# Do we need more bikes?
## Project in Machine Learning

**November, 2024**

Department of Information Technology

Uppsala University

### Abstract

This document contains the instructions for the project on classification for the course Statistical Machine Learning, 1RT700. The problem is to predict whether the District Department of Transportation in Washington D.C. should increase the number of bikes available in the public bicycle-sharing system at a particular hour or not based on various factors such as date and time, weather conditions, temperature, and other relevant features, as a binary classification task. The training set comprises 1600 instances of hourly bike rental features and you will later be given a test set of 400 instances. You are expected to (i) explore and preprocess the data, (ii) try some (or all) classification methods from the course and evaluate their performance on the problem, and (iii) make a decision about which one to use and "put in production" against a test set. Your final prediction will be evaluated and also compared to the performances of the other student groups. You will document your project by writing a report, which will be reviewed anonymously by your peers.

## 1 Workflow

Before you start with the project, you should **read the whole instructions carefully**. Below we outline the general workflow of the project. For details of each part, see the specific sections.

1. Work on your report:
   - You work on your project in groups of 3-4 students.
   - Check out the problem formulation and the provided data set described in Section 2.
   - Answer the questions of the data analysis task in Section 3.
   - Implement and tune the methods as described in all subsections of Section 4.
   - When writing your report, follow the documentation guidelines in Section 5.
2. First submission:
   - As a group you submit your report.
   - After the report submission each student gets assigned to peer review one report from another group. Read the report and make useful comments on how to improve it. See Section 6.
3. Second submission:
   - Every group has received multiple feedback from the peer review and has to revise their report. Re-upload the revised version of your report. See Section 7.
   - Upload a contribution statement, see Section 7.2.
   - Upload your model predictions, see Section 7.3.
   - After the deadline your submissions will be graded by the teachers.

4. Third submission (Section 8):
    - If your report grade after the second submission by the teachers is a revise, then you have to update your report according to the feedback and re-upload it.
    - If you failed the peer review (no submission or too little/few/not useful comments), you get assigned a second additional peer review, and failing the second peer-review submission immediately leads to failure of the complete project.

All tasks described in this document have to be done in order to pass the project and of course *all group members have to take part in the project*.

There are deadlines for each part of this workflow. For the exact dates, see the course web page on Studium.

## 2 Problem and data

Capital Bikeshare is a 24-hour public bicycle-sharing system that serves Washington, D.C., and offers transportation for thousands of people throughout the city. The problem that arises is that there are certain occasions when, due to various circumstances, there are not as many bikes available as there are demands. In the long term, this situation will result in more people taking the car instead of the bicycle, increasing $CO_2$ emissions in the city. To tackle this situation, the District Department of Transportation in the city wants to know if at certain hours an increase in the number of bikes available will be necessary.

The goal of the project is to predict whether an increase in the number of bikes is necessary or not based on various temporal and meteorological data. You are expected to use all the knowledge that you have acquired in the course about classification algorithms, to come up with *one* model that you think is suited for this problem and which you decide to put "in production". This model will then be tested against a test set made available after peer review.

### 2.1 Data set

The training data set `training.csv` consists of 1600 randomly selected observations over the past 3 years in the city of Washington. D.C. See the first row in Table 1 to understand how the output feature is defined. This output feature has to be predicted from a number of features, see the remaining rows in Table 1.

## 3 Data analysis task

The first step–before we start to build a model–is always to take a closer look at the data by analyzing some statistics of the data set. In this step, you can already gain some insights into the data which can help you build your model and interpret your results later on.

Look into the provided data set by e.g. plotting the individual features. Based on your analysis, answer the following questions:

(i) Which are the numerical features and which are the categorical features?

(ii) Is there a greater trend to need an increase in the availability of bicycles? Study this question from various perspectives:
    - Can any trend be seen comparing different hours, weeks, and months?
    - Is there any difference between weekdays and holidays?
    - Is there any trend depending on the weather? Rainy days, snowy days, etc.

Write concise answers to each question and support your findings with evidence (statistics, plots, etc.). Discuss the results. Additionally, you can explore the correlation of features, outliers, range of values, and many more aspects.

Table 1: Available label (first row) and features (remaining rows) in the data set.

| Feature Name | Description |
|---|---|
| increase_stock | `low_bike_demand` - no need to increase the number of bikes |
| (prediction label) | `high_bike_demand` - the number of bikes needs to be increased |
| 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. |

# 4 Implementation of Methods

## 4.1 Methods to explore

The course has covered the four following families of classification methods:

(i) Logistic regression

(ii) Discriminant analysis: LDA, QDA

(iii) K-nearest neighbor

(iv) Tree-based methods: classification trees, random forests, bagging

(v) Boosting

In this project, you decide upon *at least* as many "families" as you are group members, and decide in each "family" *at least* one method to explore. To be clear, **each group member should independently implement and write about one of the list methods above.** Who implemented which method should later be clearly written in the contribution statement. All group members should be able to stand for all sections of the report.

Note, that the model family of deep neural networks (DNNs) is not a part of the required model families above as it is covered later in the course. You are welcome to explore DNNs, but each member still has to fulfill the minimum requirement of implementing and describing one of the model families listed above.

**Additionally**, you have to implement a naive classification model, which serves as a benchmark to evaluate the performance of your methods. One possibility can be a model that always predicts the `high bike demand` or `low bike demand` or random labels.

### 4.2 What to do with each method

**This task can be done individually.** For *each* method you decide to explore, you should do the following:

(a) Implement the method. We suggest that you use Python, and you may write your own code or use packages (the material from the problem-solving sessions can be useful).

(b) Tune the method to perform well. Discuss which methodological approach will be used for model tuning. For example grid search, random search, or similar.

(c) Evaluate its performance using, e.g., cross-validation. **Note** that selecting hyperparameters based on $E_{k-fold}$ will invalidate its use as an estimator of $E_{new}$. Therefore, the $E_{k-fold}$ error obtained in the tuning of a model cannot be used as an estimation of $E_{new}$ to compare different methods. For a detailed explanation see ***"Using a Test Dataset"*** in Section 4.2 of the course book.

(d) `[optional]` Think about the input features. Should you use all of them or only a subset or a combination of different features? **Note** that creating new features can increase the model's performance. Good examples would be a feature that indicates whether it is day or night, or a feature that indicates whether it is good weather or not.

### 4.3 Model selection and comparison

Before you start training, please unify data pre-processing steps for all methods and choose one or multiple metrics that you want to evaluate your model such as accuracy, f1-score, recall, precision, and many more.

Once you have developed the models (individually), it is time to compare the tuned models and identify *as a group* which model you consider the "best" one for the task. It is **important** that if you decide to separate a test set to compare the models, it must be the same for all the compared models. This can be done by setting a seed in the split. By comparing the models you will decide which model to use "in production" on the test set. This test set will be made available after the first submission. Write up the results together.

## 5 Documentation

Summarize your work by writing a report, which will be first peer-reviewed by your fellow student in the course (first submission date) before being graded by the teachers (second submission date).

### 5.1 What to include in your report

The report has to include the following:

(1) An abstract to summarize the problem and your findings.

(2) A brief introduction to the problem.

(3) Your data analysis including answers to the questions and plots.

(4) The model development, including:

  (a) A concise mathematical description of each of the considered methods, and how they are applied to the problem. You should *not* describe the code commands that you use but the underlying mathematical concept.

  (b) How the methods were applied to the data including motivations of the choices made. Here you can describe which inputs were used, if the inputs were considered as qualitative or quantitative, how parameters were tuned, etc.

  (c) Your evaluation of how well each method performs on the problem.

  (d) Model selection: Which method you decided to use "in production", and your arguments for your choice!

(5) A conclusions.

(6) Appropriate references.

(7) Main code to the appendix.

You can use (and reference) online sources, but use your own words. **Note:** Copy and paste from online sources or past reports will be identified in our plagiarism check and will lead to automatic failure and (in serious cases) reporting to the University's disciplinary board.

### 5.2 How to format your report

**Important points:**

- Use the template for the Neural Information Processing Systems (NeurIPS) conference with line numbers in its draft mode for the first submission and without line numbers for the second (final) submission.
- Include all points from the list in Section 5.1.
- Obey the page limit of 7 pages, excluding the appendix and references. Your report must be fully understandable within those 7 pages. This page limit includes the title and abstract. Do not include a table of contents. A violation of the page limit will not be accepted and the report has to be resubmitted.

**Details:**

- Your report has to be submitted as a PDF file following the style used for the prestigious machine learning conference NeurIPS, which also is the style used for this document. In the NeurIPS format, your report should be *no longer than 7 pages* not counting the reference list and code appendix. Except for the page limitation, you should follow the NeurIPS style closely, including its instructions for figures, tables, citations, etc.
- The report has to be written in LaTeX. You can access the LaTeX files from the conference webpage https://neurips.cc/Conferences/2022/PaperInformation/StyleFiles. Make sure to copy the style file and link it in your main file as done in the original template. If you prefer not to install a LaTeX compiler on your computer, you can use online services such as Overleaf (https://www.overleaf.com/). In your .tex-file, add the lines

  ```
  \makeatletter
  \renewcommand{\@noticestring}{}
  \makeatother
  ```

  before `begin{document}` to suppress the conference-specific footnote.
- (A) For the first submission (including peer review), you should *not* include your own names or group name in the report or its filename (since it will be reviewed anonymously by your colleagues)! This is the default setting in the LaTeX template. However, add the number of group members at the end of the abstract (for peers to evaluate the number of methods).
  (B) For the second submission, you should include your names. In LaTeX this is done by the `final` option, i.e., use `\usepackage[final]{neurips_2022}`.
- The LaTeX template has line numbers in its draft mode. You should not remove these numbers for the first submission. They can be useful for your reviewers when they want to refer to a specific part of your report (e.g., "the equation on line 54").
- Make sure all plots are readable and support your statements. General rule: font size in figures should roughly be equal to the font size in the text.

**Code template:** Your submission should include the main code in the appendix section. To make your code readable in the PDF file, you can use a code-style template by adding the lines

```
\usepackage{xcolor}
\definecolor{codegreen}{rgb}{0,0.6,0}
\definecolor{codegray}{rgb}{0.5,0.5,0.5}
\definecolor{codepurple}{rgb}{0.58,0,0.82}
\definecolor{backcolour}{rgb}{0.95,0.95,0.92}
```

```
\usepackage{listings}
\lstdefinestyle{mystyle}{
    backgroundcolor=\color{backcolour},
    commentstyle=\color{codegreen},
    keywordstyle=\color{magenta},
    numberstyle=\tiny\color{codegray},
    stringstyle=\color{codepurple},
    basicstyle=\footnotesize\ttfamily,
    breakatwhitespace=false,
    breaklines=true, captionpos=b,
    keepspaces=true, numbers=left,
    numbersep=5pt, showspaces=false,
    showstringspaces=false,
    showtabs=false, tabsize=2,
}
\lstset{style=mystyle}
```

before `begin{document}` and use `\lstinputlisting[language=python]{code.py}` to insert code to the appendix section.

*Another choice*: If you write your code using jupyter notebook, you can also directly export the notebook to a PDF and concatenate it to the report as an additional appendix. Here we recommend a useful website for jointing PDF: https://pdfjoiner.com.

# 6 First submission

## 6.1 Submission details

- The first submission is for *anonyomous* peer-review.
- Since the report is submitted anonymously, you need to specify the number of group members in your group manually. Do this by adding "Number of group members: $K$", where $K$ is the number of group members in your group, as the last sentence in the *abstract*.
- You have read the final version of the report from start to end; making sure it is readable; and all parts listed in Section 5.1 are included.
- The report does not contain material copied from elsewhere (all reports are checked for plagiarism using Urkund).

## 6.2 Peer review:

The peer-review is a mandatory part of the examination. For the grading criteria of the peer-review please see the grading section below.

Your report will be reviewed by students from other groups. Each student will also receive the report of another group, which you have to review. This means that the peer review is done individually and each group will receive multiple reviews.

As a peer reviewer, you are expected to comment on the following aspects of the report:

(I) Before the implementation, the data is properly analyzed and the questions from Section 3 are answered.

(II) The subset of methods chosen to explore is sufficiently large (methods from at least as many "families" as there are group members).

(III) All tasks (a)-(c) from Section 4.2 are made for each method.

(IV) Assess the technical quality of the proposed solution. Have the considered methods been used in a relevant way to address the problem at hand? Are there any flaws in the reasoning and/or motivations used?

(V) The report includes everything required from Section 5.1.

(VI) The quality of the language in the report is satisfactory.

(VII) The report follows the format requirements (correct template, page limitation, etc.).

The review process is "double-blind", meaning that both the project report and the review are anonymous. The review is done by filling out scores in the rubric of the project on Studium and by adding text comments to that rubric. Please follow the instructions on Studium for how to fill in and submit your review.

Of course, you should use polite and constructive language in your review. (Tip: *think about how you would assess your own report before you submit it!*)

After the review deadline, each group will get the reviews on their report from other students.

### 6.3 Grading

**Report:** Your report has to include all parts listed in Section 5.1. If we see that your report does not meet these criteria, we have to fail your report and you will have to re-do the project in the next course iteration.

For all other groups, no official grading of the report is done at this stage. Every group adhering to the criteria has to revise with the peer review feedback.

**For the peer-review:** Every student has to pass the peer review individually. In order to pass, your peer review should cover the listed points with comments in the rubric. You are not limited by the number of comments you can give. A guideline is to write 1/4 to 1/2 page of comments in order to pass. A peer-review without or with too little/few/not-useful comments leads to a "fail" of this part.

*Note: Even if you think that a point is well done, then comment on it and explain why you think so.*

If you do not pass the peer review, you will be assigned a second peer review after the second submission. Meaning that you have to do twice the work. Failing this new peer-review leads to a failure of the project and you then need to re-take the entire project next year.

## 7 Second submission

### 7.1 Submission details

**Report** After peer-review *all groups* have to resubmit an updated report including:

- Revisions accounting for peer review.
- A clear indication of the method you "put into production", i.e. based your submitted predictions on.
- Your names on the report so the second submission is not anonymous anymore.
- Remove the line numbering from the template.

**Code** Furthermore, you upload a `.zip` file containing all your (readable and commented) code to the same submission field so that it can be run without problems. Your results and figures should be reproducible with this code.

**Other uploads** In addition, you have to upload:

- A contribution statement (as a separate document, see below).
- A prediction for the test set (as a separate file, see below).
- Write a few sentences about how you have updated your report based on the peer-review. These sentences are submitted in the comment field on Studium when you submit your updated project report.

There is only one week between the peer reviews and the second submission.

### 7.2 Contribution statements

A single page should clearly state the contributions of each group member, clarifying who contributed to which part, etc. In particular, which method each took responsibility for and who did the work and wrote the other sections. This should be uploaded as a separate single document. One document should be uploaded per group.

If we see, that some students did not contribute to the project, we remove them from the group and they receive the grade fail. So make sure that you check the uploaded contribution statement in order to avoid conflicts between group members.

You need to include the contribution statement in order to pass. An empty contribution statement is not valid.

### 7.3 Model prediction

Upload a `.csv` file with the name `predictions.csv` with the format, e.g.

`0,1,1,1,0,...,1,0`

where `1` means high bike demand and `0` means low bike demand. This should be a single **row** of comma-separated values of length 400, no text. We will evaluate all submissions and publish a top table of best performers (with group numbers only). The performance of each group on the test set is evaluated and published.

You need to include the model prediction in the correct format in order to pass. If you do not upload this, we cannot evaluate whether your model works.

### 7.4 Grading

The **second submission** of the report will be graded with one out of three possible grades:

- Fail, if the deadline is missed or the report is far from meeting the criteria. No revision is possible until the next time the course is given.
- Revise, if there are only minor issues. A revised version should be handed in before the revision deadline.
- Pass, if the report fulfills *all* criteria.

## 8 Third submission: graded by teachers

**Save us all a lot of work and do not get to this point!** However, if you got revise in the second submission, the **third submission** of the report will be graded with one out of two possible grades:

- Fail, if the deadline is missed or the revised report still does not meet the criteria. No more revision is possible until next time the course is given.
- Pass, if the report fulfills all criteria.

*Please note that sub-standard reports will not be given the chance to be revised.*

**Good luck! ♣**