Danmarks
Tekniske
Universitet

Project Report

42186 Model-Based Machine Learning

Authors

Ulrik Longva Haugland - s241658
Giovanni Fassoni - s242942
Florian Schwill - s250424
Matteo D'Souza- s242947

May 29, 2025

Contents

1 Introduction

1.1 Topic and data

In recent years, food delivery services have experienced significant growth, due to increasing demand starting from the COVID-19 pandemic. As more and more people use them, the need to optimize the delivery process has become crucial for companies to remain viable and profitable. In this regard, an accurate prediction of delivery time is extremely important for various reasons:

- Providing customers with an accurate delivery timing has a positive impact on customer satisfaction
- Keeping track of the availability of drivers can help companies in defining the number of drivers needed
- Having a baseline for delivery times can simplify drivers' performance evaluations

At a fundamental level, delivery time can be approximated as the product of distance traveled and average speed. Consequently, identifying variables that influence either distance or speed is essential for building effective predictive models. Since this is a data-driven project, we used a dataset[1] which contains the following features: vehicle type, vehicle condition, traffic conditions, weather conditions, locations of pick-up and destination, and city type (rural or urban). We assume that different vehicle types respond differently to these variables: for instance, a bike can be less affected by congestion than a car. This suggests that a hierarchical modeling approach, which accounts for group-level variation (e.g., by vehicle type), is well-suited to capture these nuanced relationships and improve the accuracy of delivery time predictions.

1.2 Research question

Based on the dataset considered and the previous assumptions, we formulate the following research question:

> *Can a Machine Learning model leverage contextual and vehicle-related features to accurately forecast food delivery times?*

To answer this question, we formulated a Probabilistic Graphical Model, employed multiple machine learning methods, and conducted an analysis to compare their predictive accuracy. The comprehensive Python notebook relative to these studies is submitted alongside this report.

[1] available at `https://github.com/Ritik1129/Food_Delivery_Dataset`

2 Methods

2.1 Data Exploration

As a first step when analyzing the dataset, we perform a basic data exploration to gain information regarding the structure of the data. We first check for missing values, which we find none of. Thereafter, we add a column with distance between the restaurants and the pick-up point for each order, as this is probably closely related to the timing. After this, we plot the distributions of all the columns with numerical values. Based on the plots, it appears that there is no linear relationship between the day of the week and the delivery time, and the time of order and delivery time. As a result, these columns will be encoded using one-hot encoding. We then plot the distributions of the columns without numerical values. We proceed to encode the road traffic density variable, which is an ordinal categorical variable with levels: Low, Medium, High, and Jam. These levels represent a natural progression in the severity of traffic conditions and are therefore suitable for label encoding using numerical values. We use one-hot encoding for the weather conditions as they lack a meaningful strict ordering among them. In such cases, label encoding could mislead the model into thinking there's a hierarchy, and to prevent this we use one-hot encoding, allowing us to treat each condition as a separate category.

2.2 Bayesian Linear Regression

The probabilistic regression model investigated is a Bayesian Linear Regression model, this was used for forecasting delivery time, based on the amount of traffic, the distance, the weather, and the day and time of the order. By labeling the deliver times as y and their corresponding attributes as X, and assuming Gaussian priors α and β, we get the following formulation:

$$p(\alpha, \beta, Y_n \mid X_n, \mu, \sigma) = p(\beta \mid \mu, \sigma)\, p(\alpha \mid \mu, \sigma) \prod_{n=1}^{N} p(Y_n \mid \alpha, \beta, X_n, \sigma)$$

2.3 Hierarchical Modeling

Now, the bayesian linear regression will be extended to create a hierarchical model.
The first addition will be that for each vehicle type separate β's are sampled from the same underlying distribution. The intuitive assumption here is that an unforeseen event, such as a traffic jam, impacts a scooter differently than a motorcycle because the travel speed might be reduced more for a motorcycle.
The second addition is that the distance-variable we introduced in the data exploration is the closest possible distance between pick-up and delivery location. In practice, the driver will follow streets, thus the actual distance will be larger. It is assumed that the added distance might be different depending on the kind of city (i.e., rural vs urban).

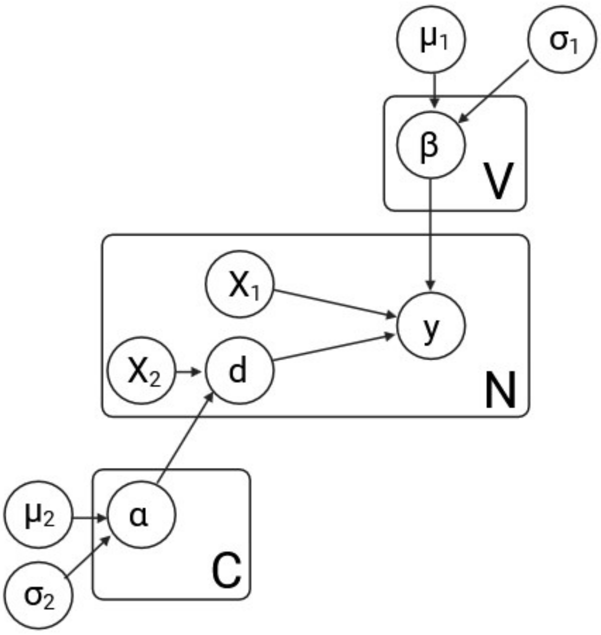

Figure 1: PGM for the hierarchical model

2.3.1 PGM for the hierarchical model

Below follows the probabilistic graphical model (pgm) for the hierarchical model.

2.3.2 Factorized joint distribution for the hierarchical model

$$
\begin{aligned}
p(y, \beta, \alpha, d, X_1, X_2, \mu_1, \sigma_1, \mu_2, \sigma_2) = {} & p(\mu_1)\, p(\sigma_1)\, p(\mu_2)\, p(\sigma_2) \\
& \times \prod_{v \in V} p(\beta_v \mid \mu_1, \sigma_1) \\
& \times \prod_{c \in C} p(\alpha_c \mid \mu_2, \sigma_2) \\
& \times \prod_{n \in N} p(X_{1n})\, p(X_{2n})\, p(d_n \mid X_{2n}, \alpha_{c[n]})\, p(y_n \mid X_{1n}, d_n, \beta_{v[n]})
\end{aligned}
$$

2.4 Inference based on MCMC NUTS

To perform inference, we use Markov Chain Monte Carlo (MCMC), a stochastic algorithm designed to sample from the posterior distribution. By drawing these samples, we can approximate the true posterior and compute expectations over model parameters. In this particular, inference on the prior parameters α (alpha) and β (beta) is conducted using the No-U-Turn Sampler (NUTS), an advanced variant of MCMC. NUTS builds on Hamiltonian Monte Carlo (HMC) by automatically adapting the trajectory length during sampling, which helps avoid inefficient random walks and improves convergence efficiency. The model is trained on the training dataset, and after convergence, samples of the intercept and Î˛ parameters are drawn from the posterior. These posterior samples are then used to make predictions on the test dataset and assess the model's accuracy.

2.5 Stochastic Variational Inference

Stochastic Variational Inference (SVI) is an inference algorithm that enable to approximate the true posterior distribution $p(z \mid x)$ by finding a simpler variational distribution $q(z)$ that lies close to it. This is achieved by minimizing the Kullback-Leibler (KL) divergence between $q(z)$ and $p(z \mid x)$:

$$\mathrm{KL}(q(z) \| p(z \mid x)) = -\underbrace{(E_q[\log p(z,x)] - E_q[\log q(z)])}_{\text{ELBO}} + \underbrace{\log p(x)}_{\text{log evidence}}$$

To understand the terms more clearly, the KL divergence can be expressed as:

$$\mathrm{KL}(q(z) \| p(z \mid x)) = -\left(\underbrace{E_q[\log p(z,x)]}_{\text{expected joint log likelihood}} \quad -\underbrace{E_q[\log q(z)]}_{\text{entropy of } q(z)} \right) + \underbrace{\log p(x)}_{\text{log evidence}}$$

The ELBO (Evidence Lower Bound) quantifies how close the variational distribution $q(z)$ is to the true posterior, while the log evidence $\log p(x)$ is constant with respect to $q(z)$ and represents the marginal likelihood.

Since $\log p(x)$ does not depend on $q(z)$, minimizing the KL divergence is equivalent to maximizing the ELBO. As the ELBO increases, the variational distribution $q(z)$ improves and becomes a better approximation to the true posterior.

3 Results

3.1 Bayesian Linear Regression

Metric	**Ridge Regression**	**Bayesian Linear Regression (SVI)**	**Bayesian Linear Regression (MCMC)**
CorrCoef	0.546	0.529	0.544
MAE	6.307	6.391	6.316
RMSE	7.830	7.952	7.842
R^2 Score	0.298	0.276	0.296

Table 1: Comparison of Ridge Regression and Bayesian Linear Regression Methods

Table 1 presents a comparison of Ridge Regression and two Bayesian Linear Regression approaches, Stochastic Variational Inference (SVI) and Markov Chain Monte Carlo (MCMC),

evaluated using four standard performance metrics.
The Correlation Coefficient (CorrCoef), which measures the strength of linear association between predicted and actual values, is highest for Ridge Regression (0.546), indicating slightly stronger predictive alignment.
The Mean Absolute Error (MAE), representing the average magnitude of prediction errors, is lowest for Ridge Regression (6.307), suggesting more accurate point estimates on average. Similarly, Ridge Regression achieves the lowest Root Mean Squared Error (RMSE), a metric that penalizes larger errors more heavily, at 7.830.
The R^2 Score, which quantifies the proportion of variance in the target variable explained by the model, is also highest for Ridge Regression (0.298). The Bayesian models perform comparably, with MCMC-based regression slightly outperforming the SVI-based approach across all metrics. This marginal advantage of MCMC may stem from its more accurate posterior sampling compared to the variational approximation used in SVI.
The differences are however small, and all models exhibit similar overall predictive behavior. These results suggest that while Ridge Regression performs best in terms of point estimates, Bayesian methods remain valuable for incorporating uncertainty, which may be crucial in downstream decision-making tasks.

3.2 Hierarchical Modeling

Metric	**Hierarchical Bayesian Model**
CorrCoef	0.562
MAE	6.286
RMSE	7.771
R^2 Score	0.308

Table 2: Performance of the Hierarchical Bayesian Model

Table 2 presents the same standard performance metrics applied to hierarchical regression. When compared to the values shown in table 1, the hierarchical approach demonstrates superior results across all metrics. It achieves the highest Correlation Coefficient (0.562), indicating stronger linear agreement between predictions and true values than Ridge (0.546), MCMC (0.544), or SVI (0.529). It also records the lowest Mean Absolute Error (6.286) and Root Mean Squared Error (7.771), suggesting it produces the most accurate predictions overall. Furthermore, its R^2 score of 0.308 surpasses all other models, meaning it explains the greatest proportion of variance in the data. These improvements underscore the advantage of the hierarchical structure in capturing group-level variations or latent factors, which the simpler models may miss.

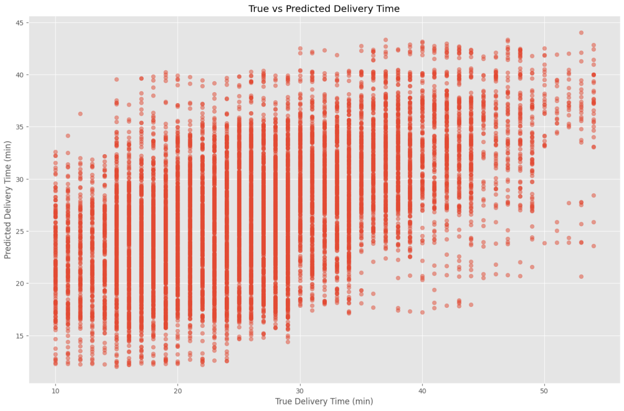

Figure 2: True vs. predicted delivery time

Figure 2 displays a scatter plot comparing true versus predicted delivery times, offering a visual assessment of model performance. Ideally, a perfect model would yield points lying along the 45-degree line, indicating equal predicted and actual values. In this plot, while there is a general upward trend, which signifies a positive correlation, predictions appear to be biased toward a narrower band, underestimating higher true delivery times and overestimating lower ones. This suggests the model may be regressing toward the mean, potentially due to limited flexibility or regularization. However, the dense clustering around the central diagonal implies that the model achieves reasonable predictive accuracy across the majority of cases.

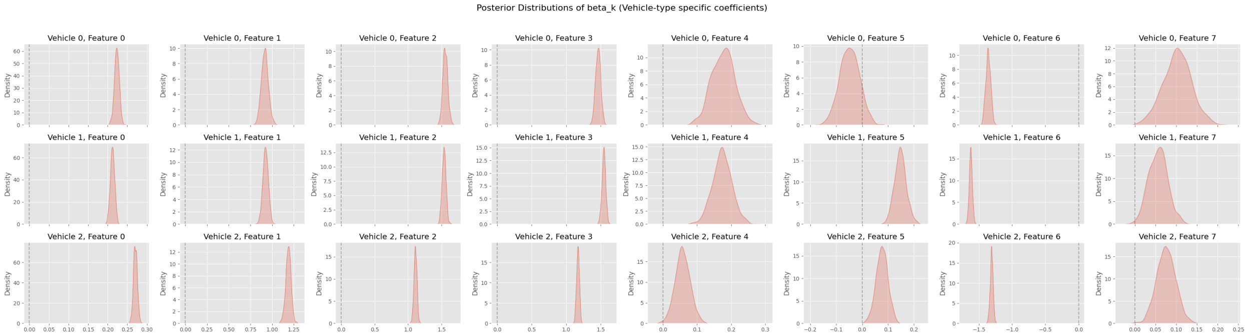

Figure 3: Posterior distributions of vehicle-type specific coefficients

Figure 3 displays the posterior distributions of vehicle-type specific coefficients for each of the vehicle types (where 0 stands for electric scooter, 1 for scooter and 2 for motorcycle) and each feature is a combination of values deriving from weather, traffic, and vehicle condition. Each subplot shows the uncertainty and central tendency of a specific feature's coefficient for one vehicle type. The plots reveal that while some features have similar effects across all vehicle types, others vary significantly, indicating feature-vehicle interactions. For example, the distributions for Feature 0 and Feature 1 are tightly clustered and distinct across vehicles, suggesting consistent and informative contributions. In contrast, some features such as Feature 4 and Feature 7 exhibit more overlap and spread, reflecting higher uncertainty or less predictive strength. These posterior estimates highlight the utility of a hierarchical model in capturing vehicle-specific behaviors within a shared framework.

4 Discussion

4.1 Conclusion

When forecasting delivery times, the hierarchical model performs slightly better than both the standard Bayesian linear regression and the Ridge Regression model. It is able to capture some structure across vehicle and city types, leading to more accurate predictions, reflected by higher correlation coefficients and R^2 scores, and lower RMSE and MAE values, although the improvement over the Ridge model and the Bayesian linear regression model is relatively small. Therefore,the results suggest that machine learning models can indeed improve contextual and vehicle-related features to forecast food delivery times with better accuracy. This highlights the value of enhancing simpler linear models by integrating hierarchical techniques, as the benefits of the hierarchical approach may become more significant with increasing model complexity.

4.2 Ideas for future work

Regarding the hierarchical model, one potential improvement is to introduce hyperpriors not only for the intercepts but also for each β coefficient associated with contextual features like traffic and weather. This would increase the level of complexity and allow for more flexible estimation across vehicle categories and city types, while still being stable. Additionally, exploring more expressive priors or regularization strategies could mitigate the issue of unrealistic forecasts. Another direction could involve integrating temporal components, such as modeling delivery time as a time-series.