Error (RMSE = 4.40), lower Mean Absolute Percentage Error (MAPE = 7.08%), and a higher $R^2$ value (0.8475), indicating a stronger and more accurate fit to the data. Although Ridge Regression slightly outperformed LASSO in predicting DO_ct with marginally better MSE and RMSE, both models exhibited negative $R^2$ values for DO_ct, suggesting that the predictions for this target do not generalize well. Importantly, LASSO achieved the lowest overall average cost (141.50), aligning with business efficiency goals. Based on these results, LASSO offers the best trade-off between accuracy, interpretability, and cost, making it the preferred model for final deployment in our production pipeline.

## Section 7: Conclusion and Recommendations

### Key Insights:

This analysis focused on predicting the daily number of pickups (`PU_ct`) and drop-offs (`DO_ct`) at the Capital Bikeshare station "22nd & H St NW," using historical trip and weather data from February to April 2024. The feature engineering process integrated weather variables (like temperature, windspeed, and visibility) with station activity to build explanatory models. Among various algorithms tested, Ridge Regression and LASSO showed the most promise. LASSO significantly outperformed Ridge on the main target variable (`PU_ct`), achieving lower prediction error (MSE = 19.32, RMSE = 4.40) and stronger model fit ($R^2$ = 0.85). Although Ridge performed slightly better on `DO_ct`, both models demonstrated limited predictive ability for drop-offs (with negative $R^2$). Importantly, both models achieved similar out-of-sample cost efficiency, with LASSO yielding the lowest average cost.

### Conclusion:

This project set out to forecast daily bike pickup (PU_ct) and drop-off (DO_ct) demand at the Capital Bikeshare station “22nd & H St NW” using historical trip and weather data. After evaluating a range of predictive models—including linear, regularized, tree-based, and ensemble methods—LASSO Regression emerged as the top performer for the primary target, PU_ct. It delivered the lowest prediction error and the highest explanatory power ($R^2$ = 0.85), while also maintaining the lowest average out-of-sample cost. Although Ridge Regression showed slightly better predictive metrics for DO_ct, its overall benefit was limited due to negative $R^2$ values. Tree-based and complex non-linear models underperformed both in accuracy and interpretability. Overall, regularized linear models, particularly LASSO, provided the most reliable and cost-effective balance between prediction quality and operational decision support.

### Recommendations:

In addition to deploying the LASSO model for daily pickup forecasting, the company should consider integrating this predictive system into a broader operational dashboard that allows planners to visualize demand trends, weather

dependencies, and capacity needs in real time. Regular retraining of the model with updated data is essential to maintain accuracy, especially during seasonal shifts or behavioral changes in rider patterns. To address the weaker performance in predicting drop-offs (`DO_ct`), we recommend incorporating temporal features such as day-of-week, time-of-day segmentation, and public event calendars, which could better capture fluctuations in return behavior. Moreover, collecting and analyzing user-level data such as subscriber vs. casual rider behavior could offer deeper insights for segmentation-based modeling. Finally, the company should consider piloting these models across multiple stations with similar usage patterns to assess scalability and operational impact at a network level. By building on these recommendations, Capital Bikeshare can move toward a data-driven rebalancing strategy that enhances rider satisfaction and operational efficiency system-wide.