Machine Learning Algorithms

MAAI | EST-IPCA

Regression model to predict the number of bike rentals

Practical Work

Filipe Matos

34017

Abstract

This project presents a study of bike rentals over a two-year period, analyzing environmental and temporal conditions across 13 variables, with more than 17,000 recorded entries.
The dataset was originally created by Hadi Fanaee-T and João Gama in 2013 and is available for download online.
Given the nature of this dataset, the objective is to reverse-engineer the cnt variable, which represents the total number of rented bikes, and subsequently predict—within an error margin—the expected number of bike rentals using a machine learning–based model.
This document serves as supporting material for the Python code, which can be accessed in a Google Colab notebook.

Keywords:
Machine Learning, Bike Rental, Random Forest, Prediction Model, Linear Regression

Disclaimer:
The topic of *bike rentals* was not my initial choice for the analysis conducted in this project. Given my professional background, I originally intended to work with a dataset related to work–life balance.
The project was initiated; however, the dataset proved to be impractical, producing results that were worse than expected. In fact, simple guess-based approaches could likely have achieved better performance.
One of the main challenges of the dataset was its reliance on self-reported symptoms and personal assessments, which introduced a high level of subjectivity and susceptibility to bias—making it particularly difficult for a student to obtain meaningful and reliable results.
For this reason, I decided to pursue a more "realistic" dataset containing more factual and objective information and, at the last minute, shifted the project focus to bike rentals.
The original Google Colab notebook related to the work–life balance dataset can be accessed online – in here.
It is important to note that the change of subject was not the reason this project did not fully comply with the proposed timeline; this was primarily due to poor time management on my part.

The project is divided into six phases, plus a final presentation.
Throughout this document, the structure follows this same sequence.

Phase 1

Problem to Be Explored

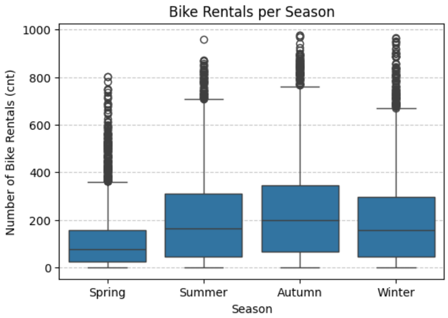
After a turbulent initial approach, I decided to follow the professor's suggestion and explore the domain of bike rentals.
Upon closer examination of the dataset, it became clear that a predictive model could be built for almost any variable. However, from a management perspective, the variable most aligned with practical business needs is *cnt*, which represents the total number of bikes rented.
The model is therefore designed to predict *cnt* based on the majority of the remaining variables, which include information such as temperature and hour of the day.
This analysis was chosen because the dataset contains more than 17,000 entries, allowing the model to be trained on a substantial amount of data.
The original dataset can be accessed online – in here.

Phase 2

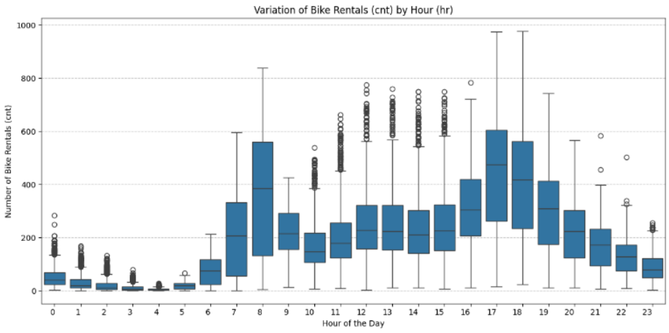
Data Understanding and Preparation

This dataset was already free of missing values, and dependent variables such as *cnt*—which depends on the *casual* and *registered* variables—were also properly structured.
At this stage, the primary focus was to gain a clear understanding of the data and the problem being addressed. The Exploratory Data Analysis (EDA) reflects the outcome of this approach.

Exploratory Data Analysis



Graphic showing the diferences of bike rentals across seasons



Graphic showing the distribution of Bike rentals along a 24H day

The overall analysis revealed results that were largely consistent with expectations:

- Higher bike rental volumes were observed when the weather was warmer.
- Bike rentals increased under clearer weather conditions, with no rain or snow.
- A higher number of bikes were rented during daytime hours.
- Peak demand occurred between 6–8 a.m. and 5–6 p.m., likely reflecting typical commuting patterns.
- No significant differences in bike rental volume were observed across weekdays; however, Saturdays and Sundays showed a slightly lower demand compared to other days.

Phase 3

Model Selection and Justification

The target variable, *cnt*, is an integer value, as it is not possible to rent a fraction of a bike. In order to determine how this final number could be predicted, it was essential to understand how climatic and temporal variables influence it.

One of the first relevant observations was the strong similarity between the *temp* and *atemp* variables—where one represents the actual temperature and the other the perceived (“feels-like”) temperature. This correlation was identified through a heatmap, which also revealed that these variables, along with *hr* (hour of the day), contribute most significantly to explaining *cnt*.

Further tests confirmed this hypothesis by measuring the impact of each variable on the target, ranked according to their individual performance:

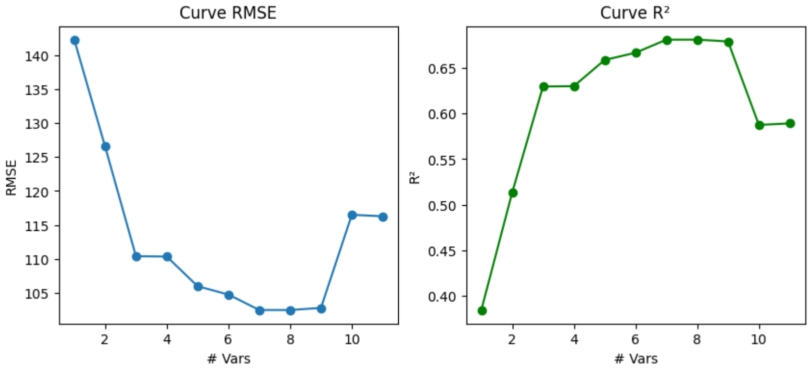
- **hr**: RMSE 136.005
- **atemp**: RMSE 166.165
- **temp**: RMSE 167.531
- **hum**: RMSE 176.339
- **weathersit**: RMSE 180.287
- **windspeed**: RMSE 181.993
- **holiday**: RMSE 182.144
- **workingday**: RMSE 182.164
- **weekday**: RMSE 182.191

- **season**: RMSE 183.188

The values above highlight the relative importance of each feature with respect to the target variable by indicating the expected prediction error when relying on a single variable. In this context, a higher RMSE corresponds to a greater margin for error.

During the remainder of Phase 3, a series of experiments was conducted to evaluate the performance of different feature combinations. The goal was to identify an optimal subset of variables capable of predicting the target effectively without introducing unnecessary noise or redundancy.

With these results, it becomes clear why the *season* variable ranks last. Compared to variables that directly describe weather conditions or time of day, *season* is less informative. For instance, users are more likely to rent a bike on a sunny winter day than during a heavily rainy summer day, making immediate environmental conditions more relevant than the season itself.



Graphics showing how the each curve behaves at the addiction of another variable.

We began by testing several models using all available variables (excluding *registered* and *casual*) in order to interpret an initial set of results. Subsequently, using the seven "optimal" variables identified earlier, we evaluated which type of model would yield the best performance. The following models were tested:

- Linear Regression
- Ridge Regression
- Lasso Regression
- Decision Tree
- Random Forest
- Gradient Boosting

It is worth noting that when the feature count reached seven, the performance curve tended to peak, indicating that seven variables were sufficient for building an effective predictive model.

Random Forest produced substantially better results than the other models. Interestingly, even when including potentially distracting variables by using all available features, the error rate was significantly lower. Although these results initially seemed counterintuitive, the project proceeded based on the observed performance metrics.

Phase 4

Implementation and Experimentation

Given the remaining uncertainty regarding the optimal number of variables, two models were developed:

- **rf_bike_model.pkl**, using seven selected variables
- **all_variables_model.pkl**, using all permitted variables

Both models were implemented using the same configuration: a Random Forest algorithm with 100 trees and a maximum depth of 15.
The dataset was split into 70% for training and 30% for testing, meaning that each model was trained on approximately 12,000 observations.
The same random seed was used for both models to ensure consistency, allowing them to learn from identical training data and enabling a fair comparison of results.

The RMSE results once again favored the model using all available variables, reinforcing the idea that a richer feature set leads to more accurate predictions of the target variable.

Phase 5

Evaluation and Validation

This phase aims to determine whether the models' performance is reliable or merely superficial. Although the target variable represents a specific and discrete value, it is unrealistic to expect the algorithm to predict the exact number of rentals with absolute precision. For this reason, an acceptable *error range* was defined to evaluate model performance.

The evaluation criteria were established as follows:

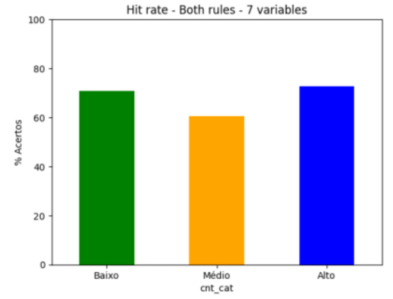
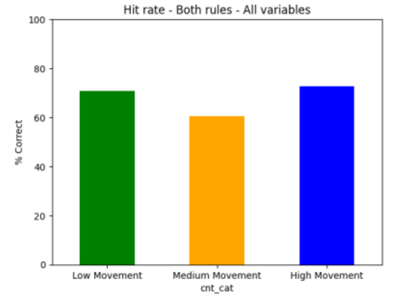
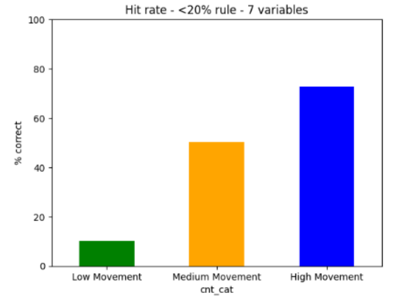
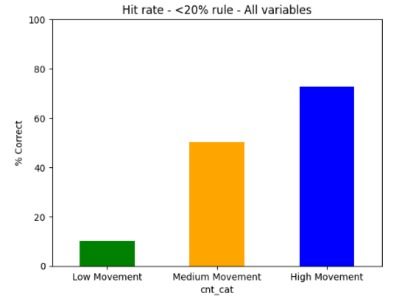
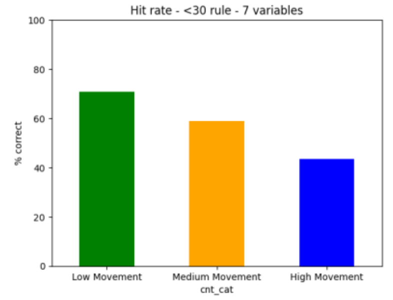
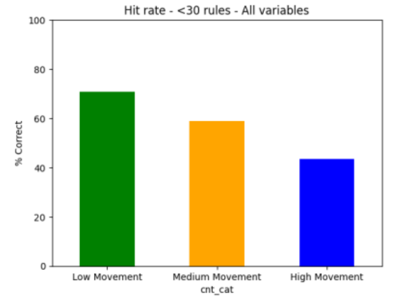
- If the difference between the actual value and the predicted value was less than **20% of the actual value**, the prediction was considered correct.
  - *Example:*
    Actual value: 100
    Predicted value: 90
    100 − 90 = 10
    20% of 100 = 20
  - Since the error is below 20, the prediction is considered correct.

- If the absolute difference between the actual value and the predicted value was less than **30 units**, the prediction was also considered correct.
  - *Example:*
    Actual value: 100
    Predicted value: 75
    100 − 75 = 25
  - Since 25 is less than 30, the prediction is considered correct.

To gain a clearer understanding of how the models make prediction errors, three distinct demand levels were defined:

- **Low Demand** – when the number of rented bikes is fewer than 50
- **Medium Demand** – when the number of rented bikes ranges from 51 to 200
- **High Demand** – when the number of rented bikes exceeds 200

As in previous phases, both models were evaluated using these categories, resulting in the following outcomes:

	7 Variables model	All Variables model
With the <20% and <30 rule	 	
With the <20% rule	 	
With the <30 rule	 	

After completing all tests, it is evident that both models benefit from the combined use of the two evaluation rules, resulting in a higher overall rate of correct predictions. A closer inspection of the results reveals the following patterns:

- **Low Movement:**
    - This category performs better when applying the **<30 units** rule, as the allowed error margin exceeds 50% of the actual value.
- **Medium Movement:**
    - This range demonstrates the most balanced performance across all tests, consistently achieving more than 50% correct predictions.
- **High Movement:**
    - Better results are observed when applying the **<20%** rule, since for larger values the absolute difference can easily exceed 100 units.
- **Both Rules Combined:**
    - The strongest overall performance is achieved when both rules are considered together, with a clear advantage for the model using all variables.

    - Correct prediction rates exceed 68%.
- **<20% Rule:**
    - This rule can be considered fair, as in a normally distributed dataset a 20% deviation still falls within one of the most populated regions.
    - However, when comparing outcomes, it does not appear to be the most suitable option for high-demand scenarios.
- **<30 Units Rule:**
    - From a managerial perspective, this rule is also reasonable, as it allows for a practical margin of error while still enabling effective inventory planning.
    - Its main limitation arises in low-demand situations, where it may overestimate the accuracy of predictions compared to the percentage-based rule.

From a more personal standpoint, these results are encouraging, particularly as a first modeling attempt, given an overall correct prediction rate of nearly 70%. Another important point is the consistency observed in the medium-demand range, which consistently achieved hit rates above 50%.
For a more in-depth analysis and improved performance, a more customized model—along with demand-specific evaluation rules—would likely lead to more accurate and reliable estimates.

Phase 6

Scalability and Deployment

With the model operational and a clear understanding of its performance in real-world scenarios, the goal of this phase was to test the model using new, randomly generated values across the selected variables.

Since the model with the best performance continued to be the **"all variables" model**, the variables that needed to be simulated were the same as those in the original dataset, except for the date variable (*dteday*) and the variables directly related to the target (*casual*, *registered*, and *cnt*).

The code automatically generates a synthetic dataset with 10,000 entries and quickly predicts the number of bikes required for each entry. The top 10 generated rows illustrate the model's output.

An example of the generated variables (from one of my test runs) is as follows:

season	yr	mnth	hr	holiday	weekday	workingday
2	0	6	16	1	5	0

weathersit	temp	atemp	hum	windspeed	Prediction
1	0.24	0.3952	0.51	0.0	51

The main challenge in this scenario is the independence of each variable. For example, it would be unusual for the actual temperature and the perceived temperature to differ significantly, or for high humidity to occur on a clear day.

Other

This last sectin was created when I had already finished the report, and was ready to deliver, when I noticed that by the ausence of nulls in the dataset, it didn’t meant that it was “clean”.

Even in one of the first approaches we can see that are multiple outliers in several variables, this outliers are moments that an entry is disaligned, by a considerable amount, from the expected.

Example, if in a heavy rainy day, in a frozen weather, the number of bikes rented is similar to one in a summer, the program should not look at this and see it as “normal” moment, and discart it, focusing on the more ordinary events.

My approach, although late remained as inssuficiente in reaching higher levels of predictment accuracy, reaching levels inferior to our previous “all variables” model.

Conclusion

At the end of this project, I feel that the main objectives of the course were, overall, satisfactorily achieved.

Coming from a Fine Arts background, my greatest challenges were related to coding. I would not have been able—or comfortable—developing a model without regularly using AI tools. My approach to programming relied heavily on a constant set of questions, such as: *“I want to compare X and Y; how do I do it?”* or *“I need to determine the value of Z from variable W; how do I obtain it?”*

However, in terms of understanding the code, I am confident that I can explain why I made each decision and where in the process it is implemented.

Regarding the assimilation of the course content, I recognize clear progress in my learning process and can justify the reasoning behind each choice made throughout the project. The phases of analysis and problem interpretation are where I feel most comfortable. Completing a practical model, even with some adjustments required, gives me confidence and motivation to continue exploring, developing, and dissecting this topic as a case study.