Statistic Machine Learning Project

Zhiwei Y. *zhiwei.yan@angstrom.uu.se

December 8, 2025

Abstract

To fulfill the demand of public bike that more fossil based transportation cans be replaced by bike and contribute to alleviating climate change, the city of Washington D.C. has recorded the observations high bike demand along with temporal and meteorological features. With this data, this project study is dedicated to help the city to predict the necessity of increasing number of bikes at certain hours. To achieve this goal, methods including Logistic Regression, KNN, bagging and boosting are deployed to predict the target label with given feature. Number of group members: 3.

1 Study case and Data

To further understand and practice the knowledge we have obtained from lectures. A case of classification problem was given that we can apply different methods on the data and understand the charactoristics and behaviours of these methods.

1.1 Case and data description

To help the city of Washington, D.C. understand whether increasing the number of public bikes is neccessary at some certain hour, a machine learning model is expected to predict the public bike demand for given temporal and meteorological features.

A data set contains 1600 random obeservations is provided for the model training. The target variable is binary for "high" or "low" demand for increasing bikes' number. The description of the features are given in Table 1

1.2 Exploratory data analysis

Expolratory data analysis has been conducted to gain the knowledge of relations among features in the dataset. Also, following questions in the project introduction are answered in this section.

*Deploying boosting method to given data and writing *Abstrect, Introduction, and Exploratory data analysis*

Table 1: Labels and features in the data set smlproject2024

Feature Name	**Description**
midrule increase_stock (prediction label)	**low_bike_demand** – no need to increase the number of bikes **high_bike_demand** – the number of bikes needs to be increased
midrule hour_of_day	Hour of the day (from 0 to 23)
day_of_week	Day of the week (from 0 – Monday to 6 – Sunday)
month	Month (from 0 – January to 12 – December)
holiday	If it is a holiday or not (0 – no holiday, 1 – holiday)
weekday	If it is a weekday or not (0 – weekend, 1 – weekday)
summertime	If it is summertime or not (0 – no summertime, 1 – summertime)
temp	Temperature in Celsius degrees
dew	Dew point in Celsius degrees
humidity	Relative humidity (percentage)
precip	Precipitation in mm
snow	Amount of snow in the last hour in mm
snow_depth	Accumulated amount of snow in mm
windspeed	Wind speed in km/h
cloudcover	Percentage of the city covered in clouds
visibility	Distance in km at which objects or landmarks can be clearly seen and identified

1. Which are the numerical features and which are the categorical features?
2. Is there any trend to need increase in the availability of bycycles?

Out of that the ways to treat categorical features are different from numerical features, the features are sorted into two groupd, numerical and categorical. This sort of them can be identified according to Table 1. The two groups are shown in Table 2, which also answers Quesiton 1.

For categorical features, first step done was to see the label balance. The result is shown in Figure 1. One thing to be noted is that the feature "snow" has only one label. A flat feature will not have input to the model, thus it was excloded from training set. This will be also seen in the correlation analysis. Similarily, histgram plot was generated to visualize the distribution of numerical features (shown in Figure 2).

To understand the correlation among features, mostly between targets and other features, correlation analysis has been done ploting corrolation maps, shown in Figure 3

Droping out the smaller correlated features ($correlation \geq 0.1$), the left ones are: "temp", "humidity", "hour_of_day", "summertime", "dew", "weekday", and "visibility". To more intuitively see if any features can help seperate the target label. Figure 4 was ploted.

Table 2: Categorical and numerical features.

Categorical features	**Numerical features**
holiday	month
weekday	temp
summertime	dew
snow	humidity
increase_stock	precip
hour_of_day	snow_depth
day_of_week	windspeed
	cloudcover

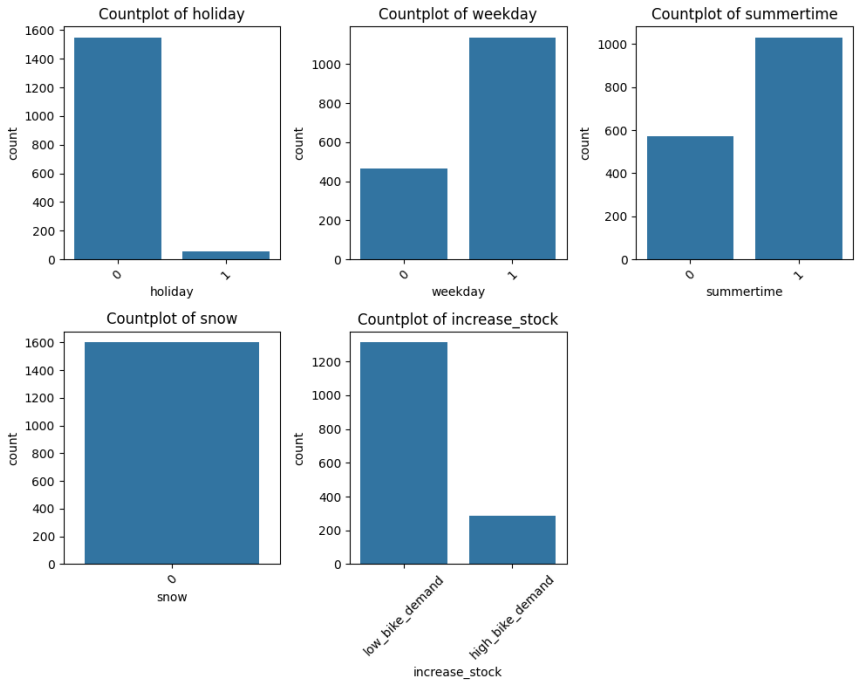

Figure 1: Label counts for categorical features

Finally, with all analysis and figures above, the Question 2 can be answered. It is seen that the label "1" ("high_bike_demand") is concerntrated around daytime, growth starts from early morning, drops till late evening after reaching its peak at 15:00 to 16:00. Weekday has more "high_bike_demand" than weekends.

Temperature also has impact on bike demand, more bike are needed when temperture is in a comfortable region, 20 30 degC in data. This can also

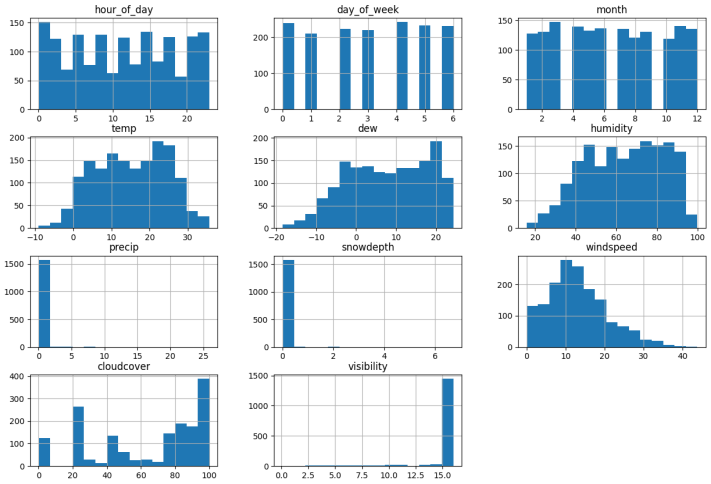

Figure 2: histgrams of numerical features

interprets the trend in the summertime, temperature is higher in summer, similarily for dew point.

2 Methodology

2.1 Machine learning models

2.1.1 Boosting

Boosting is an emsenble method which is built on the idea that even a week model can capture some patterns between target variables and given features. Starting from a base model, optmizing it based on the returned error can produce a new model which would become the base for next model. By ensembling the predictions from these models, the intention of boosting is to reduce biasLindholm2022sml.

In this study, three maintream boosting algorithm were deployed, **AdaBoost**Lindholm2022sml, **GradientBoost**Lindholm2022sml, and **CatBoost**prokhorenkova2018catboost. The methods can expressed as following equations.

$$\text{AdaBoost:} \quad \hat{y}^{(B)}_{\text{boost}}(\mathbf{x}) = \text{sign}\left(\sum_{b=1}^{B} \alpha^{(b)} \hat{y}^{(b)}(\mathbf{x})\right) \tag{1}$$

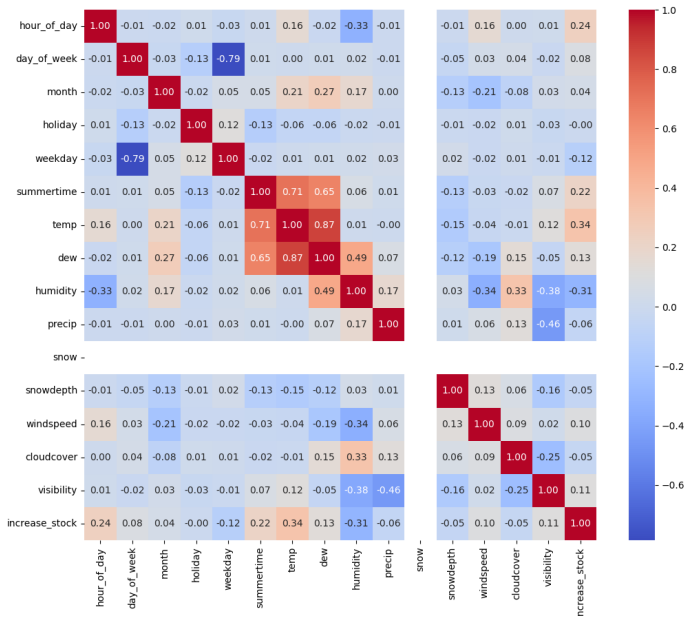

Figure 3: Corrolation among features

$$\text{GradientBoost:} \quad f^{(B)}(\mathbf{x}) = \sum_{b=1}^{B} \alpha^{(b)} f^{(b)}(\mathbf{x}), \tag{2}$$

$$\text{CatBoost:} \quad f^{(B)}(\mathbf{x}) = \sum_{b=1}^{B} \eta^{(b)} T^{(b)}(\mathbf{x}) \tag{3}$$

2.1.2 Logistic Regression

Logistic Regression is one of the most common supervised machine learning models for binary classification problems. Unlike Linear Regression, it uses a sigmoid function, where the raw output value is passed through and converted into a probability between 0 and 1. Since our problem has two possible outcomes, this model is specifically a binomial logistic regression.

Logistic Regression first calculates the linear combination of the input features:

$$z = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$$

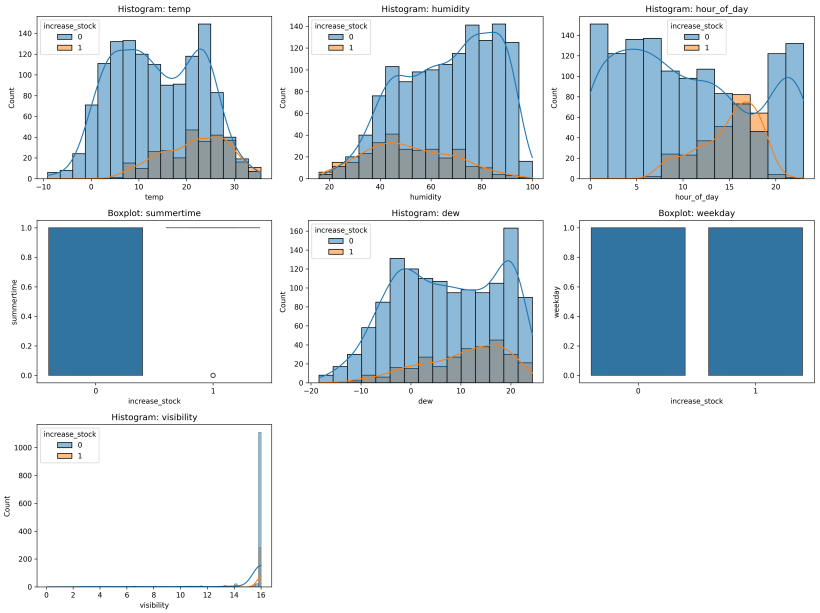

Figure 4: Boxs and Histgrams for features with different targets laebls

Here, $x_1, x_2, \ldots, x_n$ are the input feature values, $w_0, w_1, \ldots, w_n$ are the model's learned weights, and z is the model's internal score. This value is then passed through the sigmoid function to convert it into a probability between 0 and 1.

2. K-Nearest Neighbours

K-Nearest Neighbours (KNN) is a supervised machine learning algorithm used for both classification and regression problems. It works by finding the K nearest points to a given data point and making predictions based on the majority class for classification or the average value for regression.

KNN is a simple, non-parametric, and lazy learning algorithm because it performs computation during prediction rather than during training.

The distance between two data points, $\mathbf{x}_1$ and $\mathbf{x}_2$, is measured using the Euclidean distance formula:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{(x_{11} - x_{21})^2 + (x_{12} - x_{22})^2 + \cdots + (x_{1n} - x_{2n})^2} \tag{4}$$

For a new data point $\mathbf{X}_{\text{new}}$, the algorithm calculates its distance from all instances in the dataset and selects the K instances with the smallest distances. The final predicted class is determined based on the majority vote among these K nearest neighbours.

It is important to note that if K is too small, the model becomes overfitted. If K is too large, the model becomes underfitted. In our project, we selected the optimal K value using the `GridSearchCV` function.

2.2 Validation

K-fold method with 5 subset was used for cross-validation to the deployed models. Considering that label are imbalanced in the target variable, metrics like accuracy would not be a good choice for perfomance measurement. F-score, given by equation 5, is used for measuring the perfomance in this case.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{2TP}{2TP + FP + FN} \tag{5}$$

3 Results

3.1 Boosting

Three boosting models were apply in this study, AdaBoostingClassifier and GradientBoostingClassifier from *Scikit-learn*,and CatBoostClassifier. Their perfomance is shown in Table 3

Table 3: Boosting models performance

Model	**Precision**		**Recall**		**F1 Score**		**Train Time (s)**
	1	**0**	**1**	**0**	**1**	**0**	
AdaBoostClassifier	0.6427	0.9161	0.6298	0.9184	0.6334	0.9170	0.0035
GradientBoostingClassifier	0.7729	0.9238	0.6555	0.9549	0.7084	0.9391	0.0890
CatBoostClassifier	0.7828	0.9317	0.6932	0.9549	0.7337	0.9431	0.5812

According to the perfomance, CatBoostClassifier was chosen to the model to be Optimized on Hyperparameters, where *GridSearch* was used. The result is shown in Table 4.

Table 4: f1-score of best models

Class	**Cat Boost**	**KNN**	**Logistic Regression**	**Support**
0	0.93	0.92	0.94	270
1	0.61	0.57	0.62	50