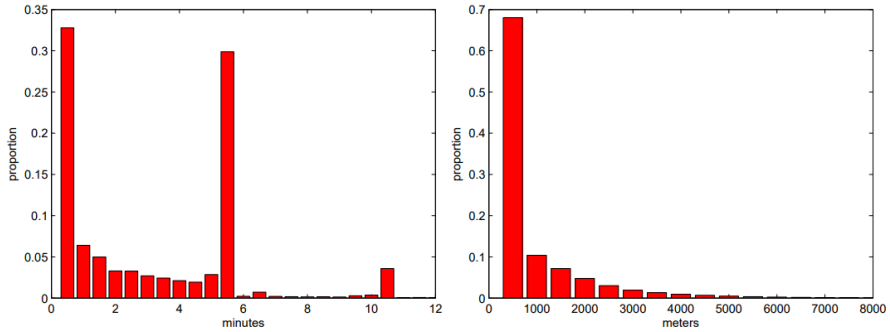
Introduction

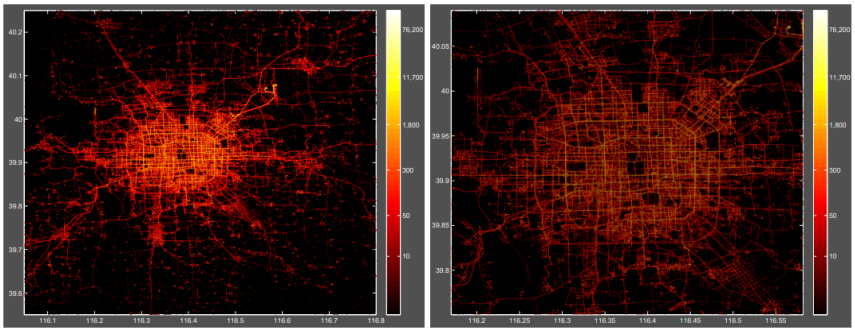
Taxi transportation is a very important factor in the lives of thousands of people, especially in large cities where public transportation is often undersized, and using private vehicles can be inconvenient and expensive. The problem we address is based on data from the Chinese city of Beijing, where, to give an idea of the scale, there are over 70,000 taxis[1].

The goal of our work is to analyze in depth a dataset containing the GPS trajectories of 10,357 taxis during the period from February 2 to February 8, 2008, within Beijing. This dataset includes about 15 million points, and the total distance covered by the trajectories reaches 9 million kilometers. To provide a better understanding of the nature of this dataset, two figures have been created: Fig. 1 shows the distribution of time intervals and distance intervals between two consecutive points, with an average sampling interval of about 177 seconds and an average distance of approximately 623 meters. Fig. 2 visualizes the density distribution of the GPS points in this dataset.



(a) time interval (b) distance interval

Figure 1: Histograms of time interval and distance between two consecutive points



(a) Data overview in Beijing (b) Within the 5th Ring Road of Beijing

Figure 2: Distribution of GPS points, where the color indicates the density of the points

After thoroughly analyzing the graphs, a data structure capable of storing and managing the information in the dataset must be created, choosing between a database or a data warehouse. Finally, a forecasting model must be built to predict where it will be easy or difficult to find a taxi in the next 3, 6, and 12 hours, aiding in understanding and anticipating patterns in taxi availability. By implementing this model, we hope to provide insights into taxi demand trends, which could support urban mobility planning and optimize resources for taxi services.

State of the art

Today, there are many projects similar to this one but with slightly different objectives. In fact, the project from which the dataset was sourced does not aim to create a forecasting model to understand where it will be easy or difficult to find available taxis. Instead, its goal is to improve individual mobility by studying the traffic flows generated by taxis to identify the best routes to follow in real-time[2]. Another example of a similar, though not identical, problem is "Finding the Shortest Path of Taxi Pick-Up Location to Customers Using A* Pathfinding Algorithm."[3] The aim of this project is to find the shortest path for the customer to reach a well-serviced pick-up location, thereby significantly reducing customer wait times by offering an efficient solution.

A further example of a project similar to ours is "Predicting the Upcoming Services of Vacant Taxis near Fixed Locations Using Taxi Trajectories"[4] which presents an effective solution based on the Hidden Markov Model to predict upcoming services of vacant taxis appearing at certain fixed locations and specific times.

Our project differs from those presented in this paragraph because it involves storing data using a database or data warehouse and, more importantly, developing a forecasting model, which the cited papers do not address. The usefulness of our project lies in providing possible insights regarding the future rather than the present, allowing people to better plan their day or commitments based on the estimates provided.

Proposed approach

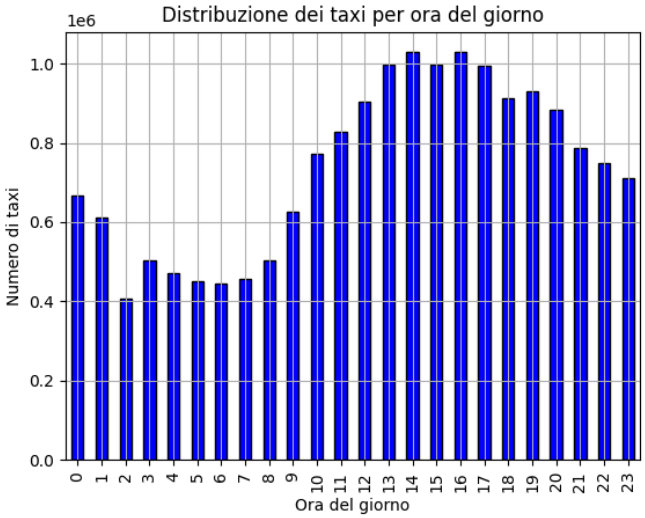
Given that the data in the dataset is structured in the order of taxi ID, date and time, longitude, and latitude, and considering the large volume of data that needs to be stored but on which we do not need to perform specific queries to achieve our goals, we decided to use a data warehouse to store it. This choice is justified by the fact that data warehouses are much more suitable for storing large amounts of data without requiring complex operations, unlike relational databases, which are well-suited for smaller datasets where numerous queries need to be executed.

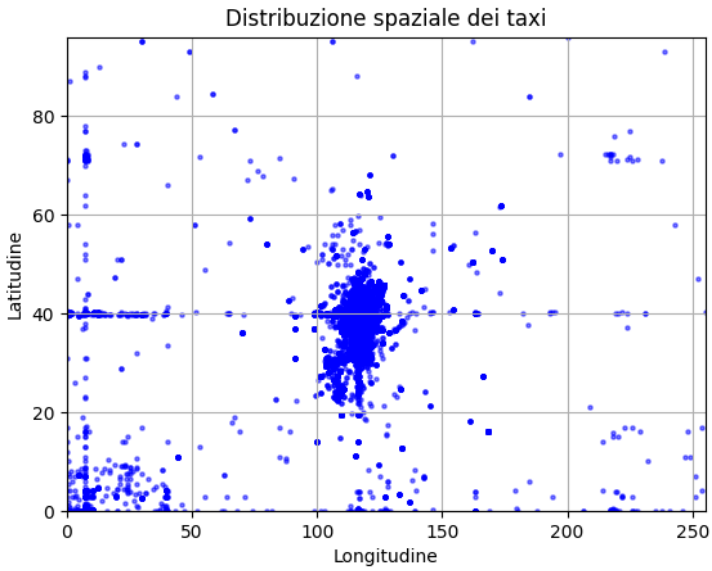
In particular, before the creation of the data warehouse, it was decided to perform a brief preprocessing of the data to make it easier to manage. Initially, the provided data was divided into more than 10,000 txt (plain text) files. Therefore, it was decided to merge all the files into a single text file (input file) to simplify the process of loading the data into the data warehouse. To achieve this, we used a simple Python program (based on Pandas) that we developed, which can be found in the github page.

Subsequently, we created the Data Warehouse using pgAdmin 4. Specifically, our database is named "taxi_data" and contains a table called "taxi", which takes the .txt file created in the previous step as input. The table consists of four columns: taxi_id, date_time, longitude, and latitude.

At this point, it is useful to study the data we have available. To do so, we use Python: with the pandas library we save all the data from the input file in a DataFrame. Using the df.describe()

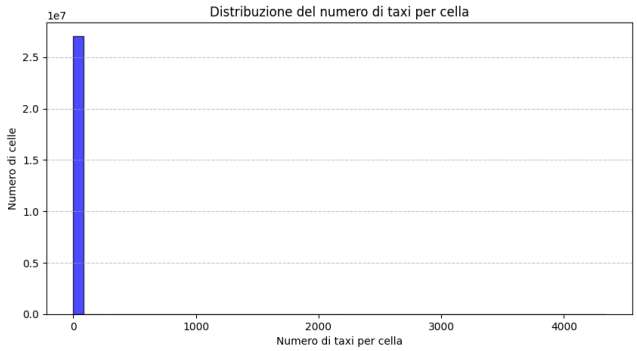
command, we extract statistics such as the mean, minimum, and maximum values to better understand the input data.
Next, we group the number of taxis by hour and create a bar chart. Additionally, to obtain a spatial visualization, we create a scatter plot using latitude and longitude. The results are as follows:





As you can see, the number of taxis is higher during the day and lower at night. Also, in terms of location, we notice a central area with many taxis, which makes our data quite varied with very different values.
At this point, we have decided to proceed with our project by dividing the city of Beijing into cells with a size of 0.03 degrees in both longitude and latitude, considering the number of taxis present in each cell. We chose the value 0.03 because it seemed like the right balance in terms of size while taking into account the number of taxis per cell. Smaller cells would have made our project more precise but would have significantly increased the number of cells with very few taxis. On the other hand, larger cells would have provided a better overall view of the problem

but would have significantly increased the number of cells with many taxis inside. Below is the graph for the chosen case:



At this point, we focused on creating a forecasting model to predict the location of taxis in the next 3 hours. Given the high variability of our data, we decided to start with a robust model, so we built a random forest regression model.

First, we created a dataframe that takes as input the division of the city into cells with the number of taxis assigned to each cell. This dataframe was then split into a training set and a testing set with a 70-30 ratio, ensuring that our model could learn effectively while minimizing the risk of overfitting.

Next, we further processed the data using a logarithmic transformation to make it more manageable, as it contained many extreme values. At this point, we separated the independent variables X from the target variable Y and trained the Random Forest Regressor model.

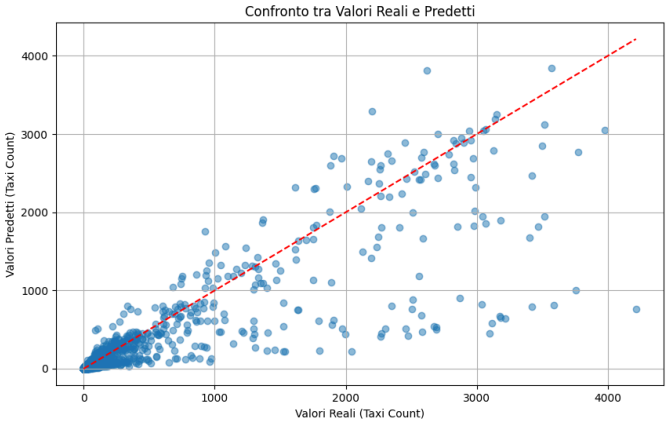
With this approach, we obtained predictions that we categorized into:

- Low probability areas (cells with less than 3 taxis)
- High probability areas (cells with more than 300 taxis)

Finally, we evaluated our model using the Mean Squared Error (MSE) and R^2 score for both the training and testing sets. We also created a graph comparing the predicted taxi values with the actual values.

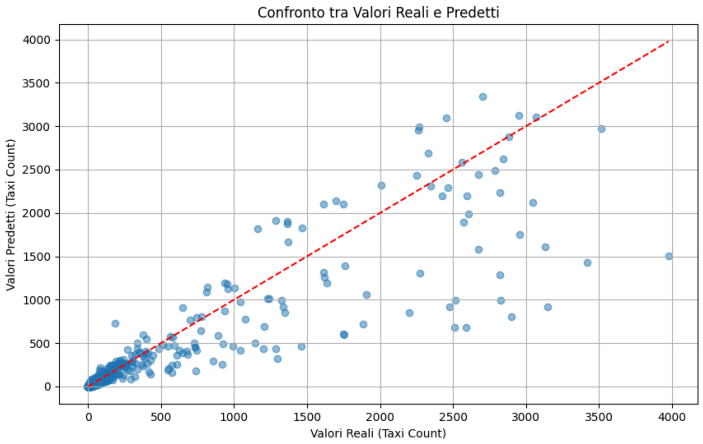
Below is a summary table of our model's performance:

Prediction After 3 Hours:



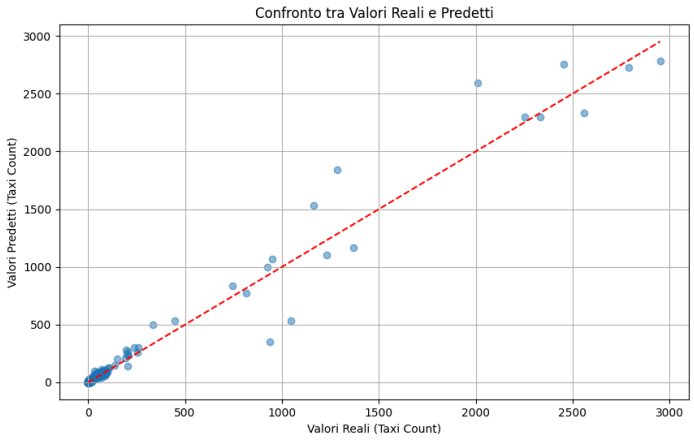
Mean Squared Error (MSE): 43.699303294167535
Varianza dei dati reali: 200.88270830875953
R² sul testing set: 0.7824635895141303
R² sul Training Set: 0.994152887740141

Prediction After 6 Hours:



Mean Squared Error (MSE): 37.02399682254985
Varianza dei dati reali: 209.9812750091864
R² sul testing set: 0.8236795313252093
R² sul Training Set: 0.994123664076674

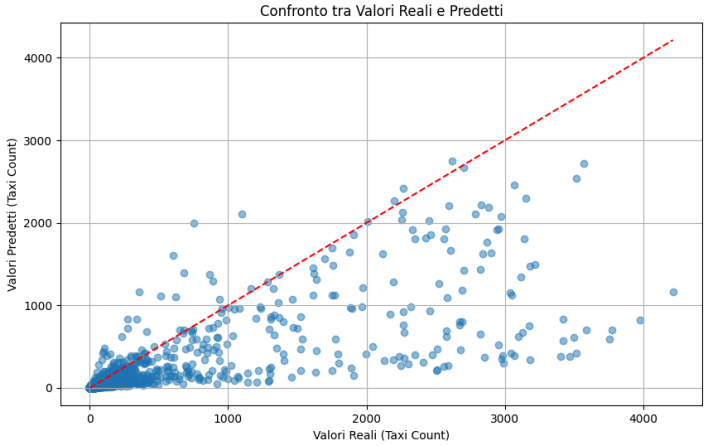
Prediction After 12 Hours:



Mean Squared Error (MSE): 5.6644568470171945
Varianza dei dati reali: 180.74527567481084
R² sul testing set: 0.9686605537773034
R² sul Training Set: 0.9965179334884996

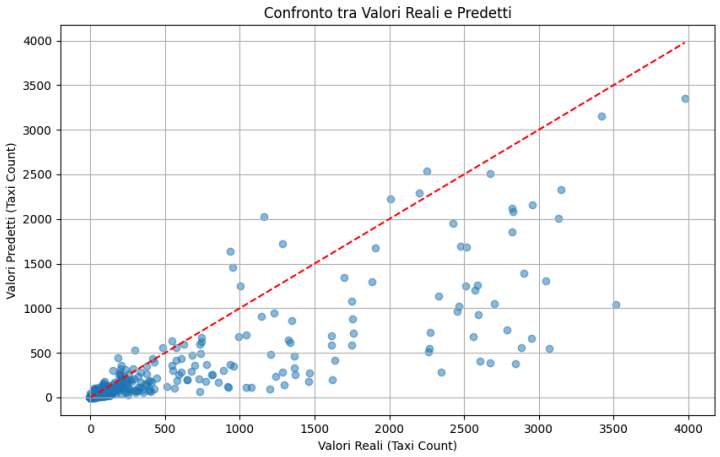
As seen from the obtained graphs, the results are quite interesting, with high R^2 values on both the training set and the testing set, showing the good performance of our model without any apparent overfitting issues. The model is particularly efficient in predicting low taxi values since this is a simpler task and very common in our input dataset.
However, we decided to implement an additional method for predicting the number of taxis: a simple linear regression model.
This model works in a very similar way to the previous one, with the main difference being the chosen model, which in this case is "Linear Regression".
This model is simpler than random forest but also less robust, as it can learn efficiently only if the input data follows a linear pattern.
The results provided by this model for the predictions over the next 3, 6, and 12 hours are as follows:

Prediction After 3 Hours:



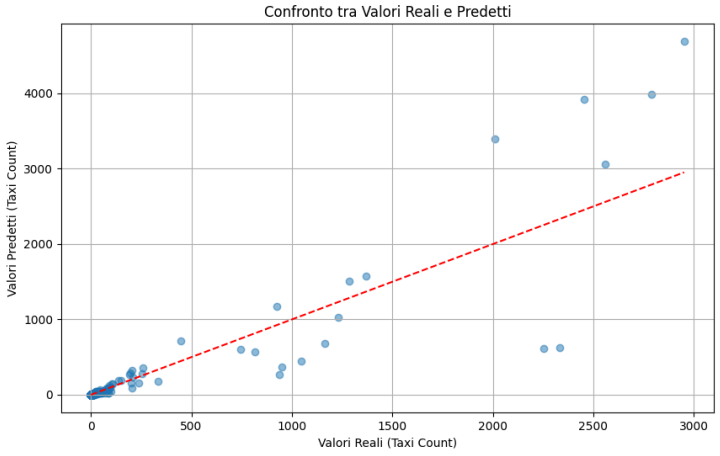
Mean Squared Error (MSE): 84.21473167306563
Varianza dei dati reali: 200.88270830875953
R^2 sul Testing set: 0.5807766015199953
R^2 sul Training Set: 0.7588857611773419

Prediction After 6 Hours:



Mean Squared Error (MSE): 68.18129073594613
Varianza dei dati reali: 209.9812750091864
R^2 sul Testing set: 0.6752982344117907
R^2 sul Training Set: 0.6397072350933082

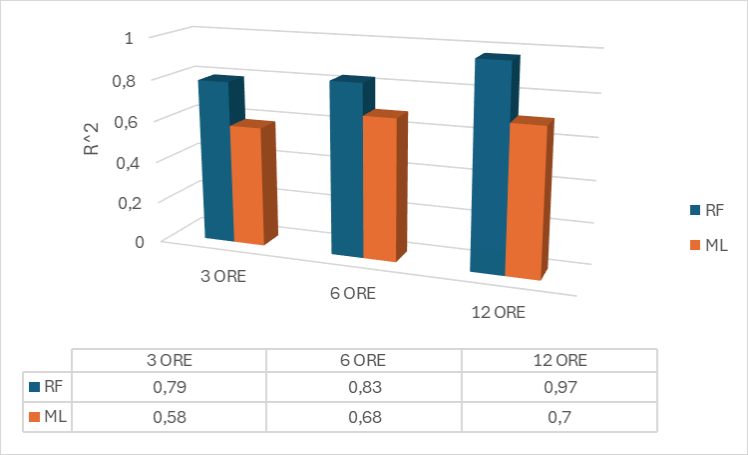
Prediction After 12 Hours:



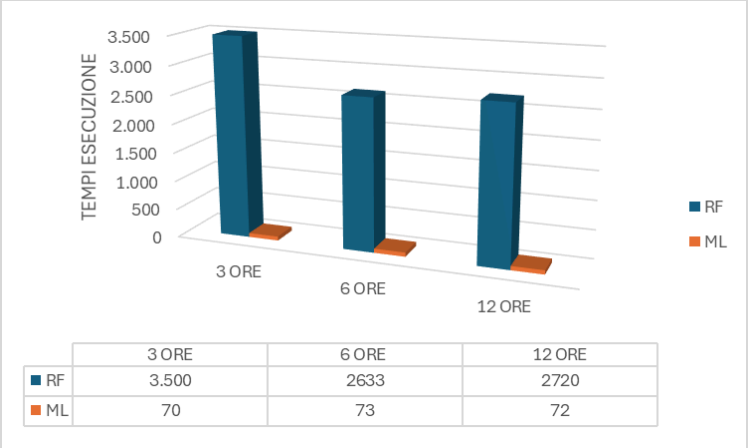
Mean Squared Error (MSE): 52.4343467598045
Varianza dei dati reali: 180.74527567481084
R^2 sul Testing set: 0.7098992127786392
R^2 sul Training Set: 0.5440488524877327

As you can see from the graphs and R^2 values, the linear model is much less effective than the random forest, probably because the input data is not well-suited for this model. Once again, the algorithm provides better predictions for low taxi values, as mentioned earlier.

Conclusions



	3 ORE	6 ORE	12 ORE
RF	0,79	0,83	0,97
ML	0,58	0,68	0,7



	3 ORE	6 ORE	12 ORE
RF	3.500	2633	2720
ML	70	73	72

As clearly shown in the previous graph, the model that achieves the best results in terms of R^2 is the random forest, likely due to its greater robustness. However, this also results in significantly longer execution times.

References

[1] https://daxing-pkx-airport.com/transportation/beijing-taxis/#:~:text=Taxis%20in%20Beijing%3A%20An%20Overview,matter%20the%20day%20or%20time.

[2] https://www.microsoft.com/en-us/research/project/t-drive-driving-directions-based-on-taxi-traces/

[3] https://www.researchgate.net/publication/262691903_Finding_the_Shortest_Path_of_Taxi_pick-up_location_to_Customers_Using_A_Pathfinding_Algorithm

[4] https://www.mdpi.com/2220-9964/8/7/295