Mini Project Report

NYC Bicycle Traffic Analysis (Path 2)

Team Information

Name 1 (Leader): [Your Name]

Purdue Username 1: [username]

Name 2: [Partner Name]

Purdue Username 2: [partner_username]

Path Chosen: 2 (Bike Traffic)

Dataset Description

The nyc_bicycle_counts_2016.csv dataset contains daily bicycle traffic data collected from four major bridges in New York City between April 1, 2016 and October 31, 2016 (214 days total). The dataset includes:

- **Date and Day of Week:** Temporal identifiers for each observation
- **Weather Variables:** High Temperature (°F), Low Temperature (°F), and Precipitation (inches)
- **Bridge Traffic Counts:** Brooklyn Bridge, Manhattan Bridge, Williamsburg Bridge, and Queensboro Bridge
- **Total:** Combined bicycle count across all four bridges

The average daily traffic across all bridges was approximately 18,545 bicyclists, with significant variation based on weather and day of week.

Methods and Analyses

The analysis was implemented in Python following coding patterns established in course homework assignments (hw4.py for functional programming, hw10.py for ML classifiers, kmeans.py/cluster.py for data grouping, and gmm.py for model selection). The sklearn library was used for machine learning models.

Question 1: Bridge Sensor Selection

Problem: With budget for only 3 sensors, which bridges should be instrumented to best predict total traffic?

Method: I evaluated all four possible 3-bridge combinations using Linear Regression. For each combination, the three selected bridges served as features (X) and total traffic as the target (y). The R^2 score and RMSE were computed to measure prediction quality. This approach follows the model selection pattern from gmm.py's gaus_mixture function, which iterates through options to find the best configuration.

Justification: Linear regression is appropriate because total traffic is literally the sum of individual bridge counts, making the relationship inherently linear. R^2 directly measures variance explained, making it ideal for comparing combinations. I also computed correlation analysis to understand bridge interdependencies.

Question 2: Weather-Based Prediction

Problem: Can next-day weather forecasts predict bicycle traffic for enforcement planning?

Method: Following the get_model() and get_model_results() patterns from hw10.py, I implemented multiple regression models: Linear Regression, Ridge Regression (α=1.0), K-Nearest Neighbors Regression (k=5, k=10), and MLP Regressor. Features were standardized using StandardScaler. Data was split 80-20 for training/testing, and 5-fold cross-validation was performed for robustness assessment.

Justification: Testing multiple model types helps determine if the weather-traffic relationship is linear or requires non-linear approaches. KNN can capture local patterns while MLP can model complex non-linear relationships. Cross-validation helps assess model stability and prevent overfitting conclusions from a single train-test split.

Question 3: Day-of-Week Patterns

Problem: Can we identify weekly traffic patterns and predict the day from bridge counts?

Method: For pattern analysis, I computed mean and standard deviation of traffic for each day of week (similar to cluster centroid analysis from kmeans.py). For classification, I used the exact conf_matrix() function from hw10.py and tested multiple classifiers: KNN (k=3, k=5), Random Forest (100 trees), and MLP. The confusion matrix implementation was copied directly from hw10.py for consistency.

Justification: Classification is appropriate since days-of-week are categorical. Testing multiple classifiers follows the hw10.py pattern of comprehensive model evaluation. The 7-class problem (Monday-Sunday) is challenging, so comparing to the random baseline (14.3%) provides context for model performance.

Results

Question 1: Bridge Sensor Selection

Answer: Install sensors on Brooklyn Bridge, Manhattan Bridge, and Williamsburg Bridge. Exclude Queensboro Bridge.

All 3-bridge combinations ranked by R^2 score:

Bridge Combination	R^2 Score	RMSE	Excluded Bridge
Brooklyn, Manhattan, Williamsburg	**0.9970**	309.72	Queensboro
Brooklyn, Manhattan, Queensboro	0.9942	434.52	Williamsburg
Manhattan, Williamsburg, Queensboro	0.9873	641.02	Brooklyn
Brooklyn, Williamsburg, Queensboro	0.9798	809.19	Manhattan

Interpretation: The Brooklyn-Manhattan-Williamsburg combination achieves R^2=0.9970, explaining 99.7% of variance in total traffic with RMSE of only 309 bicyclists. Queensboro Bridge can be excluded because its traffic patterns are highly correlated with Williamsburg Bridge (r=0.953), making it redundant. The correlation analysis shows Williamsburg has the highest correlation with total traffic (0.975), followed by Queensboro (0.963), Manhattan (0.936), and Brooklyn (0.874). Including Brooklyn despite its lower individual correlation is important because it captures traffic patterns not fully represented by the other bridges.

Question 2: Weather-Based Prediction

Answer: Yes, weather can moderately predict bicycle traffic. The best model (KNN with k=5) achieved R^2=0.64.

Correlation with Total Traffic:

- High Temperature: r = 0.574 (moderate positive)
- Low Temperature: r = 0.442 (weak positive)
- Precipitation: r = -0.421 (moderate negative)

Model Performance (80-20 train-test split):

Model	R^2 Score	RMSE	MAE
KNN Regression (k=5)	**0.6397**	3,851	3,309
KNN Regression (k=10)	0.5968	4,074	3,550
Ridge Regression (α=1.0)	0.5757	4,180	3,666

Model	R^2 Score	RMSE	MAE
Linear Regression	0.5750	4,183	3,660

Interpretation: Weather explains about 64% of daily traffic variation. The best model's MAE of 3,309 represents an 18% error relative to the daily average of 18,545 bicyclists. This means police enforcement planning could reasonably use weather forecasts, but should expect predictions to be off by roughly 3,000-4,000 bicyclists on any given day. The non-linear KNN model outperformed linear models, suggesting the relationship between weather and biking behavior isn't strictly linear—perhaps there's a temperature sweet spot where ridership peaks.

Practical Recommendation: For enforcement planning, target days with high temperatures (70-85°F) and no precipitation—these conditions consistently correlate with peak traffic. Days with >0.5 inches of rain show dramatically reduced ridership.

Question 3: Day-of-Week Patterns

Weekly Pattern Finding: Yes, clear weekly patterns exist. Weekdays average 41% more traffic than weekends.

Average Daily Traffic by Day of Week:

Day	Avg Traffic	Std Dev	Type
Wednesday	22,422	±4,198	Weekday
Tuesday	20,782	±5,842	Weekday
Thursday	20,781	±5,033	Weekday
Monday	19,394	±5,253	Weekday
Friday	17,985	±5,388	Weekday
Saturday	15,001	±4,402	Weekend
Sunday	13,716	±4,141	Weekend

Classification Performance: Predicting the exact day from bridge counts is challenging. The best classifier (MLP) achieved only 27.9% accuracy—about 1.95x better than random guessing (14.3%), but not reliable enough for practical use.

Why Classification is Difficult: The large standard deviations (±4,000-6,000) relative to day-to-day differences (~2,000-3,000) create significant overlap between classes. Weather variation within a day of week is often larger than differences between days. The model can generally distinguish weekdays from weekends, but struggles to differentiate Tuesday from Thursday, for example.

Visualizations

Figure 1 shows: (top-left) bridge traffic correlations, (top-right) temperature vs. traffic with precipitation coloring, (bottom-left) weekly traffic patterns, and (bottom-right) normalized bridge usage by day.

Code Implementation Notes

The analysis code follows patterns established in course homework assignments:

- **From hw10.py:** get_model() function pattern for creating classifiers, get_model_results() pattern for evaluation, and the exact conf_matrix() implementation
- **From gmm.py:** Model selection loop pattern (iterating through options to find best), feature concatenation approach
- **From hw4.py:** Functional programming style with helper functions, combinations generation
- **From kmeans.py/cluster.py:** Data grouping concepts applied to day-of-week pattern analysis

No significant deviations from the provided coding styles were necessary. The primary extensions were using sklearn's regression models (LinearRegression, Ridge, KNeighborsRegressor) which follow the same fit/predict API patterns demonstrated in hw10.py with classifiers.