electronics

MDPI

Article

Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 2—Model Creation and Handling Uncertainty

Jan Pomykacz, Justyna Gibas and Jerzy Baranowski *

Department of Automatic Control & Robotics, AGH University of Kraków, 30-059 Kraków, Poland; janpomykacz@student.agh.edu.pl (J.P.); justynagibas@student.agh.edu.pl (J.G.)
* Correspondence: jb@agh.edu.pl

Abstract: The e-commerce sector is in a constant state of growth and evolution, particularly within its subdomain of online food delivery. As such, ensuring customer satisfaction is critical for companies working in this field. One way to achieve this is by providing an accurate delivery time estimation. While companies can track couriers via GPS, they often lack real-time data on traffic and road conditions, complicating delivery time predictions. To address this, a range of statistical and machine learning techniques are employed, including neural networks and specialized expert systems, with different degrees of success. One issue with neural networks and machine learning models is their heavy dependence on vast, high-quality data. To mitigate this issue, we propose two Bayesian generalized linear models to predict the time of delivery. Utilizing a linear combination of predictor variables, we generate a practical range of outputs with the Hamiltonian Monte Carlo sampling method. These models offer a balance of generality and adaptability, allowing for tuning with expert knowledge. They were compared with the PSIS-LOO criteria and WAIC. The results show that both models accurately estimated delivery times from the dataset while maintaining numerical stability. A model with more predictor variables proved to be more accurate.

Keywords: online food delivery (OFD); delivery time estimation; Bayesian inference; generalized linear models

check for updates

Citation: Pomykacz, J.; Gibas, J.; Baranowski, J. Bayesian Modeling of Travel Times on the Example of Food Delivery: Part 2—Model Creation and Handling Uncertainty. *Electronics* **2024**, *13*, 3418. https://doi.org/10.3390/electronics13173418

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 19 June 2024
Revised: 15 August 2024
Accepted: 16 August 2024
Published: 28 August 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The e-commerce sector is in a constant state of growth and evolution, particularly within its subdomain of online food delivery (OFD) [1,2]. Recent market forecasts indicate a steady rise in revenue for companies offering such services. With numerous players in the market, ensuring customer satisfaction is paramount for a company's survival. Customers increasingly demand user-friendly applications that simplify the ordering process with just a few taps while also providing features such as delivery time estimates and communication channels with couriers [3]. However, estimating delivery times accurately without managing the uncertainty associated with real-time events and decision making may be suboptimal. This is indicated by recent studies, which either focus on this [4,5], account for this [6], or indicate this in future works [7].

The existing research in this field is broad. Some works focus on static origin–destination time-travel prediction [4]. Others create commercial-grade solutions capable of handling real-time data [6]. Recent studies have started to focus on more complex problems, e.g., the restaurant-meal-delivery problem, which is characterized by a fleet of delivery vehicles that serve dynamic customer requests over the course of a day [5]. A more detailed description is provided in Section 2.

Machine learning models are common in the task of time-travel prediction. Among these methods, neural networks, including Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Long Short-Term Memory Networks (LSTMs), are prominently utilized [8].

Although Bayesian statistics and inference have gained increasing popularity, there remains a notable scarcity of articles addressing their application in delivery time prediction. In their case study, Abdi et al. list less than ten methods based on either a Naive Bayes classificator, Bayesian Network, or Bayesian graphical model while examining around a hundred articles [8]. The method boasts several advantages: it offers straightforward, interpretable models; the capacity to adapt and improve with new data; and provides a measure of uncertainty for each prediction. However, it also presents challenges, notably its computational demands and the potential for poor model performance due to incorrect assumptions.

This article introduces two Bayesian models designed for predicting food-delivery times. Utilizing a linear combination of predictor variables, we generate a practical range of outputs. These models offer a balance of generality and adaptability, allowing for tuning with expert knowledge. This ensures flexibility and stability in various contexts. To assess their performance and identify any potential drawbacks, we compared the models using the PSIS-LOO criteria and WAIC.

The main contributions of this paper are as follows: (1) To the best of our knowledge, this is the first application of Bayesian inference to online food-delivery-time prediction. (2) By specifying models as linear combinations of predictors, we achieve high interpretability, which aids in identifying the primary factors influencing delivery time. (3) Our results indicate that Bayesian inference holds promise for further exploration in this context, as it can lead to promising results.

The remainder of this paper is structured as follows: In the next section, relevant studies in the extant literature are reviewed and discussed. Section 3.1 provides a short introduction to the concept of Bayesian statistics. Section 3.2 refers to the numerical computation methods utilized in our work. Section 3.3 reveals the data source and contains reference to part 1 of this article, where preprocessing is described. Section 3.4 focuses on the model definition, prior distribution selection, and prior predictive checks. In Section 4, we present and explain the results of our work as posterior predictive checks, where Section 4.1 focuses on Model 1 and Section 4.2 focuses on Model 2. In Section 4.3, we compare models to see how they fare against each other. Section 5 discusses the limitations of our models. Finally, Section 6 summarizes the conclusions drawn from this study.

2. Literature Review

Food-delivery-time estimation can be perceived in the category of Estimated Time of Arrival (ETA). In our work, it will also include meal-preparation time, but the rest focuses solely on the travel time between origin and destination. Overall, there are two common approaches to ETA: route-focused and origin–destination-focused.

The route-based approach focuses on segmenting routes and estimating the travel time of each segment. Lee et al. implemented a real-time expert system that takes present and historical data and produces travel-time-prediction rules via data-mining techniques. Also, they implement a dynamic weight combination governed by meta-rules, which allows for a real-time road events response to enhance the prediction's precision [9]. Li et al. proposed a deep generative model—DeepGTT. It is a probabilistic model designed to generate a travel time distribution, from which travel time as well as the uncertainty about it can be inferred [10]. Asghari et al. presented algorithms for computing the probability distribution of travel times for each link of a given route. It differs from other works as the authors mention that elsewhere, probabilistic link travel times are given a priori. This, and the work mentioned beforehand, are one of the few works that focus on distributions rather than strict numbers [11]. Wang et al. proposed a model for estimating the travel time of any path consisting of connected road segments based on current and historical GPS records, as well as map sources. Due to data sparsity (not every road will be traveled by a vehicle with GPS) and the trade-offs associated with multiple ways of connecting road fragments to form a route, the problem as a whole was not solved [12]. Wang et al. formulate ETA as a spatial–temporal problem. They adapted different neural networks,

as well as proposed the authorial Wide-Deep-Recurrent model and trained them on floating-car data. The solution showed promising results and was deployed for Didi Chuxing's vehicle-for-hire company [13]. Han et al. propose an incremental ETA learning framework to address issues of the scalability and robustness of real-world large-scale ETA scenarios. The framework works as an incremental travel-time predictor that is updated on newly generated traffic data. The authors also include a historical traffic knowledge consolidation module to reuse historical data and an adversarial training module to mitigate and resist traffic noise perturbations caused by low-quality data. The model was employed at Didi Chuxing's company, substantially improving the prediction accuracy [6].

The origin–destination methods refrain from estimating routes, stating that it is time consuming and potentially erroneous and gives a worse result than OD methods. Zhu et al. predict the Order Fulfillment Cycle Time (OFCT), which is the time between placing an order and receiving the meal. Their approach consists of identifying key factors behind the OFCT and capturing them within multiple features from diverse data sources, and then feeding them to the DNN created for this task. It is worth noting that their approach is specifically tailored to food delivery, which aligns with the common goal outlined in our article [14]. Li et al. proposed the MURAT model with the goal of predicting travel time given the origin and destination location, as well as the departure time. They also present a multi-task learning framework to integrate prior historical data into the training process to boost performance [15]. Wu, C. H. et al. examined a classical machine learning algorithm, which is support vector regression. Their findings show the feasibility of such a method for travel-time prediction [16]. Wang et al. leverage the increasing availability of travel data. Their approach is to use large historical datasets to accurately predict the travel time between the origin and destination without computing the route. The shown solution outperformed the services of Bing Maps and Baidu Maps at the time [17]. Lin et al. propose a framework called the Origin–Destination Travel Time Oracle to estimate travel time given the origin–destination pair and departure time. It uses historical trajectories alongside the OD pair to infer image-based Pixelated Trajectories. Based on the inferred trajectory, a Masked Vision Transformer is capable of estimating travel time. The results outperform most of the other solutions highlighted in the paper [18]. Zhou et al. examine the ETA problem in the context of e-commerce platforms. They introduce the Inductive Graph Transformer. Unlike other graph transformer architectures, it trains the transformer as a regression function that captures both information from raw features as well as dense embeddings encoded by a graph neural network. The graph neural network is also simplified to allow the solution to be applied to large-scale industrial scenarios. The results show performance improvement with metrics such as the mean absolute error, mean absolute percentage error, and mean absolute relative error compared to other models [19]. Zhang et al. propose the Graph-Structure-Learning-Based Quantile Regression model for ETA in e-commerce. According to the authors' knowledge, this is the first application of graph structure learning in this field and suggests that most of the other work utilizing fixed graph structures may be suboptimal. For the ETA, they design multi-objective quantile regression loss capable of finding a Pareto solution to the problem. The authors also propose fast sampling-based methods to reduce the computational complexity and enable the solution to be used for large-scale graphs. The results are shown to outperform baseline models [20].

Recently, new work was introduced into this field, building on existing ETA solutions to tackle problems of higher complexity. Ulmer et al. consider the restaurant-meal-delivery problem, which regard the optimization of the fleet of delivery vehicles serving dynamic customer requests throughout the day. The present anticipatory customer-assignment policy is used to handle the uncertainty of an unknown meal-preparation time as well as unknown customer localization. The policy is based on a time buffer and postponing to reduce making decisions that would result in delivery delays. Based on data from the city of Iowa, the authors show results that outperform other restaurant-delivery policies [5]. Hildebrandt and Ulmer combined ETA and the restaurant-meal-delivery problem. They

proposed an offline method, which maps a set of features to expected arrival times using gradient-boosted decision tree. The results show that it has a better performance in comparison to planning on means, which is a sum of the expected times of each action on route. The second proposed model is called offline–online, with real-time predictions in mind. It uses a pretrained DNN to approximate the exact route of delivery in a full online-simulation scenario. The authors show that this approach achieves a near full-optimal online-simulation accuracy with a fraction of the computational time [21]. Xue et al. focus on minimizing the cost of the restaurant-delivery problem with an uncertain cooking time and travel time and give insight into the influence of those uncertainties on food-platform preference. They propose a scenario-based chance-constrained programming model to capture the variability of cooking and travel times and develop an island harmony search algorithm to generate high-quality solutions. The results show that both uncertainties are critical for the restaurant-delivery problem [4]. Gao et al. combine the ETA problem with the estimation of the delivery route. While problems are closely related when it comes to food delivery, they are often examined separately. The authors propose a deep network named FDNET, consisting of route- and time-prediction modules. The route-prediction module is used to determine the next localization that a courier will visit in a multi-delivery scenario. The time-prediction module estimates the travel time between two adjacent locations based on the drivers and spatiotemporal features. Offline experiments show promising results compared to the frequently used machine learning models [7].

3. Materials and Methods

3.1. Bayesian Inference

For a better understanding of problem formulation and the proposed solution, a short introduction to Bayesian inference is in order. It is a method of statistical inference, in which we fit a predefined probability model to a set of data and evaluate the outcomes with regard to the observed parameters of the model and unobserved quantities, like predictions for new data points [22]. It is performed with the use of Bayes' rule, shown in Equation (1):

$$p(\theta \mid y) = \frac{p(\theta, y)}{p(y)} = \frac{p(\theta)p(y \mid \theta)}{p(y)}, \tag{1}$$

or rewritten in an unnormalized version:

$$p(\theta \mid y) \propto p(\theta)p(y \mid \theta). \tag{2}$$

This tells us the relation between *theta*, which is an unobservable vector of variables of interest, and y, which is a vector of observed variables. The left-hand side of the equation is called the posterior distribution, while the right-hand side is a product of prior distribution and the likelihood function. We define the prior predictive distribution as

$$p(y) = \int p(\theta)p(y \mid \theta)\, d\theta, \tag{3}$$

and the posterior predictive distribution as

$$p(\tilde{y} \mid y) = \int p(\tilde{y} \mid \theta)p(\theta \mid y)\, d\theta. \tag{4}$$

The prior predictive distribution is not conditional on the previous observation y of the process and refers to observed data, while the posterior predictive distribution is conditional on y and predicts potential future observations $\tilde{y}$ [22].

Bayesian statistics is widely utilized in the behavioral and social sciences, largely due to the increasing availability of user-friendly software and comprehensive tutorials tailored for scientists in these fields. It is primarily employed for theory development and estimation. This approach is particularly well-suited for these disciplines because meaningful priors can be derived from extensive literature, and informative priors are

valuable for modeling complex behaviors and working with small sample sizes, both of which are common in the social sciences. Bayesian methods are also used in ecological modeling due to their ability to handle complex, high-dimensional, and spatiotemporal models, as well as imperfect or incomplete data. These models often involve computationally expensive likelihoods. Bayesian techniques, such as data augmentation, can fit these models more effectively without requiring oversimplification, which may be necessary in a frequentist framework. Applications of Bayesian statistics in ecology span various scales, from individual organisms to entire ecosystems, and include tasks like understanding population dynamics, modeling spatial patterns, studying population genetics, estimating abundance, and assessing conservation efforts [23].

3.2. Stan Programming

The models were created in Stan. It is a programming language written in C++ and used for statistical inference. It provides a concise way of defining Bayesian models as simple scripts, yet allows for the efficient computation of Markov Chain Monte Carlo methods, which are essential parts of Bayesian inference [24].

The algorithm used in Stan sampling is Hamiltonian Monte Carlo. It is a Markov Chain Monte Carlo (MCMC) method, which uses derivatives of the density function being sampled to generate efficient transitions spanning the posterior distribution. The goal of the sampler is to draw from density $p(\theta \mid y)$, where θ is a vector of parameters and y is a data sample. HMC introduces momentum variables ρ and draws from the joint density:

$$p(\rho, \theta) = p(\rho \mid \theta)p(\theta), \tag{5}$$

$$\rho \sim MultiNormal(0, M), \tag{6}$$

where M is a Euclidean metric.

The joint density $p(\rho, \theta)$ defines a Hamiltonian:

$$\begin{aligned} H(\rho, \theta) &= -logp(\rho, \theta) \\ &= -logp(\rho \mid \theta) - logp(\theta) \\ &= T(\rho \mid \theta) + V(\theta) \end{aligned} \tag{7}$$

where $T(\rho \mid \theta)$ and $V(\theta)$ are called kinetic and potential energy, respectively.

Transitions are generated in two steps. First, a value of momentum is generated independently of the current parameters. Then, the joint system of current parameters and new momentum is defined as Hamilton's equations:

$$\frac{\partial \theta}{\partial t} = \frac{\partial H}{\partial \rho} = \frac{\partial T}{\partial \rho} \tag{8}$$

$$\frac{\partial \rho}{\partial t} = -\frac{\partial H}{\partial \theta} = -\frac{\partial T}{\partial \theta} - \frac{\partial V}{\partial \theta} \tag{9}$$

Since the momentum density is independent of the parameters' density $p(\rho \mid \theta) = p(\rho)$, the term $-\frac{\partial T}{\partial \theta}$ is zero, canceling the first term of the second equation.

Stan's implementation of HMC uses the Leapfrog integrator, as it provides stability for Hamiltonian systems of equations. It starts with sampling new momentum independently of parameters or a previous momentum value. In discrete time steps, denoted as ϵ, it half-step updates momentum and full-step updates parameters:

$$\rho \leftarrow \rho - \frac{\epsilon}{2}\frac{\partial V}{\partial \theta} \tag{10}$$

$$\theta \leftarrow \theta + \epsilon M^{-1}\rho \tag{11}$$

$$\rho \leftarrow \rho - \frac{\epsilon}{2}\frac{\partial V}{\partial \theta} \tag{12}$$

After applying L leapfrog steps, a total of $L\epsilon$ time is simulated. The resulting state of the simulation is denoted as (ρ^*, θ^*). Lastly, the proposal (ρ^*, θ^*) generated by the transition from (ρ, θ) has a probability of being accepted defined as

$$min(1, exp(H(\rho, \theta) - H(\rho^*, \theta^*))) \tag{13}$$

If the proposal is not accepted, the previous parameter value θ is utilized in the next iteration [24].

Stan is able to automatically optimize ϵ to match an acceptance-rate target, able to estimate M based on warmup sample iterations, and able to dynamically adapt L on the fly during sampling (and during warmup). This helps to mitigate the risks associated with divergence caused by improper algorithm-parameter selection [24].

3.3. Data

The data used for inference come from Kaggle [25]. Data preprocessing was performed in two steps. The first one was generating the shortest route between an origin and destination pair. It was necessary as the geographical coordinates presented in the raw data were unsuitable to create a meaningful probability distribution for our models. It was performed using OSRM API. The second step was data cleaning and analysis. We decided to remove coordinates outside of India's geographical boundaries, which was the country where the data originated from. We computed the meal-preparation time as the difference between the time when the order was picked up by the courier and the time when the order was made in the restaurant. Similar to coordinates, timestamps would provide difficulties when trying to associate them with distributions. We used z-score standardization for numerical variables. It was necessary to do so, as our models use an exponential function on a linear combination of predictors, and if the latter were too large, it caused numerical problems with computation. Finally, we mapped categorical variables to numerical indices, which would be used to associate the category with its corresponding distribution. In-depth preprocessing was described in part 1 of this article [26]. The chosen features are presented in Table 1. There were 45,593 raw data samples. After processing, we ended with 34,920, which will be further denoted as N. Histograms of the data are presented in Figure 1 [26].

Table 1. Features computed from dataset.

Model Variable	Data Type	Description	Obtained From
distance	Vector of floats	Standardized [1] route distances.	Computed via OSRM API [26].
traffic_level	Array of integers	Mapping categorical traffic level to number (1—jam, 2—high, 3—medium, 4—low).	Provided in raw dataset.
meal_preparation_time	Vector of floats	Standardized [1] meal-preparation times.	Difference between order date and pickup by the courier, both of which were in raw data.
delivery_person_rating	Vector of floats	Standardized [1] rating of delivery person.	Provided in raw dataset.
number_of_deliveries	Array of integers	Number of deliveries.	Provided in raw dataset.

[1] Standardization was conducted in preprocessing step [26].

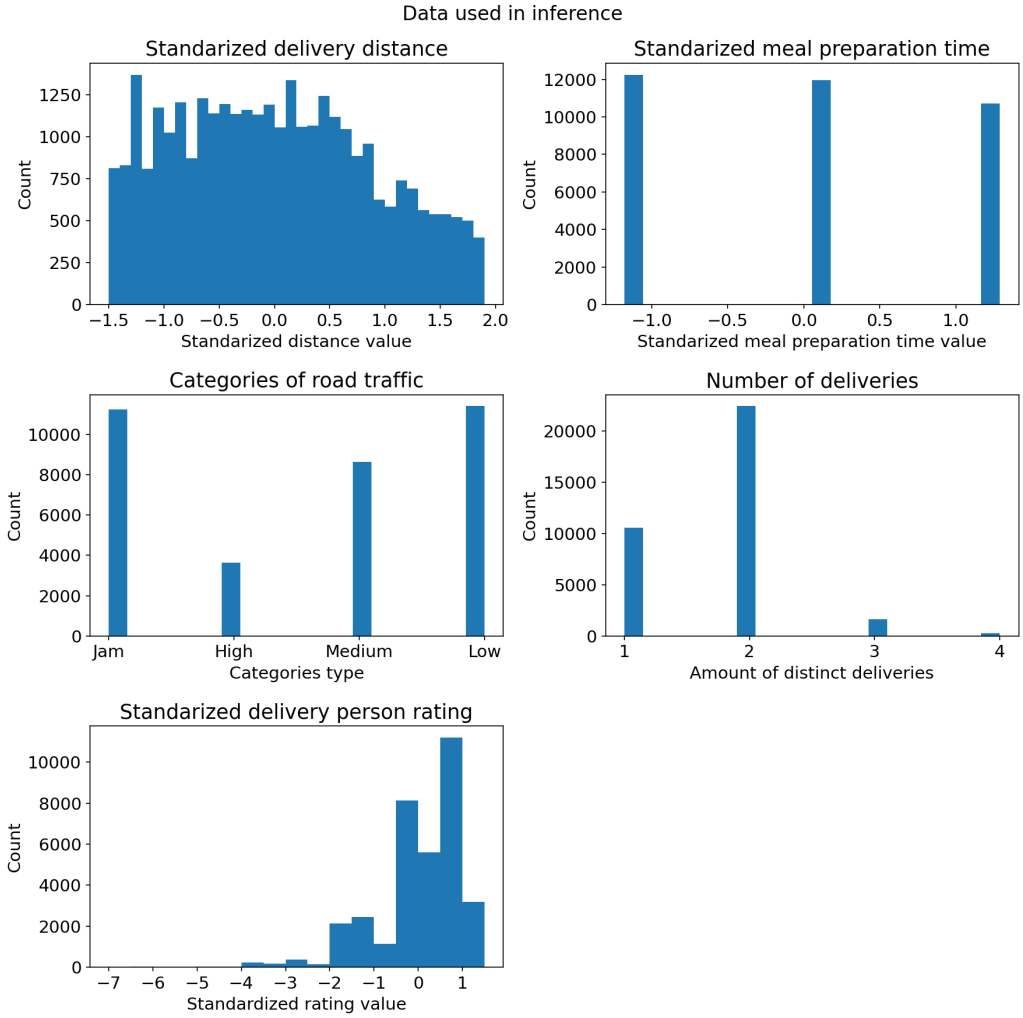

Figure 1. Histograms of data used in inference. Standardization was computed as z-score. X-axis represents value of the predictor and Y-axis is their count for predefined bins. (**top-left**) Standardized distance, which is z-score of distance data received from OSRM API. Raw distances were limited to 30 km. (**top-right**) Standardized meal-preparation time, which is z-score of meal-preparation time. Meal-preparation time was calculated as difference between time the order was received and the time when courier picked up delivery. (**center-left**) Categories of road traffic, which are raw categorical data describing traffic conditions during each delivery. It can be one of four states: low, medium, high, and jam. (**center-right**) Distinct deliveries count, which describes number of deliveries that courier had to make during his trip. (**botom-left**) Standardized delivery-person rating, which is z-score of the delivery-person rating. Original data had rating in range of 2.5 and 5.0 with 0.1 quantization.

3.4. Models

Both models are generalized linear models. We defined the linear predictor as $\eta = X\beta$, where X denotes the vector of features described in Table 1 and B is the vector of coefficients. Each coefficient's distribution is described in the appropriate model section. Both vectors are size Nx1 [22].

We then used the logarithmic link function to transform the linear predictor's domain to positive real numbers. It was one of the possible options, but nevertheless necessary, as both models are defined by the inverse gamma function. This way, we obtained the explanatory variable μ_i, representing the mean of the outcome variable [22]. We defined the prior distribution for the standard deviation of our models, denoted as σ, to be an exponential distribution with a rate parameter equal to 0.5.

Lastly, we defined likelihood as an inverse gamma distribution with parameter shape (α) and scale (β), computed from μ and σ in such a way that the resulting distribution had a mean and standard deviation of μ and σ, respectively. The reasoning behind this particular distribution was to model the skewness of the data effectively. Also, time has to be strictly positive and continuous, which the inverse gamma also provides. The variables are defined in Table 2, and the predictors are defined in Table 1.

Table 2. Models' variables.

Model Variable	Explanation
$delivery_time_i$	Posterior distribution of delivery time, defined by inverse gamma distribution.
α_i	Shape parameter of inverse gamma function, computed from mean and standard deviation of this distribution.
β_i	Scale parameter of inverse gamma function, computed from mean and standard deviation of this distribution.
σ_i	Prior distribution of standard deviation of model, defined by exponential distribution.
μ_i	Generalized linear model with logarithmic link function. Represents mean of the delivery times as a function of selected predictors.
$mean_i$	Prior distribution of the average meal-delivery scenario. Defined by normal distribution. Acts as an intercept to linear model, since after z-score standardization, the most average case of delivery would yield exp(0), which would invalidate model.
$distance_coeff_i$	Prior distribution of the distance's linear coefficient. Defined by normal distribution. Used to represent influence of this predictor on the model output.
$meal_prep_coeff_i$	Prior distribution of the meal-preparation time's linear coefficient. Defined by normal distribution. Used to represent influence of this predictor on the model output.
$traffic_level_coeff_i[j]$ [1]	Prior distributions of the traffic level's linear coefficient. Defined by normal distributions. Used to represent influence of these predictors on the model output.
$person_rating_coeff_i$	Prior distribution of the delivery person's rating's linear coefficient. Defined by normal distribution. Used to represent influence of this predictor on the model output.
$deliveries_number_coeff_i[j]$ [1]	Prior distributions of the delivery number's linear coefficient. Defined by normal distributions. Used to represent influence of these predictors on the model output.

[1] Here, predictor acts as an index variable.

While the models themselves were defined in Stan, the experiments were conducted via CmdStanPy, which is one of Python's interfaces for it [24]. For each sample from the dataset, the entire inference process was performed. It consisted of 1000 warmup iterations and 1000 regular iterations. The warmup iterations were discarded. Each equation

described in the models' sections below was computed according to formulas (denoted by the equal operator), while sampling was performed with HMC (see Section 3.2, (denoted by the tilde operator).

3.4.1. Model 1

The first model is defined as follows:

$$delivery_time_i \sim InverseGamma(\alpha_i, \beta_i) \tag{14}$$

$$\alpha_i = \frac{\mu_i^2}{\sigma_i^2} + 2 \tag{15}$$

$$\beta_i = \frac{\mu_i^3}{\sigma_i^2} + \mu_i \tag{16}$$

$$\sigma_i \sim Exponential(0.5) \tag{17}$$

$$\begin{aligned} \mu_i = \exp(&distance_coeff_i \cdot distance_i + traffic_level_coeff[traffic_level_i] + \\ &+ meal_prep_coeff_i \cdot meal_preparation_time_i + mean_i) \end{aligned} \tag{18}$$

$$mean_i \sim N(3, 0.1) \tag{19}$$

$$distance_coeff_i \sim Normal(0, 0.3) \tag{20}$$

$$meal_prep_coeff_i \sim Normal(0, 0.3) \tag{21}$$

$$traffic_level_coeff[1] \sim Normal(0, 0.3) \tag{22}$$

$$traffic_level_coeff[2] \sim Normal(0, 0.3) \tag{23}$$

$$traffic_level_coeff[3] \sim Normal(0, 0.3) \tag{24}$$

$$traffic_level_coeff[4] \sim Normal(0, 0.3) \tag{25}$$

3.4.2. Model 2

The second model is an extension of the first model by two predictors: the number of deliveries and standardized delivery-person rating. As such, we only present the changes necessary to create Model 2 out of Model 1:

$$\begin{aligned} \mu_i = \exp(&distance_coeff_i \cdot distance_i + traffic_level_coeff[traffic_level_i] + \\ &+ meal_prep_coeff_i \cdot meal_preparation_time_i + \\ &+ deliveries_number_coeff[number_of_deliveries_i] + \\ &+ person_rating_coeff_i \cdot delivery_person_rating_i + mean_i) \end{aligned} \tag{26}$$

$$person_rating_coeff_i \sim Normal(0, 0.3) \tag{27}$$

$$deliveries_number_coeff[1] \sim Normal(0, 0.3) \tag{28}$$

$$deliveries_number_coeff[2] \sim Normal(0, 0.3) \tag{29}$$

$$deliveries_number_coeff[3] \sim Normal(0, 0.3) \tag{30}$$

$$deliveries_number_coeff[4] \sim Normal(0, 0.3) \tag{31}$$

3.4.3. Priors and Prior Predictive Checks

We decided to use unbounded weakly informative priors for all parameters for two main reasons. First, we lack expert knowledge on the influence of each feature. Second, the abundance of data reduces the influence of priors on the final distribution as more data points are added.

We chose a normal distribution with a mean of 0 and a standard deviation of 0.3 for our parameters. This distribution provides a value range of approximately -1 to 1, with most values likely clustering around 0. This choice reflects our initial assumption that each parameter is not highly influential while still allowing for a small probability that they could be significantly influential. The standard deviation of 0.3 was selected to accommodate the exponential distribution coming from the inverse link function, as larger values from the linear combination could numerically destabilize the model.

The exception to this is the intercept parameter $mean_i$, for which we chose a strong prior: $N(3, 0.1)$. It results in the base mean delivery time around 15–30 min. We opted for a strong prior here to ensure that our model accurately reflects the most average delivery time. Given that our features are standardized with a mean of 0, the linear combination for the most average case would be close to zero, leading to an unrealistic delivery time in the posterior distribution. The $mean_i$ prior helps anchor the model, providing a trusted average time for an average case.

For both models, prior predictive checks gave good results, i.e., the observed data were included within the simulated data range, and no outright impossible values were generated from either of the models. They can be observed in Figures 2–7.

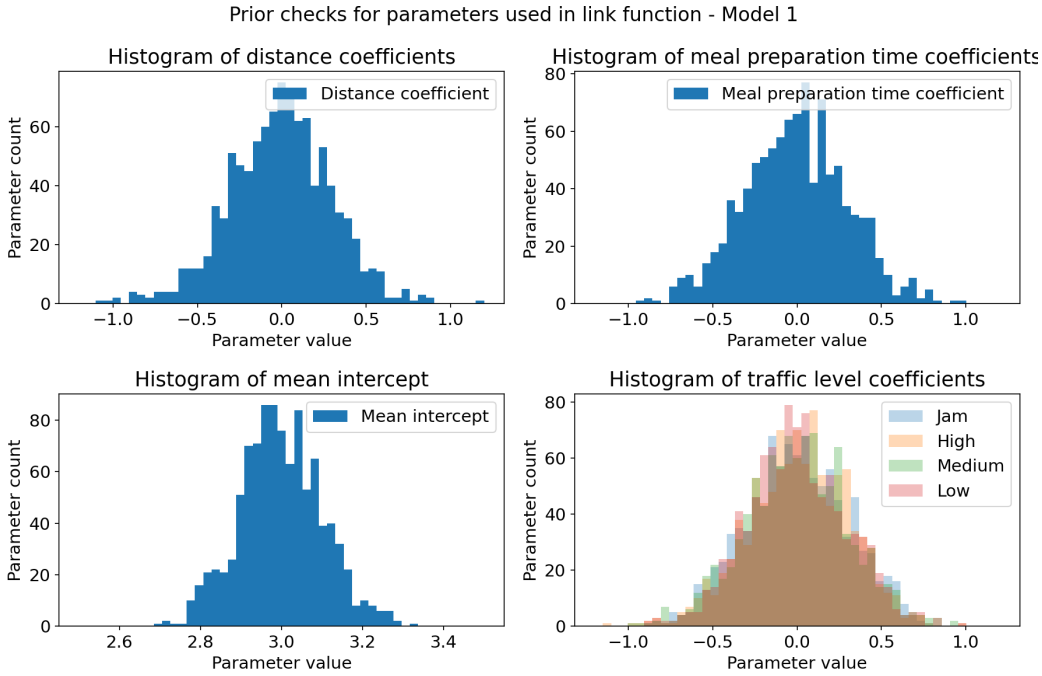

Figure 2. Sampling check for prior distributions of Model 1's link-function parameters (parameters with _coeff suffix). X-axis represents coefficient values and Y-axis represents sample count. Each of the coefficients follows its distribution, which is necessary for prior check to be successful. (**top-left**) Prior distribution of distance coefficient, defined as $Normal(0, 0.3)$. (**top-right**) Prior distribution of meal-preparation-time coefficient, defined as $Normal(0, 0.3)$. (**bottom-left**) Prior distribution of *mean* parameter, defined as $Normal(3, 0.1)$. *mean* parameter represents our belief of what mean delivery time should be in case all other parameters are 0. (**bottom-right**) Joint plot of prior distributions of traffic-level coefficients, all defined as $Normal(0, 0.3)$.

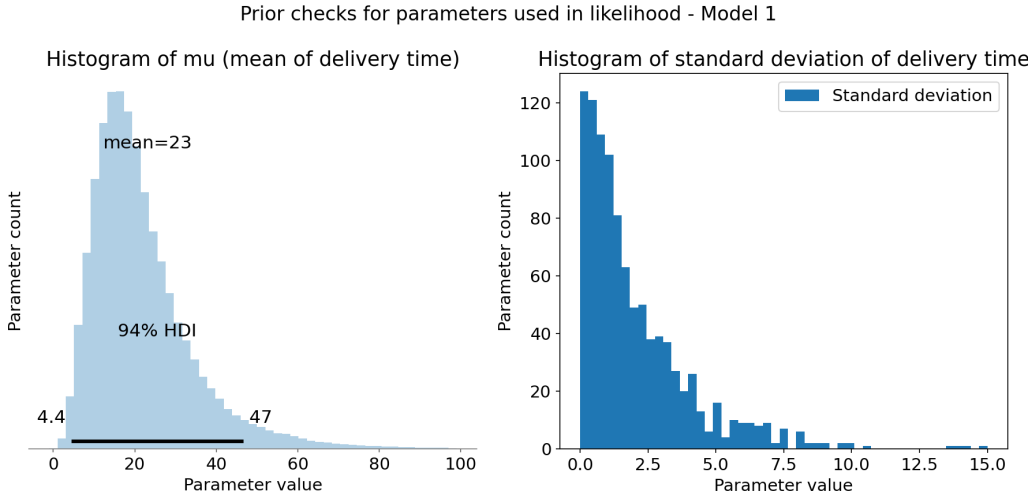

Figure 3. Computation and sample check of Model 1's likelihood parameters. X-axis is time in min and Y-axis represents sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 4.4 to 47 min. Mean of this distribution (at the top of the plot) is 23 min, which is reasonable value. (**right**) Prior distribution of standard deviation of the model, defined as $Exponential(0.5)$.

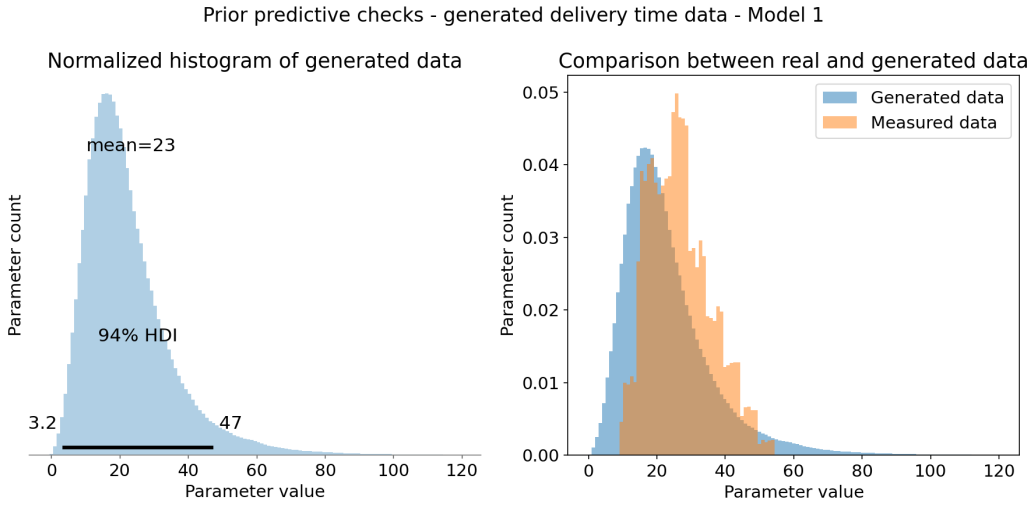

Figure 4. Prior predictive checks—Model 1. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 3.2 to 47 min, which is broad range. Mean of this distribution (at the top of the plot) is 23 min, which is reasonable value. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data are included within generated data, which means that all observations are possible within prior model. This means that prior checks are successful.

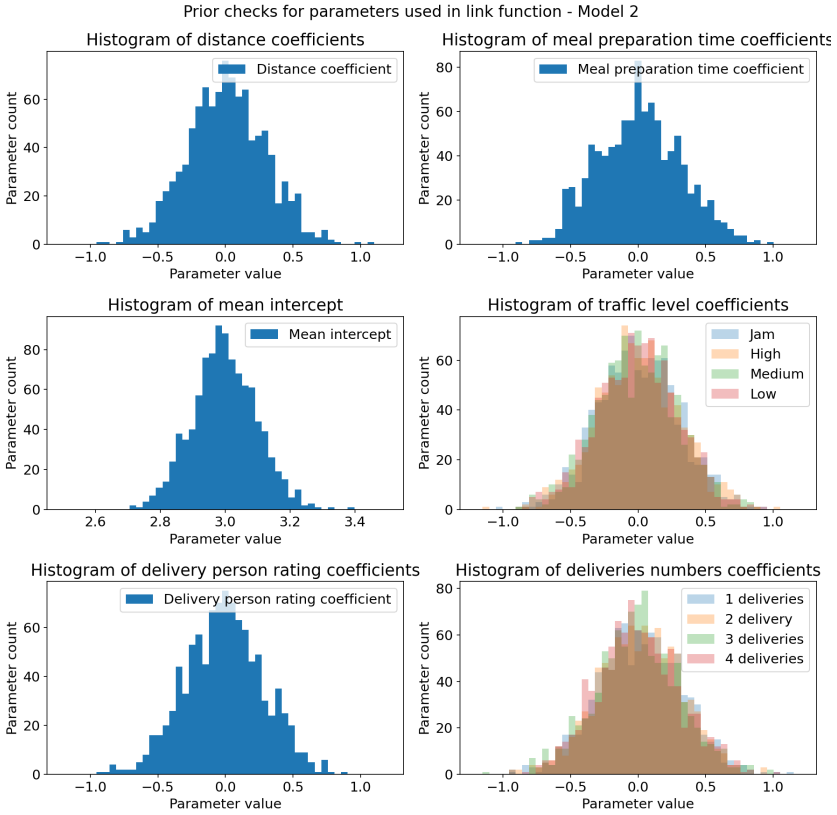

Figure 5. Sampling check for prior distributions of Model 2's link-function parameters (parameters with _coeff suffix). *X*-axis represents coefficient values and Y-axis represents sample count. Each of the coefficients follows its distribution, which is necessary for prior check to be successful. (**top-left**) Prior distribution of distance coefficient, defined as $Normal(0, 0.3)$. (**top-right**) Prior distribution of meal-preparation-time coefficient, defined as $Normal(0, 0.3)$. (**center-left**) Prior distribution of *mean* parameter, defined as $Normal(3, 0.1)$. *mean* parameter represents our belief of what mean delivery time should be in case all other parameters are 0. (**center-right**) Joint plot of prior distributions of traffic-level coefficients, all defined as $Normal(0, 0.3)$. (**bottom-left**) Prior distribution of delivery-person-rating coefficient, defined as $Normal(0, 0.3)$. (**bottom-right**) Joint plot of prior distributions of deliveries-number coefficients, all defined as $Normal(0, 0.3)$.

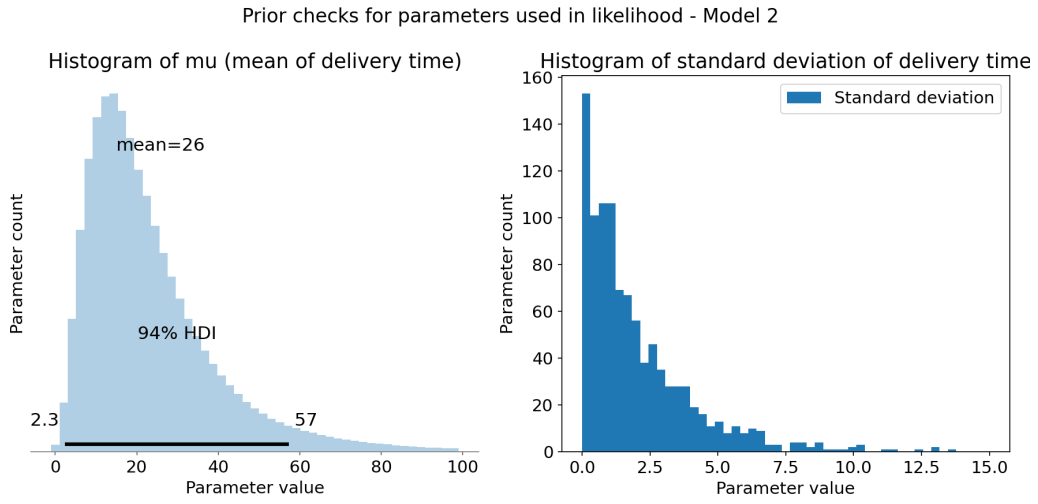

Figure 6. Computation and sample check of Model 2's likelihood parameters. X-axis is time in min and Y-axis is sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI

94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 2.3 to 57 min. Mean of this distribution (at the top of the plot) is 26 min, more than for Model 1, but still within reasonable range. (**right**) Prior distribution of standard deviation of the model, defined as $Exponential(0.5)$.

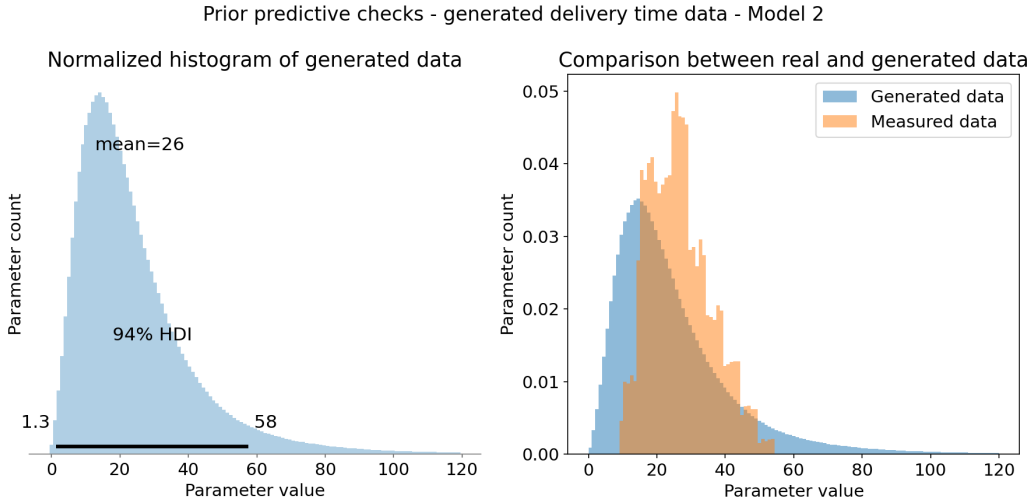

Figure 7. Prior predictive checks—Model 2. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 1.3 to 58 min. It is very broad, improbable range, but for prior checks it is sufficient. Mean of this distribution (at the top of the plot) is 26 min, which is reasonable value. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data are included within generated data, which means that all observations are possible within prior model. This means that prior checks are successful.

4. Results

In this section, we present the posterior distributions of our models. Each one was trained on full data. There were 1000 warmup and 1000 sampling iterations performed on four parallel chains. The selected algorithm was Hamiltonian Monte Carlo with the engine No-U-Turn Sampler.. For computation Stan 2.34 was used.

Posterior predictive checks were performed by simulating new data from the posterior distribution obtained during model training. These simulations were used to verify if the simulated data resembled the original data, with histograms chosen for comparison purposes [23]. The link-function parameters were subjectively assessed for their numerical influence, considering the z-score standardization of predictors. This standardization implies that the average delivery time corresponds to a predictor value of zero. Narrow distributions indicate a near-constant predictor effect, while wide distributions suggest uncertainty in the predictor's impact. Positive values reflect a direct relationship, and negative values indicate an inverse relationship between the predictor and the outcome. The parameters in the likelihood function were evaluated for their plausibility in real-life scenarios. Since the parameters of our models are coefficients of a linear function, rather than the priors for the data distributions themselves, we believe that the subjective interpretation of them is justified.

4.1. Posterior Predictive Checks for Model 1

Model 1 gave decent results. All of the observed data fall within the samples from the posterior distribution, and visual overlap is quite high. The posterior distribution exhibits a long tail, which is the drawback of using the inverse gamma function. The data are represented in Figure 8.

The model coefficients for the distance and meal-preparation time ended with very narrow distributions, albeit positive ones, indicating that they impact the output variable.

The mean intercept parameter ended with a mean closer to 3.1, which is also closer to the mean of the dataset ($e^{3.1} \approx 22.18$ while the mean of the dataset is ≈27.05). The traffic-level coefficient represents a trend in which low traffic contributes to faster delivery times, and as the traffic level increases, the delivery times become longer. This interpretation is viable as it is not a multiplicand but a sum component, so negative values will result in a smaller mean and positive values in a larger mean. There is almost no distinction between the influence of high traffic and jams. The data are represented in Figure 9.

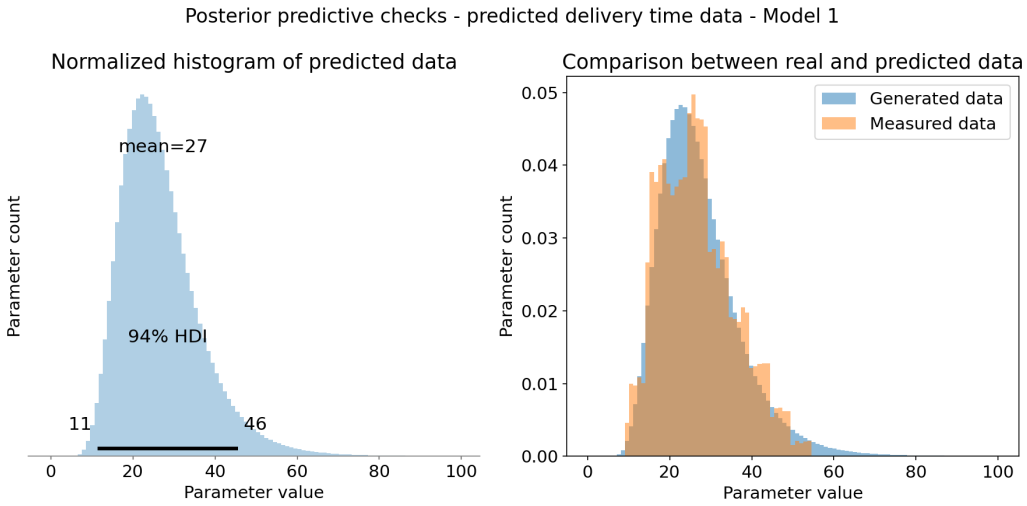

Figure 8. Posterior predictive checks—Model 1. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 11 to 46 min. It is broad range, but realistic nevertheless. Mean of this distribution (at the top of the plot) is 27 min, which is reasonable value. It follows inverse gamma distribution as defined. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data have high overlap with sampled data from posterior distribution.

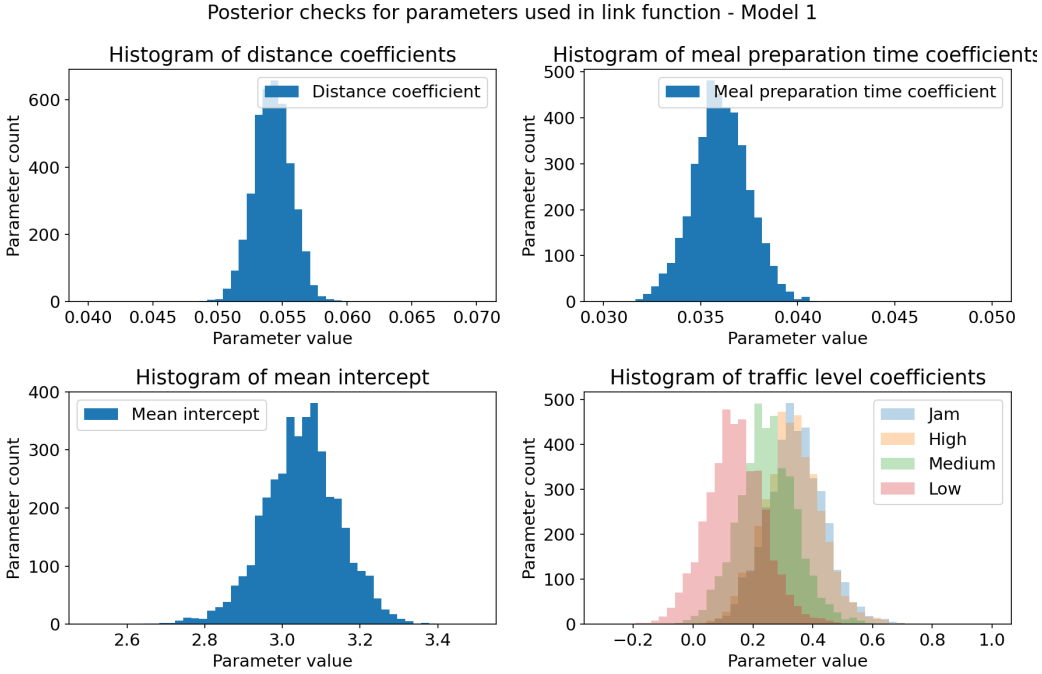

Figure 9. Sampling check for posterior distributions of Model 1's link-function parameters. X-axis represents coefficient values and Y-axis represents sample count. (**top-left**) Posterior distribution of

distance coefficient. It is much narrower than prior distribution, but still follows normal distribution. Positive value indicates that it has impact on the output variable. (**top-right**) Posterior distribution of meal-preparation-time coefficient. Conclusions are the same as for the distance coefficient. (**bottom-left**) Posterior distribution of *mean* parameter. It has mean closer to 3.1, which more likely represents mean of the dataset. (**bottom-right**) Joint plot of posterior distributions of traffic-level coefficients. The bigger the traffic level, the more impact it has on the outcome variable. Jams and high levels have the same impact.

The linear model of mean delivery times μ_i results in probable values with respect to the dataset. The standard deviation completely changed its distribution and now follows a normal distribution centered around 9.5. It is quite close to the std of the dataset, which is ≈8.99. The data are represented in Figure 10.

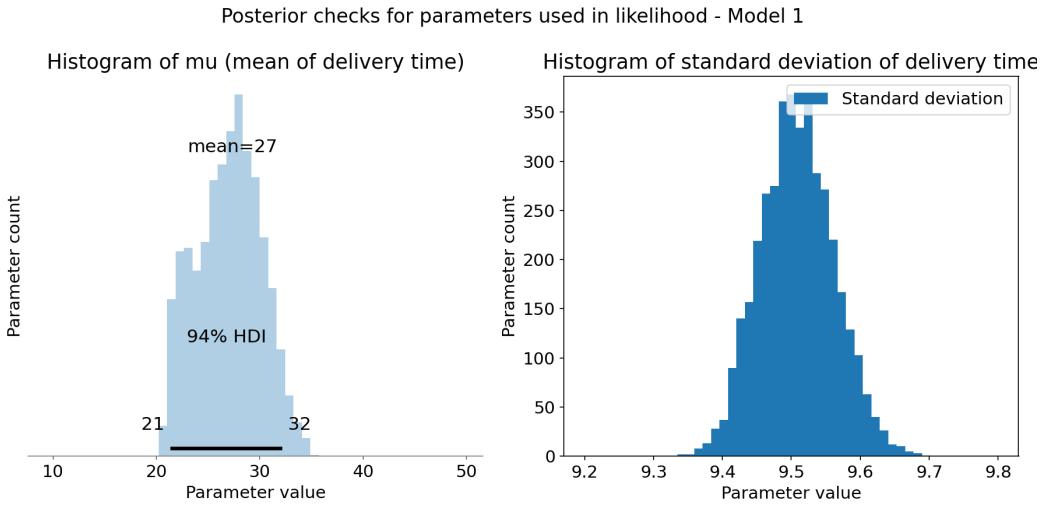

Figure 10. Computation and sample check of Model 1's likelihood parameters. X-axis represents coefficient values and Y-axis represents sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 21 to 32 min. Those are much more realistic values than the ones from prior distribution. Mean of this distribution (at the top of the plot) is 27 min, a reasonable value. (**right**) Posterior distribution of standard deviation of the model; it no longer resembles prior, and now it follows normal distribution with mean ≈9.5.

4.2. Posterior Predictive Checks for Model 2

Model 2 gave visually better results. All of the observed data fall within the samples from the posterior distribution, as with Model 1. The tail is shorter than in Model 1. The data are represented in Figure 11. Since there is not much difference between the posterior distributions for the shared features of both models, we will only comment on new features, distinct to Model 2, as well as on likelihood-related parameters.

The delivery-person rating follows a narrow normal distribution centered around −0.085. Since it is negative and has a small std, we can reason that the delivery-person rating is inversely related to delivery time. This is expected as couriers with higher scores are more likely to deliver food faster. The number of deliveries follows exactly the same trend as traffic-level coefficients, but numerically is more important as the values range is greater. The data are represented in Figure 12.

The linear model of the mean delivery times μ_i has a much longer tail with regard to Model 1, which results in a different 94% HDI. The standard deviation has a similar distribution to Model 1, although its mean value is smaller, around 7.85. The data are represented in Figures 11 and 13.

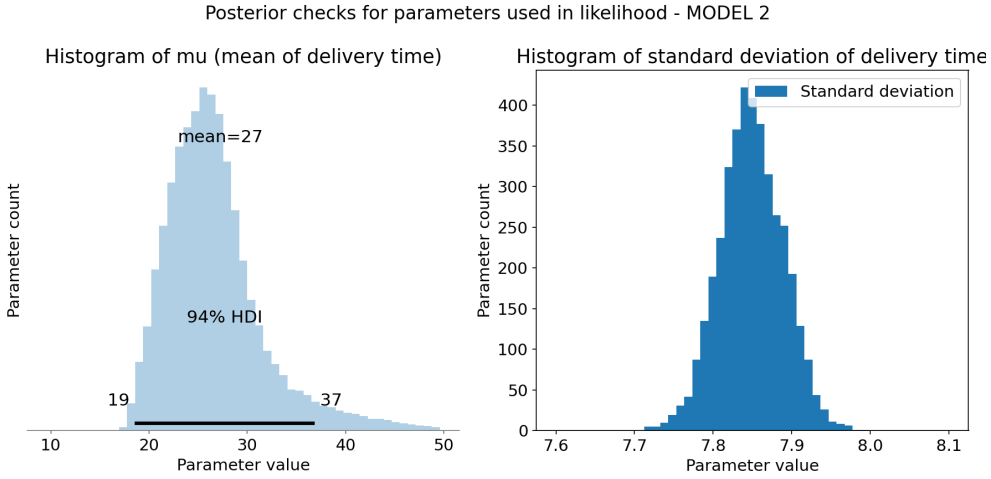

Figure 11. Computation and sample check of Model 2's likelihood parameters. X-axis represents coefficient values and Y-axis represents sample count. (**left**) Computed μ represents mean delivery time for each sample. HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 19 to 37 min. Those are much more realistic values than the ones from prior distribution. Mean of this distribution (at the top of the plot) is 27 min, a reasonable value. (**right**) Posterior distribution of standard deviation of the model; it no longer resembles prior, and now it follows normal distribution with mean ≈7.85.

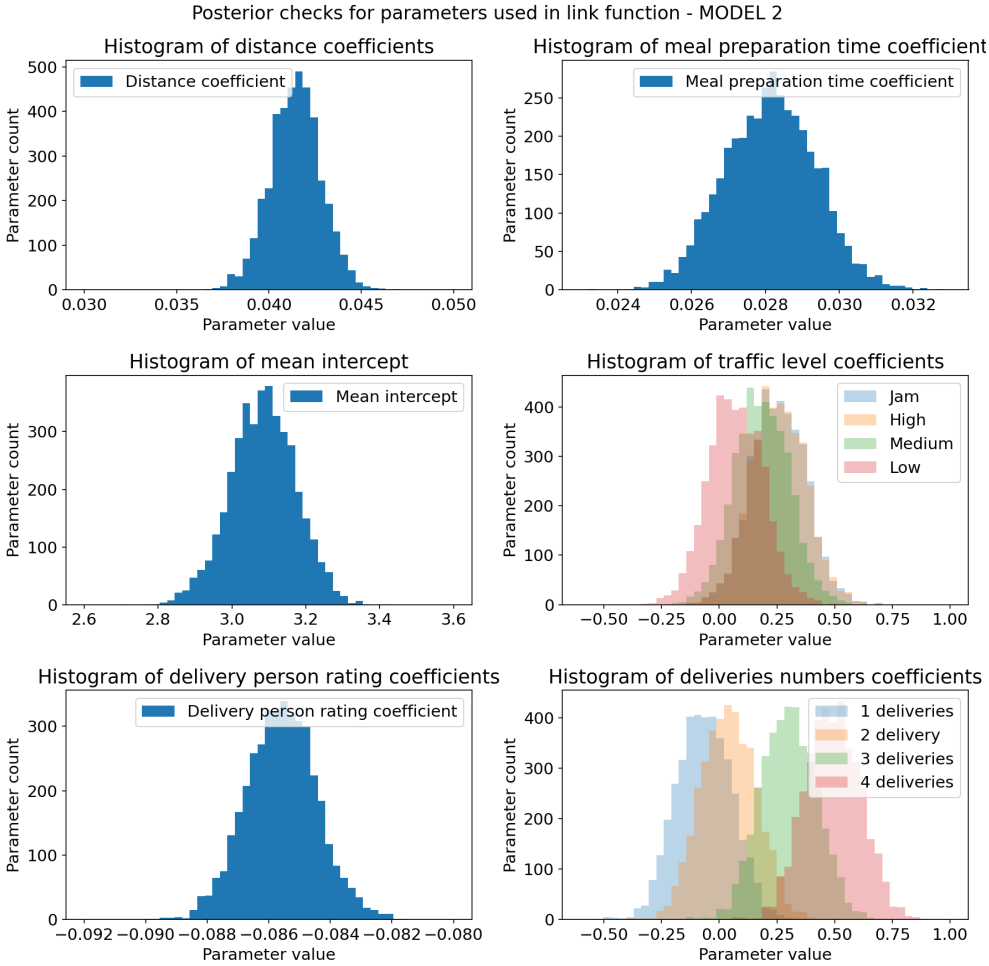

Figure 12. Sampling check for posterior distributions of Model 2's link-function parameters. X-axis represents coefficient values and Y-axis represents sample count. (**top-left**) Posterior distribution of

distance coefficient. It is much narrower than prior distribution, but still follows normal distribution. Positive value indicates that it has impact on the output variable. (**top-right**) Posterior distribution of meal-preparation-time coefficient. Conclusions are the same as for the distance coefficient. (**center-left**) Posterior distribution of *mean* parameter. It has mean closer to 3.1, which more likely represents mean of the dataset. (**center-right**) Joint plot of posterior distributions of the traffic-level coefficients. The bigger the traffic level, the more impact it has on the outcome variable. Jams and high levels have the same impact. (**bottom-left**) Posterior distribution of the delivery-person-rating coefficient. It is much narrower than prior distribution, but still follows normal distribution. Negative values indicate inverse relationship between delivery time and rating; the bigger the courier rating, the faster delivery will be made. (**bottom-right**) Joint plot of posterior distributions of deliveries-number coefficients. The more deliveries, the more impact it has on the outcome variable. This is the same trend as for the traffic level, but greater range translates to greater impact.

4.3. Model Comparison

The models were compared using the WAIC and PSIS-LOO criteria using ArviZ library for an exploratory analysis of Bayesian Models [27].

The WAIC (Widely Applicable or Watanabe–Akaike Information Criterion) is a statistical measure used to estimate the out-of-sample predictive accuracy of a model. It does this by evaluating the within-sample predictive accuracy and making necessary adjustments. The WAIC calculates the log pointwise posterior predictive density (LPPD) and includes a correction for the effective number of parameters to account for overfitting. This correction is performed by subtracting the sum of the posterior variances of the log predictive densities for each data point [28].

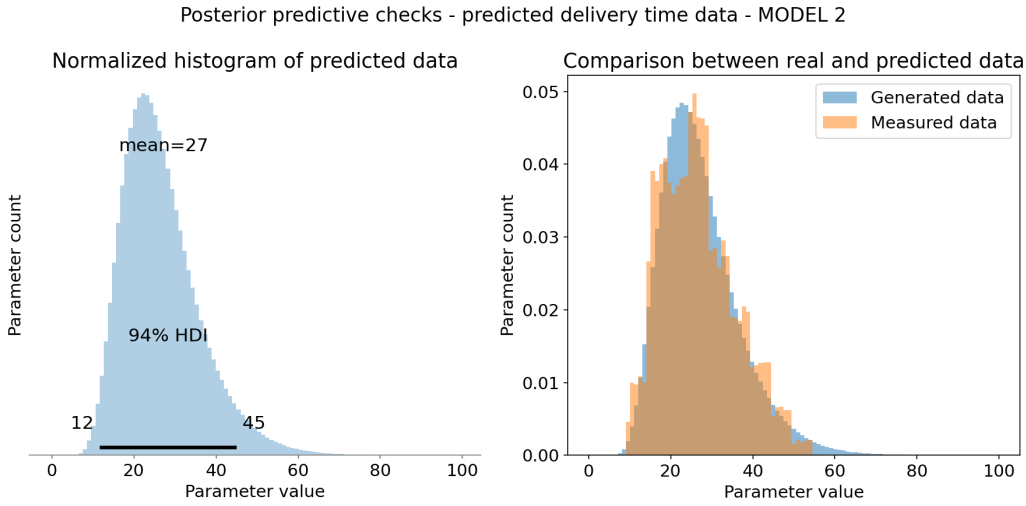

Figure 13. Posterior predictive checks—Model 2. (**left**) HDI 94% is represented as black bar at the bottom of the plot and tells us that 94% of shown mean times fall in range of 12 to 45 min. It is slightly narrower than Model 1 range, but realistic nevertheless. Mean of this distribution (at the top of the plot) is 27 min, which is reasonable value. It follows inverse gamma distribution as defined. (**right**) Real and simulated data overlay. Both are normalized so that integral of the graph is 1. It was necessary for comparison. Measured data have high overlap with sampled data from posterior distribution. Generated data have shorter tail than Model 1, which is desirable.

Model 2 has a higher ELPD score (denoted as waic), which indicates its better within-sample fit. The WAIC also correctly states that it has a higher number of effective parameters (p_waic). The weight parameter clearly states that Model 2 has nearly one probability within the given data. It is slightly more uncertain than Model 1, which is indicated by the SE (standard error) parameter, but when compared to differences in the WAIC score

and size of the dataset, it is not overly large. Overall, the WAIC clearly evaluated Model 2 as superior. It is presented in the Table 3 and Figure 14.

Table 3. Comparison results with WAIC.

Model	Rank	waic	p_waic	d_waic	Weight	SE	dSE
2	0	−117,249.732164	14.307179	0.000000	0.997408	145.013769	0.000000
1	1	−122,442.256308	9.314501	5192.524145	0.002592	137.412164	88.510163

Parameters description [29]: Model: indicates whether it is Model 1 or Model 2. Rank: rank of the models, with 0 indicating the best model. waic: ELPD score, where higher values suggest better out-of-sample predictive fit. p_waic: estimated effective number of model parameters. d_waic: difference between ELPD scores, relative to the best model. Weight: relative weight of each model, interpreted as the probability of the model given the data. SE: standard error of the ELPD estimate. dSE: standard error of the difference in ELPD between each model and the top-ranked model.

The PSIS-LOO (Pareto Smoothed Importance Sampling using Leave-One-Out validation) method is used to calculate the out-of-sample predictive fit by summing the log leave-one-out predictive densities. These densities are evaluated using importance ratios (IS-LOO). However, the importance ratios can exhibit high or infinite variance, leading to instability in the estimates. To mitigate this issue, a generalized Pareto distribution is fitted to the largest 20% of the importance ratios [28].

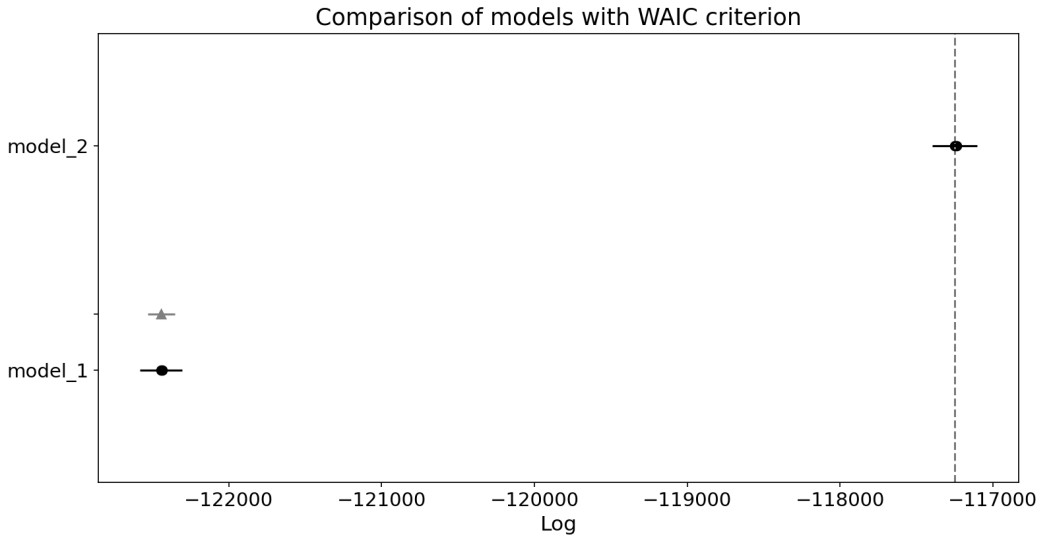

Figure 14. Comparison plot for WAIC. Black dots indicate ELPD of each model with their standard error (black lines). Grey triangle represents standard error of difference in ELPD between Model 1 and top-ranked Model 2. Plot indicates that Model 2 performs better with a dashed line.

The PSIS-LOO evaluation provides near identical results as the WAIC, and the same conclusions as above can be drawn. It is presented in the Table 4 and Figure 15. An alternative approach for comparison is of course statistical significance testing, using, for instance, Kolmogorov–Smirnov-based tests (see, for example, [30]). This approach is very popular in applications that rely on frequentist statistics. Our work, however, is based on a Bayesian paradigm, where classical statistical significance or hypothesis testing is not as interpretable; this is why we decided to not include them in this paper.

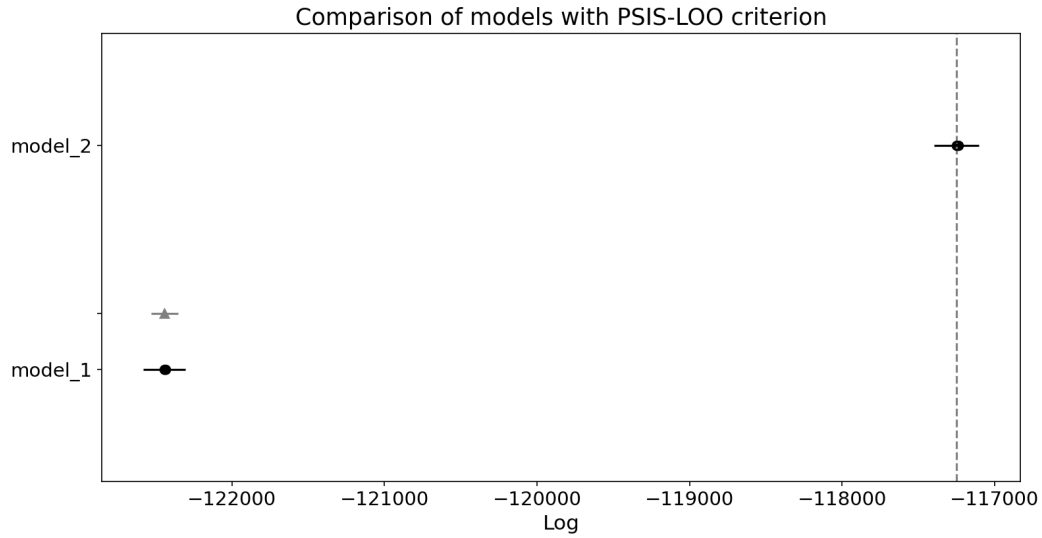

Figure 15. Comparison plot for PSIS-LOO criterion. Black dots indicate ELPD of each model with their standard error (black lines). Grey triangle represents standard error of difference in ELPD between Model 1 and top-ranked Model 2. Plot indicates that Model 2 performs better with a dashed line.

Table 4. Comparison result with PSIS-LOO criterion.

Model	Rank	loo	p_loo	d_loo	Weight	SE	dSE
2	0	−117,249.732784	14.307799	0.000000	0.997408	145.013813	0.000000
1	1	−122,442.256329	9.314522	5192.523545	0.002592	137.412164	88.510227

Parameters description [29]: Model: indicates whether it is Model 1 or Model 2. Rank: rank of the models, with 0 indicating the best model. loo: ELPD score, where higher values suggest better out-of-sample predictive fit. p_loo: estimated effective number of model parameters. d_loo: difference between ELPD scores, relative to the best model. Weight: relative weight of each model, interpreted as the probability of the model given the data. SE: standard error of the ELPD estimate. dSE: standard error of the difference in ELPD between each model and the top-ranked model.

5. Discussion

One of the potential limitations is scaling models to a larger dataset. MCMC methods are computationally expensive and time consuming. For the presented data, it took around 7 h on an Intel Core i7-11370H 3.30 GHz chip to run the inference. It also produced approx. 3.7 GB of data per model.

It is important to recognize that the methodology of Bayesian inference itself has inherent limitations. Bayesian inference is optimal when the assumed model is correct. However, since models are never perfect and only approximate reality, every model introduces implicit limitations. Additionally, models are influenced by the subjective choice of priors. Very diffuse or uniform priors can lead to overcertainty in estimates, while strong priors that do not accurately represent the true probability distribution of the data can result in poor model generalization. Furthermore, all models require subjective interpretation and decisions by researchers, which, if not properly justified, can be a significant shortcoming [23].

6. Conclusions

In this paper, we explored the application of Bayesian inference for predicting food-delivery times, a novel approach not previously employed for this specific task to the best of our knowledge. Our results indicate significant potential in this methodology, particularly with Model 2, which, as an extension of Model 1, demonstrated a superior performance.

A major advantage of our approach is its ability to capture model uncertainty and provide interpretability, as well as to assess the impact of predictors, thereby offering insights into areas of improvement for food-delivery companies. We hope that our findings empower future exploration of Bayesian methods. For the field of time-travel estimation, this work could serve as a baseline for researchers to build upon and refine. In other areas, we aim to demonstrate an accessible approach and encourage scientists to experiment with it.

Future research should aim to test these models on data from more reputable sources. This endeavor may prove challenging, as food-delivery data are often proprietary and not publicly accessible. Additionally, it is crucial to validate the models with out-of-sample datasets to ensure robustness. Expert input on prior selection should also be considered. Finally, the business applications of these findings merit consideration, both for historical data analysis and real-time implementation. Further work could involve developing additional models with diverse datasets to explore predictor relationships. This deeper understanding could potentially enhance feature selection in machine learning models, which are commonly used for the ETA task.

Author Contributions: Conceptualization, J.P., J.G., and J.B.; methodology, J.P., J.G., and J.B.; software, J.P. and J.G.; validation, J.P. and J.B.; formal analysis, J.P.; investigation, J.P.; resources, J.B.; data curation, J.P. and J.G.; writing—original draft preparation, J.P.; writing—review and editing, J.P. and J.B.; visualization, J.P.; supervision, J.B.; project administration, J.B.; funding acquisition, J.B. All authors have read and agreed to the published version of the manuscript.

Funding: The first author's work was supported by AGH's Research University Excellence Initiative under the project "Research Education Track". The third author's work was partially realized in the scope of a project titled "Process Fault Prediction and Detection". This project was financed by The National Science Centre on the base of decision no. UMO-2021/41/B/ST7/03851. Part of this work was funded by AGH's Research University Excellence Initiative under the project "DUDU—Diagnostyka Uszkodzeń i Degradacji Urządzeń".

Data Availability Statement: All code prepared as part of this project is available in the repository https://github.com/JohnnyBeet/Food-delivery-time-prediction/tree/modelling. (accessed on 25 August 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OFD	Online food delivery
GPS	Global Positioning System
PSIS-LOO	Pareto Smoothed Importance Sampling using Leave-One-Out validation
WAIC	Watanabe–Akaike Information Criterion
DNN	Deep Neural Network
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
ETA	Estimated Time of Arrival
OFCT	Order Fulfillment Cycle Time
OD	Origin–destination
OSRM	Open-Source Routing Machine
HDI	Highest-Density Interval
LPPD	Log pointwise posterior predictive density
ELPD	Expected Log pointwise Predictive Density
SE	Standard error
dSE	Standard error of the difference in ELPD between each model
MCMC	Markov Chain Monte Carlo
HMC	Hamiltonian Monte Carlo

References

1. Statista. Online Food Delivery—Worldwide. 2024. Available online: https://www.statista.com/outlook/emo/online-food-delivery/worldwide (accessed on 4 May 2024).
2. IMARC Group. India Online Food Delivery Market Report. 2023. Available online: https://www.imarcgroup.com/india-online-food-delivery-market (accessed on 4 May 2024).
3. Alalwan, A.A. Mobile food ordering apps: An empirical study of the factors affecting customer e-satisfaction and continued intention to reuse. *Int. J. Inf. Manag.* **2020**, *36*, 28–44. [CrossRef]
4. Xue, G.; Wang, Z.; Wang, Y. The restaurant delivery problem with uncertain cooking time and travel time. *Comput. Ind. Eng.* **2024**, *190*, 110039. [CrossRef]
5. Ulmer, M.W.; Thomas, B.W.; Campbell, A.M.; Woyak, N. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transp. Sci.* **2021**, *55*, 75–100. [CrossRef]
6. Han, J.; Liu, H.; Liu, S.; Chen, X.; Tan, N.; Chai, H.; Xiong, H. iETA: A Robust and Scalable Incremental Learning Framework for Time-of-Arrival Estimation. In Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Long Beach, CA, USA, 6–10 August 2023; pp. 4100–4111.
7. Gao, C.; Zhang, F.; Wu, G.; Hu, Q.; Ru, Q.; Hao, J.; He, R.; Sun, Z. A deep learning method for route and time prediction in food delivery service. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Singapore, 14–18 August 2021; pp. 2879–2889.
8. Abdi, A.; Amrit, C. A review of travel and arrival-time prediction methods on road networks: Classification, challenges and opportunities. *PeerJ Comput. Sci.* **2021**, *37*, e689. [CrossRef] [PubMed]
9. Lee, W.H.; Tseng, S.S.; Tsai, S.H. A knowledge based real-time travel time prediction system for urban network. *Expert Syst. Appl.* **2009**, *36*, 4239–4247. [CrossRef]
10. Li, X.; Cong, G.; Sun, A.; Cheng, Y. Learning travel time distributions with deep generative model. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 1017–1027.
11. Asghari, M.; Emrich, T.; Demiryurek, U.; Shahabi, C. Probabilistic estimation of link travel times in dynamic road networks. In Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2015, Bellevue, WA, USA, 3–6 November 2015; pp. 1–10.
12. Wang, Y.; Zheng, Y.; Xue, Y. Travel time estimation of a path using sparse trajectories. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 25–34.
13. Wang, Z.; Fu, K.; Ye, J. Learning to estimate the travel time. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2018; pp. 858–866.
14. Zhu, L.; Yu, W.; Zhou, K.; Wang, X.; Feng, W.; Wang, P.; Chen, N.; Lee, P. Order fulfillment cycle time estimation for on-demand food delivery. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual, 6–10 July 2020; pp. 2571–2580.
15. Li, Y.; Fu, K.; Wang, Z.; Shahabi, C.; Ye, J.; Liu, Y. Multi-task representation learning for travel time estimation. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Francisco, CA, USA, 26–29 August 2018; pp. 1695–1704.
16. Wu, C.H.; Ho, J.M.; Lee, D.T. Travel-time prediction with support vector regression. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 276–281. [CrossRef]
17. Wang, H.; Tang, X.; Kuo, Y.H.; Kifer, D.; Li, Z. A simple baseline for travel time estimation using large-scale trip data. *ACM Trans. Intell. Syst. Technol. TIST* **2019**, *10*, 1–22. [CrossRef]
18. Lin, Y.; Wan, H.; Hu, J.; Guo, S.; Yang, B.; Lin, Y.; Jensen, C.S. Origin-destination travel time oracle for map-based services. *Proc. ACM Manag. Data* **2023**, *1*, 1–27. [CrossRef]
19. Zhou, X.; Wang, J.; Liu, Y.; Wu, X.; Shen, Z.; Leung, C. Inductive graph transformer for delivery time estimation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, New York, NY, USA, 27 February 2023; pp. 679–687.
20. Zhang, L.; Zhou, X.; Zeng, Z.; Cao, Y.; Xu, Y.; Wang, M.; Wu, X.; Liu, Y.; Cui, L.; Shen, Z. Delivery time prediction using large-scale graph structure learning based on quantile regression. In Proceedings of the 2023 IEEE 39th International Conference on Data Engineering (ICDE), Anaheim, CA, USA, 3–7 April 2023; pp. 3403–3416.
21. Hildebrandt, F.D.; Ulmer, M.W. Supervised learning for arrival time estimations in restaurant meal delivery. *Transp. Sci.* **2022**, *56*, 1058–1084. [CrossRef]
22. Gelman, A.; Carlin, J.B.; Stern, H.S.; Dunson, D.B.; Vehtari, A.; Rubin, D.B. *Bayesian Data Analysis*, 3rd ed.; Chapman and Hall/CRC: New York, USA, 2013;
23. van de Schoot, R.; Depaoli, S.; King, R.; Kramer, B.; Märtens, K.; Tadesse, M.G.; Vannucci, M.; Gelman, A.; Veen, D.; Willemsen, J.; et al. Bayesian statistics and modelling. *Nat. Rev. Methods Prim.* **2021**, *1*, 1. [CrossRef]
24. Team, S.D. Stan Modeling Language Users Guide and Reference Manual, 2.34. Available online: https://mc-stan.org/docs/reference-manual/mcmc.html. (accessed on 21 July 2024).
25. Food Delivery Dataset. 2023. Available online: https://www.kaggle.com/datasets/gauravmalik26/food-delivery-dataset (accessed on 13 May 2024).
26. Gibas, J.; Pomykacz, J.; Baranowski, J. Bayesian modelling of travel times on the example of food delivery: Part 1—Spatial data analysis and processing. *Electronics* **2024**, *accepted*.

27. ArviZ. API Reference. 2024. Available online: https://python.arviz.org/en/stable/api/index.html (accessed on 20 May 2024).
28. Vehtari, A.; Gelman, A.; Gabry, J. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC. *Stat. Comput.* **2017**, *27*, 1413–1432. [CrossRef]
29. ArviZ. Arviz.Compare. 2024. Available online: https://python.arviz.org/en/stable/api/generated/arviz.compare.html (accessed on 20 May 2024).
30. Hassani, H.; Silva, E.S. A Kolmogorov-Smirnov Based Test for Comparing the Predictive Accuracy of Two Sets of Forecasts. *Econometrics* **2015**, *3*, 590–609. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.