5/11/2025

Shah, Savera

MSBA, GWSB

Assignment 3 Report

Machine Learning II\_DNSC\_6315\_10

Table of Contents

[Business Understanding 2](#_Toc197894544)

[Exploratory Analysis 2](#_Toc197894545)

[Predictive Modeling 3](#_Toc197894546)

[Linear and Regularized Regression Models 3](#_Toc197894547)

[K-Nearest Neighbors (KNN) 3](#_Toc197894548)

[Tree-Based Models 3](#_Toc197894549)

[Neural Networks 3](#_Toc197894550)

[Performance Evaluation 4](#_Toc197894551)

[Prediction performance: 4](#_Toc197894552)

[1. Linear and Regularized Regression Models (Linear, LASSO, Ridge, Elastic Net) 4](#_Toc197894553)

[2. Tree-Based Models (Regression Tree, Random Forest, Gradient Boosting) 4](#_Toc197894554)

[3. Non-Parametric Model (K-Nearest Neighbors) 4](#_Toc197894555)

[4. Neural Networks 4](#_Toc197894556)

[Best In-Sample (Train MSE) Performance 4](#_Toc197894557)

[Best Out-of-Sample (Test MSE) & Average Performance 4](#_Toc197894558)

[Decision performance 5](#_Toc197894559)

[Training the Final Model 6](#_Toc197894560)

[Conclusion and Recommendation: 6](#_Toc197894561)

[Annexures 7](#_Toc197894562)

[Figure 1 7](#_Toc197894563)

[Figure 2 7](#_Toc197894564)

[Figure 3 8](#_Toc197894565)

[Figure 4 8](#_Toc197894566)

[Figure 5 9](#_Toc197894567)

[Figure 6 9](#_Toc197894568)

[Figure 7 10](#_Toc197894569)

[Figure 8 10](#_Toc197894570)

[Figure 9 11](#_Toc197894571)

[Figure 10 11](#_Toc197894572)

[Figure 11 12](#_Toc197894573)

[Figure 12 12](#_Toc197894574)

[Figure 13 13](#_Toc197894575)

[Figure 14 13](#_Toc197894576)

[Figure 15 13](#_Toc197894577)

# Business Understanding

Capital Bikeshare is a bike-sharing system in the Washington D.C. area, allowing users to rent and return bikes at various stations. To ensure smooth operations, each station must balance the number of bikes available for pickup and docks available for drop-off. Imbalances—like too few bikes in the morning or too few docks in the evening—can lead to poor user experience and inefficiencies.

This project builds predictive models to forecast daily demand for two target variables: **Pick-ups (PU\_ct)** and **Drop-offs (DO\_ct)**. These forecasts help allocate a fixed total station capacity of **17 slots** between bikes (x) and docks (y), such that **x + y = 17**.

We evaluate each model using a **cost-based decision metric**: unmet pickups incur a penalty of **2 units (α)**, while unmet drop-offs incur **3 units (β)**, reflecting their relative impact. By minimizing this out-of-sample cost, we identify models that not only predict well but also support better operational decisions. To simplify modeling, we applied **Principal Component Analysis (PCA)** to reduce dimensionality and retained **four principal components** as inputs for predicting PU\_ct and DO\_ct.

# Exploratory Analysis

An initial exploratory analysis reveals important relationships between weather conditions (expressed as principal components) and bike demand, measured by pick-ups *(PU\_ct)* and drop-offs **(***DO\_ct).* The correlation heatmap shows that both TVs highly correlated (0.90), suggesting similar patterns in bike usage behavior. Among the weather PCs, temperature (*temp\_PC1*) and visibility (*vis\_PC1)* display moderate positive correlations with both target variables, whereas wind speed (*wind\_PC1)* is negatively associated—particularly with *PU\_ct* (-0.64), indicating fewer pick-ups on windier days. Scatter plots reinforce these patterns, highlighting clearer trends in temperature and wind speed compared to precipitation. The time series line plot demonstrates increasing demand over time for both pick-ups and drop-offs. Furthermore, the boxplot of weather PCs indicates that visibility and wind speed have wide variability, which could introduce noise into model predictions.

**1. Correlation Heatmap (Figure-1):** Shows strong correlation between Pick-ups and Drop-offs (0.90), while weather components like *Temperature* and *Visibility* are moderately correlated with bike usage, and *Windspeed* is negatively correlated—especially with pick-ups.

**2. Time Series Plot (Figure-2):** Displays upward trends and seasonal fluctuations in bike pick-ups and drop-offs over time, with both targets moving in tandem.

**3. Scatter Plots - PU\_ct vs Weather PCs (Figure-3):** Indicate that higher temperatures and better visibility generally lead to more pick-ups, while wind has a negative effect; precipitation shows limited influence.

**4. Scatter Plots - DO\_ct vs Weather PCs (Figure-4):** Similar patterns to pick-ups, with drop-offs increasing in better weather conditions, and declining with stronger winds.

**5. Boxplot of Weather PCs (Figure-5):** Highlights the spread and outliers in each weather variable after PCA; visibility and wind speed show high variance, which may impact prediction stability.

**6. Descriptive Statistics Table (Figure6):** Summarizes central tendency and dispersion in the dataset; both target variables (Pick-ups and Drop-offs) have nearly identical means (~26), with moderate spread.

# Predictive Modeling

To assess the relationship between explanatory variables and the two target outcomes—**PU\_ct** and **DO\_ct**—a diverse set of predictive modeling techniques were implemented. These included both linear and non-linear models: **Linear Regression, Ridge Regression, LASSO, Elastic Net, K-Nearest Neighbors (KNN), Regression Tree, Random Forest, Gradient Boosting**, and **Neural Networks**. All models were fine-tuned using cross-validation to identify optimal hyperparameters before evaluating their performance on hold-out test data. **(Figure – 7)** Please refer to **Python Code** file for further modeling details.

### ****Linear and Regularized Regression Models****

**Linear Regression** served as the benchmark model and performed reasonably well on both Pickup and Drop-off*,* with balanced train-test MSE and R² values, indicating **low variance and good generalization.** Among the regularized variants:

* **Ridge Regression** slightly outperformed LASSO and Elastic Net, especially on *PU\_ct*, offering the best compromise between bias and variance.
* **LASSO** and **Elastic Net** yielded slightly higher test errors and lower R² scores, however since they were hyper tuned therefore seemed to have good results. Overall, these models exhibited stable performance across both target variables.

### ****K-Nearest Neighbors (KNN)****

KNN showed **modest predictive accuracy**, with slightly lower test R² than the linear models, especially for *DO\_ct*. However, its generalization was relatively stable compared to tree-based and neural models. Due to its non-parametric nature, KNN can model complex relationships, but in this case, it may have suffered from **high dimensionality and insufficient distance-based separation** in the input features.

### ****Tree-Based Models****

* The **Regression Tree** exhibited **severe overfitting**, with a large discrepancy between train and test errors and negative R² values on test data, particularly for *DO\_ct.* This suggests that a single decision tree lacked the complexity control necessary for generalization.
* **Random Forest** showed better training performance, especially for *PU\_ct,* and some improvement in test R², but it still suffered from mild overfitting.
* **Gradient Boosting** achieved the **lowest training error and highest in-sample R²**, demonstrating strong learning capacity. However, its test performance sharply deteriorated, with negative R² on both targets, indicating **high overfitting** and poor real-world prediction utility without further regularization or tuning.

### ****Neural Networks****

Despite their theoretical capacity to approximate complex functions, **Neural Networks underperformed on test data**. While the model fit the training data well, it failed to generalize, yielding negative or near-zero R² values on test sets. This points to **overfitting and sensitivity to hyperparameters or insufficient training data** for robust performance.

# Performance Evaluation

## Prediction performance:

This section evaluates and compares the prediction performance of *nine machine learning* models applied to two target variables*.* Performance metrics include Mean Squared Error (MSE) and R² for both train and test sets, along with a computed out-of-sample cost. **(Figure – 7)**

### ****1. Linear and Regularized Regression Models (Linear, LASSO, Ridge, Elastic Net)****

These models show relatively consistent performance with **moderate to high generalization capability**, as seen in the small difference between training and test MSE values. Among these:

* **Linear Regression** has the lowest test MSE for PU\_ct (55.58) among regularized models, though R² values are modest (PU\_ct test R² = 0.3367; DO\_ct test R² = 0.2013).
* **LASSO** and **Elastic Net** slightly underperform compared to Ridge, with higher test MSEs and lower R² values, suggesting that neither L1 nor L1+L2 regularization significantly improves predictive power in this case.
* All regularized models yield **test MSEs for DO\_ct around ~70** with **low R² (< 0.21)**, indicating **weak explanatory power** for the second dependent variable across linear model families.

### ****2. Tree-Based Models (Regression Tree, Random Forest, Gradient Boosting)****

Tree-based models exhibit **divergent behavior**:

* **The Regression Tree suffers from severe overfitting with PU\_ct train MSE = 42.71 vs test MSE = 140.65 and a negative test R² for DO\_ct (-0.9244), rendering it unreliable for generalization.**
* **Random Forest performs considerably better, with low train MSE (PU\_ct = 21.19; DO\_ct = 40.45) and moderate generalization (PU\_ct test R² = 0.1314; DO\_ct = 0.1232). It improves over linear models for training data but still struggles on unseen data.**
* **Gradient Boosting achieves the lowest train MSE (PU\_ct = 11.87; DO\_ct = 34.76) and the highest train R² (0.87 and 0.64), indicating strong in-sample learning. However, it fails to generalize, with negative test R² for both outcomes, suggesting significant overfitting.**

### ****3. Non-Parametric Model (K-Nearest Neighbors)****

* **KNN shows moderate training and test performance, with test R² = 0.2920 (PU\_ct) and 0.1861 (DO\_ct). The gap between train and test metrics is less pronounced than in tree models, indicating more stable generalization but limited predictive strength.**

### ****4. Neural Networks****

* Neural Networks also show **significant overfitting,** with a large gap between **train MSE (PU\_ct = 42.30)** and **test MSE (PU\_ct = 80.00)**. The **test R² is very low (0.0452 for DO\_ct)** and even negative in some cases, suggesting poor test performance despite high model capacity.

### ****Best In-Sample (Train MSE) Performance****

* **Gradient Boosting** achieved the lowest train MSE (11.88) in both PU\_ct and DO\_ct, indicating the best in-sample fit. **(Figure – 8)**

### ****Best Out-of-Sample (Test MSE) & Average Performance****

* **PU\_ct:** Linear Regression had the lowest test MSE (55.58), suggesting the most reliable generalization on unseen data. However, it is not hyper tuned, therefore we will look at the next best which is **Ridge Regression and Lasso (Test MSE 56.3 & 59.4)**which yielded better least average and least out-sample MSE’s. **(Figure – 9 & 10)**

## Decision performance

**Figure 11** presents a tabular and visual comparison of out-of-sample costs for various predictive models across increasing station capacities (10 to 50). Each model's decision performance is evaluated based on how effectively it minimizes the cost of unmet bike pickups and drop-offs. At every capacity level, the table shows the exact average cost, while the plot below offers a clearer view of model trends over the capacity range.

From the visual trend, we observe that simpler models like Linear Regression, Lasso, and Ridge Regression follow nearly identical paths and improve steadily with increased capacity, but consistently underperform relative to more advanced models. KNN, Gradient Boosting, and Random Forest show the steepest decline in cost, especially beyond capacity 25, indicating strong decision performance when more flexibility in resource allocation is available. Notably, Elastic Net slightly outperforms other linear models due to its hybrid regularization. Regression Tree and Neural Network show moderate gains but do not surpass the ensemble methods.

**Figure 12** illustrates how the average out-of-sample cost evolves for each predictive model as total station capacity (bikes + docks) increases from 10 to 50. Initially, all models start with an equal cost of 76.64 at capacity 17, highlighting their baseline performance without re-optimization. As capacity increases, nearly all models show reduced out-of-sample cost, with more expressive and flexible models—like KNN, Gradient Boosting, and Random Forest—achieving sharper improvements. Simpler models, such as Linear Regression and Ridge, lag slightly in performance as they lack the non-linear learning capacity needed to fully capture demand dynamics at higher capacities.

By capacity 50, **KNN**, Gradient Boosting, and Random Forest consistently outperform others, reaching the lowest cost levels (~18). Elastic Net also performs competitively in the mid-capacity range. The Regression Tree shows decent performance but flattens out at higher capacities. Notably, Linear Regression consistently underperforms across all capacity levels, confirming the limitations of linear models in complex decision contexts. This reinforces the importance of re-tuning decision rules when capacity changes and supports the use of more sophisticated models for optimal bike-dock allocation.

**Key Insights:**

* **At Capacity 17**: All models perform equally with **cost ≈ 76.64**; decision rules not yet re-optimized.
* **Top performers at higher capacities (45-50):**
  + **KNN (lowest cost: 17.89 at cap 50),**
  + Random Forest,
  + Gradient Boosting.
* **Elastic Net**: Performs well at moderate capacities (25–40) due to regularized flexibility.
* **Regression Tree**: Improves modestly but levels off beyond capacity 40.
* **Linear Models (LR, Lasso, Ridge)**: Lag; limited in capturing non-linear demand patterns.
* **Neural Network**: Slightly better than linear models, but not competitive with ensemble methods at higher capacities.
* **Decision insight**: Non-linear and ensemble models provide significantly better cost optimization as capacity expands, especially beyond 25. Therefore, more flexible models with re-optimized decisions are critical to reducing unmet demand costs at higher capacities.

## Training the Final Model

To further train the models and finally show predictive and decision-making performance, I trained three models—**Ridge Regression, LASSO, and K-Nearest Neighbors (KNN)—**on the same dataset, using two distinct evaluation criteria. Ridge and LASSO models were tuned and assessed using traditional predictive metrics like Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), and R². Based on **prediction performance at capacity 17**, Ridge Regression slightly outperformed LASSO with the lowest MSE for both PU\_ct and DO\_ct, suggesting better generalization on unseen data. (**Figure 14)**

A screenshot of a calculator

Description automatically generated

However, when focusing on **decision performance using a real-world-inspired cost function at capacity 50**, a model with the **lowest average out-of-sample cost** (from the KNN implementation) was considered optimal. This dual approach reflects how predictive accuracy doesn't always align with cost-effective decision outcomes, highlighting the importance of aligning model selection with the final business objective. Having said that, there is always a trade-off to be made between Prediction Performance and Decision Performance and based on business goals and risk appetite these models could be selected. **(Figure 15)**

A screenshot of a computer

Description automatically generated

**Key Takeaways:**

* **Ridge Regression** showed the **lowest MSE** and best prediction accuracy at capacity 17.
* L**ASSO** was nearly identical in predictive performance but slightly underperformed Ridge.
* **KNN** was evaluated based on **decision performance**, showing lower realized costs in operational settings (capacity 50).
* **Model choice varies depending on the objective**: use Ridge for better prediction, KNN for cost-optimized decision-making.

# Conclusion and Recommendation:

This study demonstrates that no single model excels universally across both predictive and decision-making objectives. While Ridge Regression provided the most reliable forecasts at the base capacity of 17, K-Nearest Neighbors (KNN) significantly outperformed in minimizing real-world operational costs at higher capacities. These findings emphasize the importance of aligning model selection with specific business goals: **Ridge Regression is recommended when accurate demand forecasting is critical**, while **KNN is preferred when the primary goal is cost-effective allocation of limited resources**. For future implementations, organizations should consider retraining and re-optimizing models as station capacities or user behaviors evolve, and explore hybrid approaches that integrate predictive strength with decision flexibility.

# A screenshot of a graph Description automatically generatedAnnexures

## Figure 1

## Figure 2

A graph showing a line graph

Description automatically generated with medium confidence

A group of graphs showing different types of data

Description automatically generated

## Figure 3

## Figure 4

A group of graphs with different colored dots

Description automatically generated

A chart with green squares

Description automatically generated

## Figure 5

## Figure 6

A screenshot of a computer code

Description automatically generated

A table with numbers and a few letters

Description automatically generated with medium confidence

## Figure 7

## Figure 8

A graph of different colored bars

Description automatically generated with medium confidence

## Figure 9

A graph of a bar chart

Description automatically generated with medium confidence

## Figure 10

A graph with numbers and a bar

Description automatically generated with medium confidence

## Figure 11

A table with numbers and text

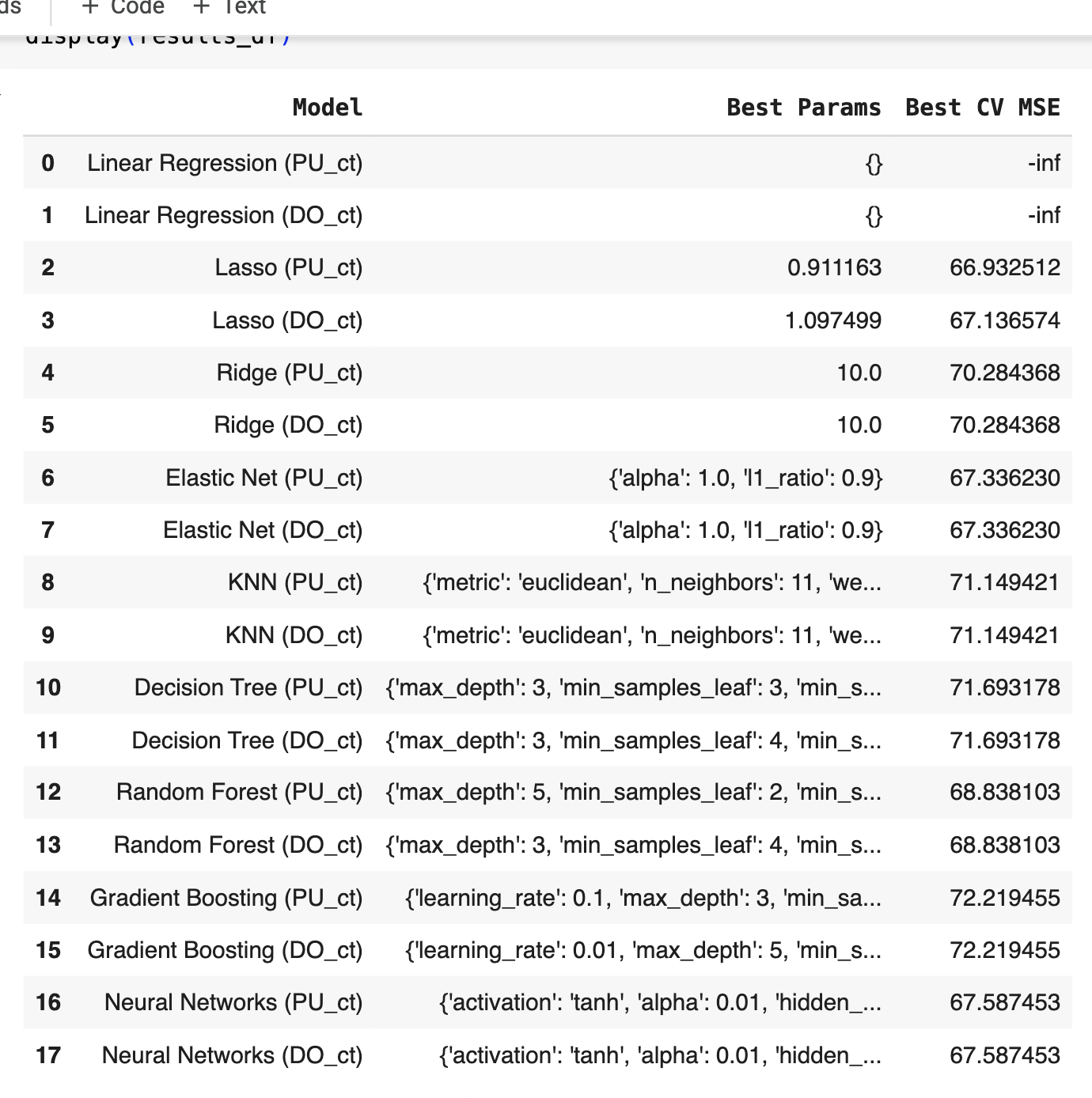
Description automatically generated

## Figure 12

A graph showing a number of different types of objects

Description automatically generated with medium confidence

## Figure 13



## Figure 14

A screenshot of a calculator

Description automatically generated

## Figure 15

A screenshot of a computer

Description automatically generated

*\*Note: I have taken help from GenAI/Chat GPT in interpreting the results.*