
Do we need more bikes?

Project in Statistical Machine Learning

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 In this project we develop, and study different statistical machine learning models
2 for predicting whether the number of available bikes at a given hour should be
3 increased, a project by the District Department of Transportation in Washington
4 D.C. The training data set consists of 1600 instances of hourly bike rentals, and
5 a test set of 400 instances. The models for prediction we have used are: *Logistic*
6 *regression*, *Discriminant methods: LDA, QDA, k- Nearest Neighbour*, and *Tree*
7 *Based Methods*. We have found that THE MODEL gives best prediction, with
8 accuracy ??????

9 **1 Plan**

10 **1.1 From Intro**

11 (i) Explore and preprocess data

12 (ii) try some or all classification methods, which are these?

13 • Logistic Regression

14 • Discriminant analysis: LDA, QDA

15 • K-nearest neighbor

16 • Tree-based methods: classification trees, random forests, bagging

17 • Boosting

18 (iii) Which of these are to be "put in production"?

19 **1.2 From Data analysis task**

20 • Can any trend be seen comparing different hours, weeks, months?

21 • Is there any difference between weekdays and holidays?

22 • Is there any trend depending on the weather?

23 **1.3 From Implementation of methods**

24 Each group member should implement one family each, who did what shall be clear!

25 DNNs are encouraged to be implemented, do this if there is time. (DNN is not a thing a group

26 member can claim as their family.)

27 Implement a naive version, let's do: *Always low_bike_demand*

28 **1.3.1 What to do with each method**

29 1. Implement the method (each person individually)

30 2. Tune hyper-parameters, discuss how this is done (each person individually)

31 3. Evaluate with for example cross-validation. Don't use E_{k-fold} (what is that?) (need to do

32 together)

33 4. (optional) Think about input features, are all relevant? (together)

34 Before training, unify pre-processing FOR ALL METHODS and choose ONE OR MULTIPLE

35 metrics to evaluate the model. (is it necessary to have the same for all?, is it beneficial?) Examples:

36 • accuracy

37 • f1-score

38 • recall

39 • precision

40 Use same test-train split for ALL MODELS

2 Introduction

Statistical machine learning is a subject that aims to build and train algorithms, that analyse large amount of data, and make predictions for the future, which are computed by using established statistical models, and tools from functional analysis. This is a project in supervised, statistical machine learning, where several models were created, and trained, in order to analyse which one of them gives best prediction for the project "Do we need more bikes", where we want to understand, and predict if there is a high, or low demand of city bikes in the public transportation of Washington, a project by the District Department of Transportation in Washington D.C..

The data set used for training our models, consist of 15 variables, containing quantitative/qualitative data. We developed several models, and evaluated them with cross-validation, in order to understand which algorithm gives the best prediction.

3 Data analysis

4 Theoretical Background

4.1 Mathematical Overview of the Models

4.1.1 Logistic Regression

The backbone of logistic regression is linear regression, i.e. finding the least-squares solution to an equation system

$$X\theta = y \quad (1)$$

given by the normal equations

$$X^T X \theta = X^T y \quad (2)$$

where X is the training data matrix, θ is the coefficient vector and b is the training output. The parameter vector is then used in the sigmoid function:

$$\sigma(z) = \frac{e^z}{1 + e^z} : \mathbb{R} \rightarrow [0, 1], \quad (3)$$

$$z = x^T \theta, \quad (4)$$

where x is the testing input. This gives a statistical interpretation of the input vector. In the case of a binary True/False classification, the value of the sigmoid function then determines the class.

4.1.2 Random forest

The random forest method is a based upon decision trees, i.e. dividing the data point into binary groups based on Gini-impurity, entropy or classification error, Gini being the most common. These divisions are then used to create a binary tree shown in figure ??Tree) and where thee leaf-nodes are used to classify the target variables bases on the input. As of itself the diction tree tends to have unsatisfying results which leads to methodes like random forest that boost its accuracy.

4.1.3 Non-parametric method: k-Nearest Neighbour

k -Nearest Neighbour(k -NN) is a distance based method that takes a k amount of points from the training data set, called *neighbours*, computes the distance between them, then assumes that the predicted value $\hat{y}(x_*)$ follows the trend of the k - nearest neighbours. Since k -NN uses the training data explicitly it is also called a *nonparametric* method.

The k -NN method can be divided into several subcategories, inter alias *classification* k -NN method, *regression* k -NN method. In this project, we are using the classification method, since we are trying to predict in which of the two classes low, or high demand, the given, and predicted data points belong.

The classification k -NN algorithm evaluates $\hat{y}(x_*)$ by computing the most frequently occurring class among the k nearest neighbours. Here, we try to identify whether a data point belong to the high

80 demand-class. Denote $c = \text{high demand class}$. For simplicity, assume Euclidean ambience. Then

$$\hat{y}(x_*) = \arg \max_c \sum_{n \in \mathbb{N}} \chi_{(y_i=c)},$$

81 where y_i is the class of the nearest neighbour, χ is the characteristic function

$$\chi_{(y_i=c)} = \begin{cases} 1 & \text{if } y_n = c, \\ 0 & \text{otherwise.} \end{cases}$$

82 It is very common to use a weighted sum to predict the next value, i.e.

$$\hat{y}(x_*) = \arg \max_c \sum_{n \in \mathbb{N}} \frac{\chi_{(y_n=c)}}{d(x, x_n)},$$

83 where d is the standard Euclidean metric, computing the distance between an input x , and a neighbour
84 x_n .

85 When using this model it is important to choose an optimal k -value. There are several tests for this,
86 here we implement *uniform weighting*, and *distance weighting*. The first algorithm creates a k -NN
87 model for each new $k \in [1, 500]$, and trains the model with uniform weights, i.e. the contribution of
88 all neighbours is equal. Similarly, the latter trains a k -NN classifier for each $k \in [1, 500]$, with the
89 difference that it uses distance based weighting, i.e. closer neighbours have greater influence. After
90 testing different upper boundaries for k , the two models gave good results in the interval $[1, 500]$, see
91 Figure ???. From the figures, we can see that the second test gives a better value for k , since the plot
92 follows smoother trend, in comparison to the uniform weighting test, which makes it easier to identify
93 an optimal k value ($k = 120$). Moreover, the distance weighting algorithm is providing results for
94 larger values of k , that is for $k \in [1, 400)$ before the curve converges, while the uniform weighting
95 algorithm converges earlier, when $k = 120$. This means that for large k , both test algorithms make
96 prediction based on the most common class in the data set, instead of making prediction based on the
97 behaviour of the neighbours. Thus for sufficiently large k , for any given data point, the model will
98 consider unnecessarily large amount of neighbours, and the prediction will be evaluated to belong to
99 the most frequent class. Since the distance weighting has a larger range of k -value, it should be more
100 trustworthy.

101 When $k = 120$, the accuracy of the model is 92%.

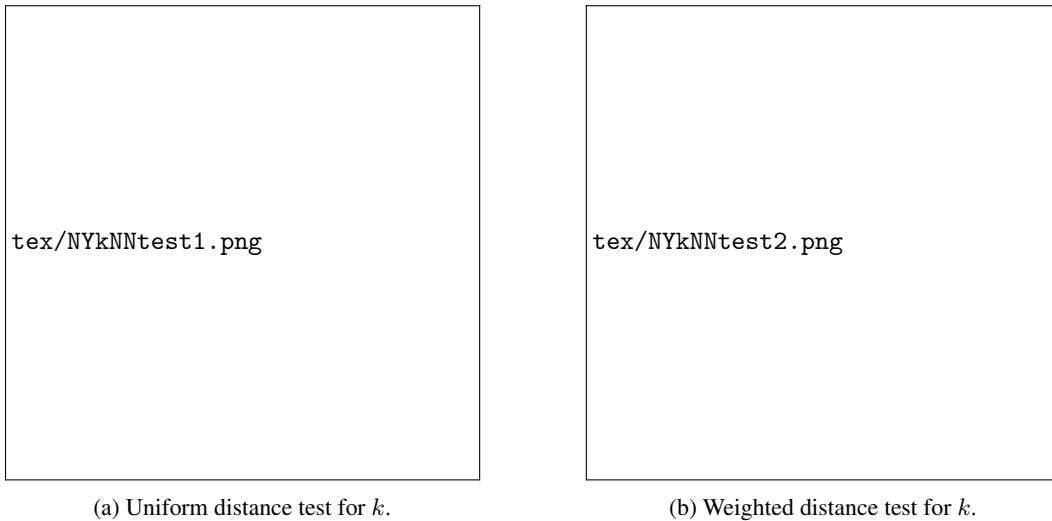


Figure 1: Test for choosing an optimal k -value.

102 4.1.4 Discriminant analysis: LDA and QDA

103 Linear Discriminant Analysis is a generative model, which means it is a model that's creating and
104 using a probability distribution $P(\mathbf{x}, y)$ to create an estimation for the probability $P(y = m|\mathbf{x})$ using

105 bayes theorem.
 106 Bayes theorem is:

$$p(y|\mathbf{x}) = \frac{p(y, \mathbf{x})}{p(\mathbf{x})} = \frac{p(y)p(\mathbf{x}|y)}{\int_y p(y, \mathbf{x})}$$

107 For the discrete version it is obtained:

$$p(y = m|\mathbf{x}) = \frac{p(y = m)p(\mathbf{x}|y = m)}{\sum_{m=1}^M p(y = m)p(\mathbf{x}|y = m)}$$

108 For this form of the equation to be useful, it is necessary to obtain an accurate estimation of $p(y = m)$
 109 and $p(\mathbf{x}|y = m)$ for all classes m .

110 In LDA, $p(y = m)$ is estimated by counting the percentage of data points (in the training data) being
 111 in each of the classes and using that percentage as the probability of a data point being in that class.

112 In mathematical terms:

$$p(y = m) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}\{y_i = m\} = \frac{n_m}{n}$$

113 To estimate the probability distribution $p(\mathbf{x}|y = m)$, a multi-dimensional gaussian distribution is
 114 used:

$$\mathcal{N}(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu)\right)$$

115 Where \mathbf{x} is the d-dimensional data point, μ is the (d-dimensional) mean of the random variable. Σ is
 116 the symmetric, positive definite covariance matrix defined by:

$$\Sigma = \frac{1}{n - M} \sum_{m=1}^M \sum_{i: y_i = m} (\mathbf{x}_i - \mu_m)(\mathbf{x}_i - \mu_m)^T$$

117 Using these estimations results in an expression for the quantity $p(y = m|\mathbf{x}) \forall m$. LDA then uses
 118 maximum likelihood to categorize an input \mathbf{x} into a class m .

119
 120 Quadratic discriminant analysis (QDA) is heavily based on LDA with the sole difference
 121 being how the covariance matrix Σ is created. In LDA, the covariance matrix is assumed to be the
 122 same for data in each and every class. In QDA however, the covariance matrix is calculated for each
 123 class as follows:

$$\Sigma_m = \frac{1}{n_m - 1} \sum_{i: y_i = m} (\mathbf{x}_i - \mu_m)(\mathbf{x}_i - \mu_m)^T$$

124 4.2 Input Data Modification

125 By plotting the data and analyzing the .csv file, some observations were made. The different inputs
 126 were then changed accordingly:

- 127 • *Kept as-is*: weekday, windspeed, visibility, temp
- 128 • *Modified*:
 - 129 – month - split into two inputs, one cosine and one sine part. This makes the new inputs
 - 130 linear and can follow the fluctuations of the year. The original input was discarded.
 - 131 – hour_of_day - split into three boolean variables: demand_day, demand_evening,
 - 132 and demand_night, reflecting if the time was between 08-14, 15-19 or 20-07 respec-
 - 133 tively. This was done because plotting the data showed three different plateaus of
 - 134 demand for the different time intervals. The original input was discarded.
 - 135 – snowdepth, precip were transformed into booleans, reflecting if it was raining or
 - 136 if there was snow on the ground or not. This was done as there were no times where
 - 137 demand was high when it was raining or when there was snow on the ground.
- 138 • *Removed*: cloudcover, day_of_week, snow, dew, holiday, summertime. These were
- 139 removed due to being redundant (e.g. summertime), not showing a clear trend (e.g.
- 140 cloudcover), giving a worse score when used, or all three (e.g. day_of_week).

141 5 Data Analysis

142 In the given data, there are some numerical and categorical features:

- 143 • *Numerical*: temp, dew, humidity, precip, snow, snowdepth, windspeed, cloudcover
144 and visibility.
- 145 • *Categorical*: hour_of_day, day_of_week, month, holiday, weekday, summertime, and
146 increase_stock

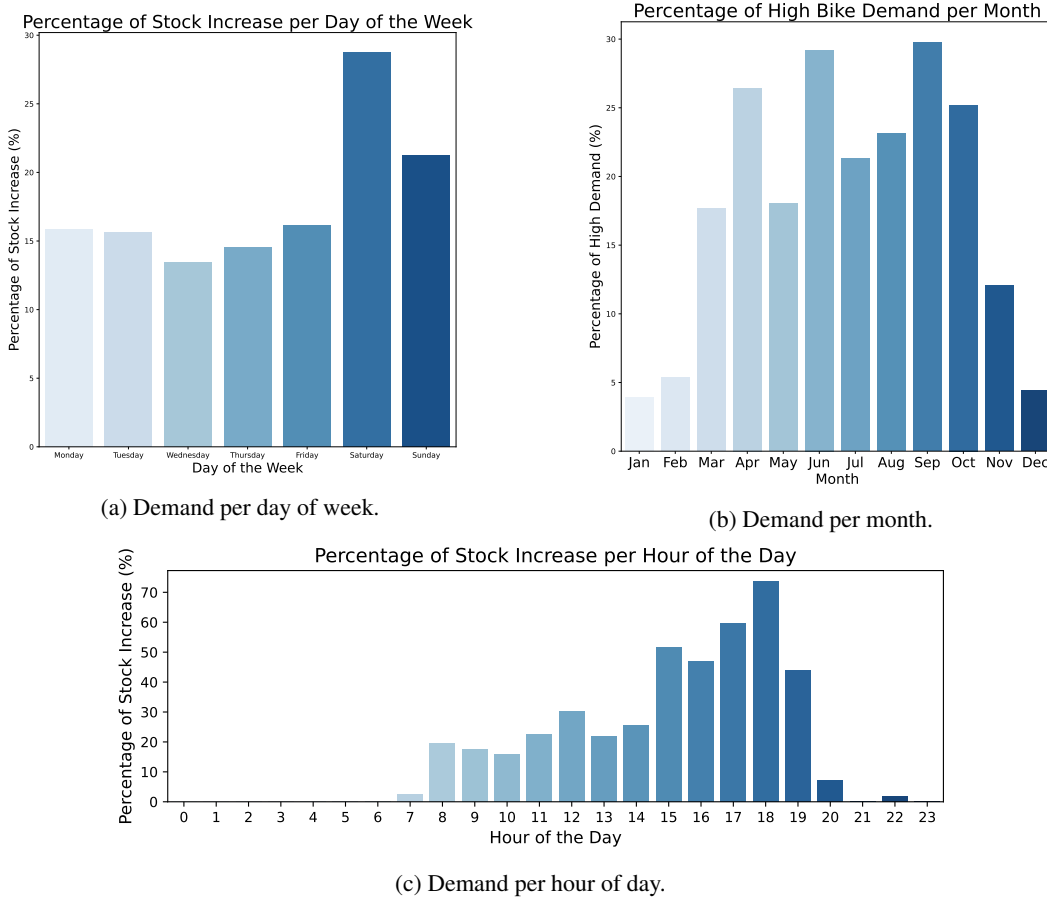
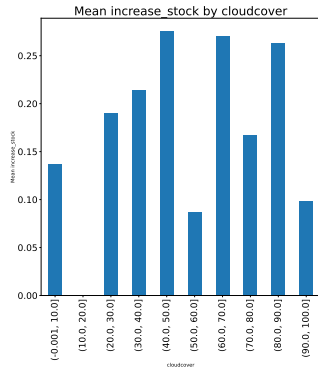


Figure 2: Bike demand vs. day of week and month.

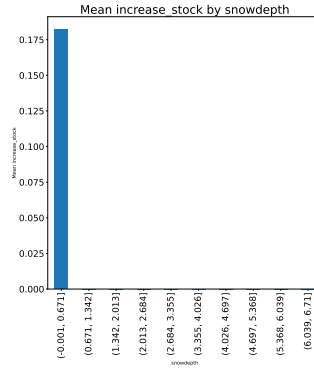
147 There are some trends seen in the data when it comes to time and weather. From figure ??, one can
148 see a periodic relationship for the months, where there is a higher demand during the warmer months,
149 loosely following a trigonometric curve. Over the week, the demand is rather stable, with a peak on
150 the weekend, especially Saturdays.

151 Looking at the weather (figure ??); if there is rain or if there is snow on the ground, there is close
152 to always low demand. Cloudcover did not make a big impact, which is also intuitive, as a cloudy
153 day does not make biking more difficult. Dew point also does not have a clear trend, while humidity
154 however has a clear trend downwards as the humidity increases. Temperature had a more clear impact,
155 where more people wanted to bike the warmer it got.

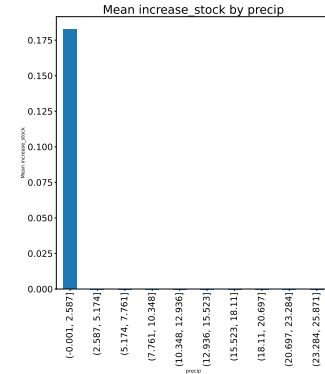
156 The overall trend is that about one eighth of observations correspond to a high bike demand. During the
157 night, or in bad weather, the demand is (intuitively) low. But during rush hour (figure ??), the demand
158 is very high, and should probably be increased in order to minimize excessive CO₂ emissions.



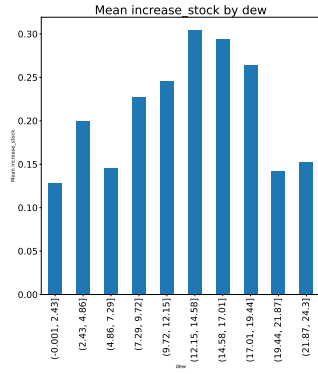
(a) Demand per cloudcover (percentage).



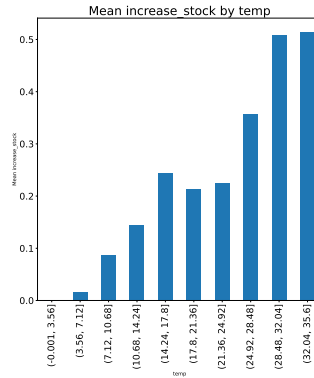
(b) Demand per day of week.



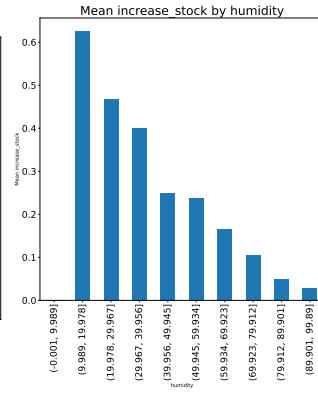
(c) Demand per day of week.



(d) Demand per dew point ($^{\circ}\text{C}$).



(e) Demand per temperature ($^{\circ}$).



(f) Demand per humidity level (percentage).

Figure 3: Bike demand vs. various weather parameters.