

Liquidity in the Used Car Market: an Application of Survival Analysis

Marcell P. Granát

Advisor: Péter Vékás

Corvinus University of Budapest, Hungary

Abstract

Buying a car is a major item in the household consumption basket, so participants in this market make their decisions based on carefully considered arguments. After a few years of use, it is very common to sell a vehicle on the used car market, where it is an important question how long it will take to hand over. In this paper, I build a survival model based on a large dataset of used car ads which I collected from the Internet by webscraping, revealing what makes a used car liquid, which is valuable knowledge for all participants in this market. Clearly, one of the most important causes is the price-to-value ratio, which can be derived from the offer price and the price estimated by machine learning models (Ordinary Least Squares, Regression Tree, Random Forest, eXtreme Gradient Boosting, K-Nearest Neighbor, Neural Network, Support Vector Machine). These can provide reliable estimates in a dataset that contains a large number of variables. For computational reasons, I reduced the dimension of the data by Multiple Correspondence Analysis.

In summary, the results show that: (1) eXtreme Gradient Boosting significantly outperforms other prediction methods in estimating the value for money; (2) the value for money is indeed the most important determinant of the time to sell and (3) and some other features of cars can be identified to influence selling prices, such as whether they were American brands, exhibit cars, or had an amplifier output.

Keywords: Used car market, Market liquidity, Survival analysis, Webscraping, Machine learning, Multiple correspondence analysis

1. Introduction

A car is one of the most expensive consumer durables on a household's bill, so buyers make their decisions based on carefully considered principles. Obviously, the most important consideration is the price-to-value ratio. These vehicles are commonly sold on the used car market after use. For this reason, it is important to know how much owners can sell their cars for and how long it will take.

It is simple economics that if someone sells the same car cheaper on the second-hand market, it is likely to find a buyer sooner. However, the value of every vehicle is different, so to determine how quickly it will be sold at a given offer price, the value of the car must first be calculated. \$5,000 can be very expensive for an old car with many miles, but it would be cheap for a lightly used one. Therefore, the correlation between value for money and the duration of the sale is worth investigating. But in

addition to price, many other factors can affect how quickly a vehicle can be sold. As an extreme example, a pink car will certainly attract fewer buyers than a black one.

The market is not stable over time, and the economic environment can have a relevant impact. In 2021, Hungarian Central Statistical Office (2022) documented a 4.5% price increase in vehicle purchases. Price fluctuations may also be due to changes in the exchange rate, as numerous used automobiles are imported into Hungary. Consumers' car purchases may also be affected by the economic cycle. After the crisis, in 2010, a total of 61,324 passenger cars were placed on the market for the first time, compared to 149,137 in 2000 and 295,431 in 2018 (Hungarian Central Statistical Office 2020).

Thus, it is likely that the economic environment will have an impact on the used car market over a period of time, but as the study explains later (Section 2), the data do not allow a methodologically correct examination of this. However, this must be taken into account when analysing the factors mentioned above.

Understanding these mechanisms will provide valuable information to all players in the used car market. Dealers' planning can be helped by knowing how quickly they will be able to dispose of the assets they currently hold, which may have storage costs and capacity constraints. In the event of capacity constraints, it may be worth reducing the offer price for cars whose selling time is significantly affected.

And it helps buyers to consider not only value for money, but also how quickly they will be able to sell the car, as this may become a relevant problem for them years later. This study focuses on two research questions: (Q1) the **relationship between value for money and selling time** and (Q2) the other **vehicle characteristics that can influence selling time**. Figure 1 visualises the theoretical framework outlined above in which the questions are discussed.

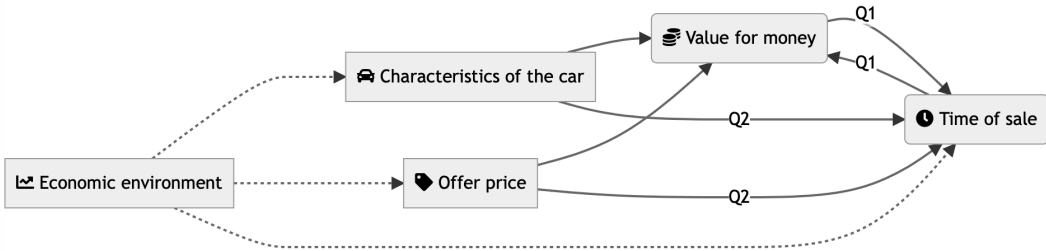


Figure 1. Preliminary hypothesised causal map of the factors under investigation.

I used webscraping to collect data regularly between May 2021 and March 2022, thus containing hundreds of thousands of observations and 250 observed characteristics. The steps to download and clean the data are described in Section 2, while the properties of the market supply relevant to the study are presented in Section 2.5.

All downloaded cars are unique, and I price them using several machine learning methodologies (Ordinary Least Squares, Regression Tree, Random Forest, eXtreme Gradient Boosting, K-Nearest Neighbor, Neural Network, Support Vector Machine). In this step of the paper, the goal is to find the *model with the best predictive power*, interpreting the effects is not a priority. However, prediction with more than 250 predictors is highly demanding from a computational standpoint, thus I reduced the dimension of binary variables using the *Multiple Correspondence Analysis* (MCA) methodology, which is presented in Section 2.3, while the models determining value for money are described in Section 3.1.

Using the calculated relative price, one can determine how quickly a given car is expected to sell in the market at a given price. This requires a statistical model whose output variable is a time

interval. For this purpose, the survival model is applied in Section 3.2. The main steps of the study listed above are summarised in Figure 2.

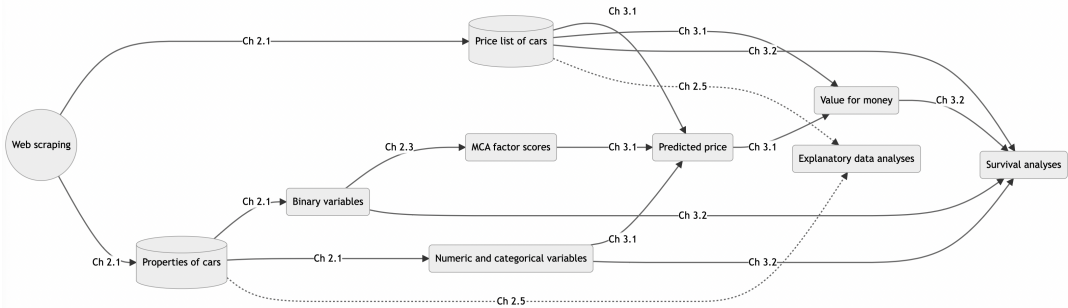


Figure 2. Overview of the study design.

2. Data

2.1 The source of the data

The starting point of the analysis is the data downloaded from the website <https://www.hasznaltauto.hu>, which has the highest traffic in Hungary¹. During the scraping, the IDs and prices of all cars available that day were collected, as well as all available information about the vehicle ads that were first available that day. The simplicity of the latter is intended to reduce the computational need for regular data downloads and the memory size of the data collected, because I assume that all technical details of a given car remain constant over the period it is advertised, while the offer price can change frequently (commonly, the offer price is denominated in euros).

This paper is based on data collected over 10 months. Due to technical difficulties, it has not been possible to download all data daily without omission, but with a personal computer dedicated to this purpose, the process will continue without interruption in the future. Appendix 1 shows the days on which data were downloaded. Data from a total of 613,604 ads were downloaded throughout the period.

Although we cannot be absolutely sure, in my analysis, I make the assumption that if a car is available on the website one day but not the next, it has been sold at the last listed price². The temporary disappearance of a car can also be due to a down payment³, so for the purpose of compiling the data for the analysis, I consider as sold until March 2022 all cars which had not been listed on the website until March 14, when I completed the analysis. Due to the regular downloading, it is known when a particular car ad started and ended, so it is possible to calculate how long it took to sell, which is used in Chapter 3.2.

An important technical detail for the analysis is that three types of data are linked to each car during data collection. The (1) *identifiers and prices*, which are downloaded daily. (2) *General characteristics* of the car, which are available for all cars (sometimes refused to be filled in). These include the number of kilometres driven, the year, engine power, etc. These 23 variables are categorical or numeric (see the full list in Figure 3). And there are *accessories* for cars that are not available for all cars. For example, cruise control, bluetooth, leather seats. These are listed in the description of the cars and have been handled in the following way: all the accessories that occur have been listed, and binary (whether the car has them or not) variables have been created. This

1. Based on the introduction of the website: <https://www.hasznaltauto.hu/araink>

2. My personal experience is that dealers do not really budge on the advertised price, or take the vehicle off the site immediately after purchase.

3. Also personal experience with dealers.

created a total of 235 dummy variables, from which I later created factor scores using MCA analysis to reduce the number of dimensions.

2.2 Handling missing data

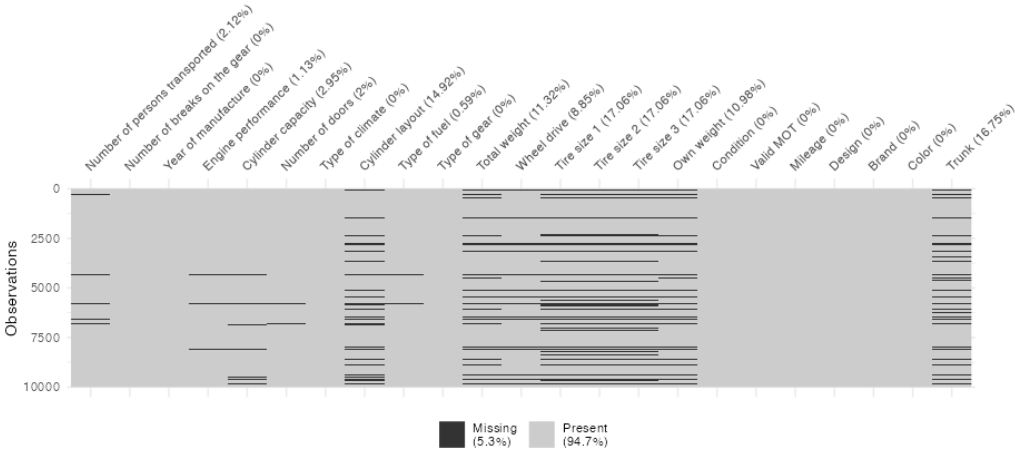


Figure 3. Missingness among the general characteristics of cars.

Values in parentheses after the names of variables indicate the proportions of missing data. I created this figure on a sample of 10,000 cars taken without replacement (the proportions remained stable after regenerating the sample).

In the case of missing general data, I imputed them by the K-Nearest-Neighbour (KNN) procedure. The ratio of missing data points is shown in Figure 3 for each variable. The applied distance function was Gower’s distance, which can be used for mixtures of nominal and numeric data. Gower’s distance is calculated as the average of the distances with respect to all variables. The distance with respect to a nominal variable is simply zero or one depending on whether the two values are identical (Liao et al. 2014, p. 5) and the distance with respect to numeric variables is the Euclidean distance. Once the five nearest neighbours are determined, the mode is used to predict the nominal variables and the mean is used for the numeric data. Applying all possible explanatory variables during imputation is very computationally demanding in the KNN algorithm (see Chapter 3.1), so I used only the most important predictors for the variables to be imputed based on theoretical considerations.⁴ Table 1 details each imputed variable with the predictors that I used to determine the nearest neighbours.

2.3 Dimension reduction

Estimating relative prices for cars with machine learning models is a very computationally demanding task. However, the goal is to maximise the accuracy of predicting the prices of hundreds of thousands of cars, not to interpret the effects. For this reason, it is worth reducing the dimension of variables so that the algorithms do not conflict with memory or time constraints.

4. For example, in the case of the size of the car tyres, I considered the 3 numbers to be categorical, as these are usually characteristics of a car type, strongly related to the price of the car, but not linear, i.e., there are sizes that are significantly more expensive for premium models. This step is motivated by the advice of Schnell Service 2000 kft. (auto tyre manufacturer and distributor).

Table 1. Variables used for imputation.

Imputed variable	Predictors
Cylinder capacity	Brand, year of manufacture, number of persons transported
Cylinder layout	Brand, year of manufacture, cylinder capacity, engine performance
Number of doors	Brand, year of manufacture, own weight, number of persons transported
Number of persons transported	Brand, year of manufacture, own weight
Own weight	Brand, year of manufacture, engine performance, number of persons transported, number of doors
Tire size 1	Brand, year of manufacture, own weight
Tire size 2	Brand, year of manufacture, own weight
Tire size 3	Brand, year of manufacture, own weight
Total weight	Brand, year of manufacture, engine performance, number of persons transported, number of doors
Trunk	Brand, year of manufacture, number of persons transported, own weight, total weight
Type of fuel	Brand, year of manufacture, engine performance, cylinder capacity
Wheel drive	Brand, year of manufacture, engine performance, type of fuel

Principal component analysis (PCA) is a widely used method for generating a new set of artificial variables that capture the most information about the original data. It is an advantageous tool if the multicollinearity among the predictors is high or if we have numerous predictors and the computational capacities are not enough to estimate our model (Sharma and Saroha 2015).

Like PCA, *Multiple Correspondence Analysis* (MCA) performs dimension reduction, but it can be applied to categorical variables rather than numerical ones (Abdi and Valentin 2007). This methodology is better suited to this problem, as the majority of characteristics are represented by dummy variables. Out of the general variables, only the brand was included in the MCA analysis, as it had so many levels that it was worth applying dimension reduction to it. It is also worth including here for theoretical reasons, as the brand could be significantly intertwined with the accessories of cars, so it is expected to share a common pattern with dummy variables⁵.

The first step in the MCA is to create new variables from the variables with a one-hot dummy encoding. This means that for each possible output of each variable, there will be a column in which the value is 1 if a given variable in a given row takes that given value, and 0 otherwise. Each variable is then transformed by dividing the value by the number of 0 values in the column and subtracting one. The algorithm is now exactly the same as for PCA (Venables and Ripley 2013).

The next important question is how many of the factor scores are worth keeping: with fewer variables, the estimation of later models becomes faster, but the retained information decreases. The usual method in this case is to apply the rule of thumb outlined by Kaiser (1960): it is worth keeping the factors where the eigenvalue of the factor exceeds the average of the eigenvalues of the factors. This has resulted in 236 total factors⁶.

2.4 Removing outliers

There may, of course, be intentional or unintentional typing errors in the filling in of the advertising forms. It is most critical for analysis if the offer price is not given correctly.

5. The barrier caused by the oversized dimension occurred during the first modellings. The inclusion of brands in the MCA analysis was also a problem because the R package applied to the random forest modelling (*randomForest*) did not allow a variable with more than 50 levels to be included in the model.

6. While this seems still a lot since we started with 235 categorical variables, several latter models handle these variables with one-hot coding, leading to more than 600 variables, so this is already a significant reduction in computational demand.

There may have been an unintended input when a major price change has affected the car's offer price (IDs and prices have been downloaded daily). We excluded from the analysis if a 400% or greater difference was observed between the highest and lowest offer price for a car. Although it is possible that the correct price was indicated in the ads during the corrections, this affects only 100 cars out of 613,604. In the case of most of the changes, the price of the cars was multiplied by the power of 10, so presumably, the introduction of the unit of measure caused the typo (Hungarian Forints or thousands of Hungarian Forints).

Ads for which the sale price is below HUF 300,000 or above HUF 30,000,000 were also ignored. This was the case for about 10,000 vehicles. In most cases, the advertising price was HUF 1, which was presumably entered to show the car in front of the advertisements in ascending order by price. In a few cases, we found an ad with a price above HUF 30,000,000, which may have been the case of the input error mentioned in the previous paragraph, as it could not be worth that amount based on the car's technical details, but it completely impairs the accuracy of the estimate. The lower and upper limits given above are based on browsing the ads on the website, these are realistic values for a car.

We also removed cars for which the mileage was not reported, as this is likely to be a strong influence on the value for money, and imputing it would lead to very noisy estimates.

After removing all outliers and handling missing data, the modelling was performed with a total of 566,998 complete observations.

2.5 Explanatory data analyses

As the correlation between the time to sell a car and the value of the offer price is the focus of this study, it is worth taking a closer look at the distribution of offer prices. Table 2 provides an overview of the offer prices. As can be seen in the table, prices on the second-hand car market also increased during the observation period, not only in the total vehicle market, which the Central Statistical Office of Hungary officially reports (Hungarian Central Statistical Office 2022). The descriptive statistics also revealed the strongly positively skewed (the tail on the right side of the distribution is longer) and leptokurtic (data have heavy tails and outliers) distribution of the prices. Hence, the logarithm of the offer price is used for modelling.

Table 2. Descriptive statistics of offer prices.

Quarter	Mean	Median	Standard deviation	Skeness	Kurtosis
2021-04/2021-07	3,710,100	1,949,000	4,821,653	2.795249	11.63271
2021-07/2021-10	3,774,025	1,950,000	4,883,133	2.731550	11.19031
2021-10/2022-01	3,904,783	1,990,000	5,057,759	2.626280	10.32879
2022-01/2022-04	4,068,776	2,200,000	4,873,486	2.524668	10.10797
Total	3,871,298	1,999,000	4,931,980	2.659425	10.71338

Only the cars used in the modeling were taken into account in the calculation, so values without outliers are included in the table. Prices are denominated in HUF.

In addition to the offer price, the time to sell is a crucial variable in this analysis. One frequently applied way to visualise the duration distribution is to compute *the Kaplan Meier estimate of the survival function*. The survival function (G) (Equation 1) gives the probability that the event time (T ; selling in the current case) exceeds a given time (t) (Gareth et al. 2013).

$$G(t) = \mathbb{P}(T \geq t) \quad (1)$$

The estimated survival function is visualised in Figure 4. The median survival time was 11 days,

which means that 50% is the probability that you sell your car in 11 days. Not surprisingly⁷, the average survival time is significantly higher, 21.1 days. This reveals that the distribution of selling time is also positive skewed.

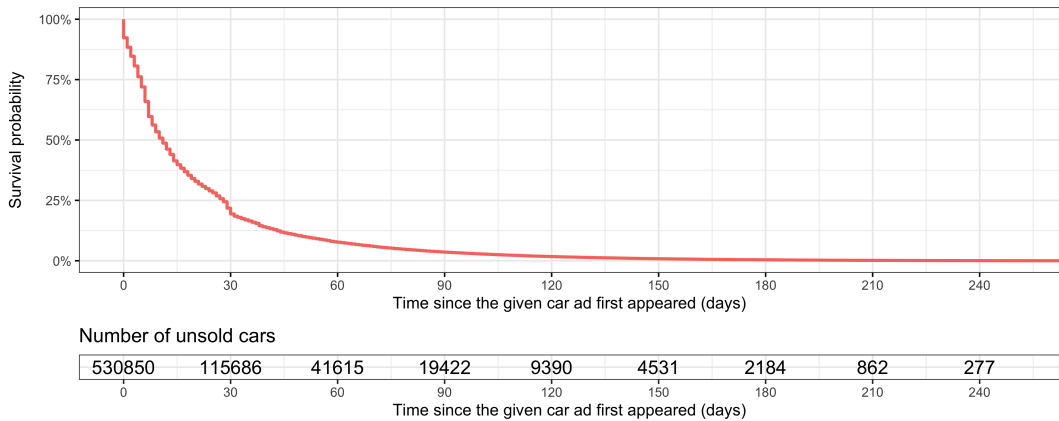


Figure 4. Survival curve showing probability of not having disappeared from the supply.

3. Methodology

To understand the effect of price-to-value ratio on selling time, we must first determine the price-to-value ratio and then use it to create a survival model for sales time. Therefore, the applied methodology is also described in two separate subsections. Section 3.1 presents the methods used to determine the value for money, while Section 3.2 describes the methodology of the survival model.

3.1 Determining the market price

As mentioned in Section 1, the value of each vehicle is different, and we must build a model to estimate it. As the focus of this study is not exploring the causal effects that influence the sale price, but the accuracy of the estimate, it may be a suitable field of application for 'black box' methodologies beyond standard statistical models (ordinary least squares): regression tree, random forest, eXtreme gradient boosting, K-Nearest Neighbor, neural network, support vector machine⁸.

Except for OLS, each of these methodologies generates an estimate with several hyperparameters for which it is necessary to determine the value at which the best estimate is returned. This process is called hyperparameter tuning. For these models, it is important to address the issue of overfitting. Overfitting means that "models have excellent predictivity for the samples on which they were built, but poor predictivity for new samples" (Kuhn and Johnson 2013).

For this reason, I tune the hyperparameters of the model in a **stratified k-fold cross-validation framework**: A sample of 10,000 (this is necessary due to the high computational demands of the models) from the observations sold until July (filtering out the inflation; see the latter paragraph) is split 10 times into training (9,000 observations) and validation sets (1,000 observations). The resulting data sets are balanced in terms of the logarithm of the offer price. Random sampling is conducted within the quartiles of the offer price. I consider the hyperparameter combination to be the best that has the highest average R^2 in the 10 validation samples, since this measures the prediction accuracy in a new sample.

7. Selling duration has an effective lower bound (0), but not an upper bound (length of observation window in our case.)

8. For a detailed description of the models see Gareth et al. (2013).

After tuning (selecting the best hyperparameters for each model) to choose from 7 possible models, I created another sample of 10,000 cars (test set) sold before July 2021 in order to select the best model.

I tested different numbers of parameter combinations for each model during tuning. This happens because the models have different numbers of hyperparameters and runtimes. Table 3 shows how many hyperparameters needed to be tuned, how many parameter combinations were tested, and how long this took⁹ per model. See Appendix 2 for a full list of tuned hyperparameters.

Table 3. Number of hyperparameters, combination of tested parameters, and total runtime of hyperparameter tuning.

Model	# hyperparameters	# parameter combinations	Runtime (seconds)
EXtreme Gradient Boosting Training	7	128	66,280
Ordinary Least Squares	0	1	4
Single-hidden-layer neural network	3	46	37,353
K-Nearest Neighbor	3	32	213,280
Random Forest	2	32	60,923
Linear support vector machines	3	128	184,857
Regression tree	3	64	814

Another issue is that supply prices have generally increased in the market (see Table 2). Therefore, I constructed the model by grouping them into groups of 10,000 items in terms of sales time and built a new model for each subsample to estimate the market price, thus filtering out inflation during the period. As there are also cars that were already available before March 2022 but have not yet been sold, I estimate the price of unsold cars with a model that was made on the last subsample of sold cars.

3.2 Survival analysis

Once we have calculated the price estimated by the 'black box' models, the ratio of that to the offer price can be used to determine the value for money that can be used to estimate *selling time* in addition to the characteristics of the car.

I use the Cox model (also known as Proportional Hazards Regression Model) to estimate sales time, because unlike standard regression models, in this model the outcome variable can only take non-negative values (duration).

The model described by Cox (1972) is based on the fact that each individual has a *hazard rate* that is expressed as a function of explanatory variables. The hazard rate (h) is derived (Equation 2) from the survival function (see Equation 1) and can be interpreted as the intensity process.

$$h(t) = \lim_{\Delta \searrow 0} \frac{P(T < t + \Delta \mid T \geq t)}{\Delta} = \frac{f(t)}{G(t)} \quad (2)$$

The assumption of the model is that the explanatory variables have a *proportional effect on the intensity process* (equation 4), and this effect is constant over time (equation 3).

$$h_i(t) = c_i h_0(t), \quad (3)$$

$$\ln c_i = \sum_{j=1}^p \beta_j x_{ij}, \quad (4)$$

9. CPU: Apple M1 Pro (8 cores in parallel, the combination numbers of the tested hyperparameters are therefore multiples of 8), RAM: 17,2 GB

where i refers to individual cars and j refers to a given explanatory variable.

The latter assumption is crucially related to the validity of the Cox model. The time-invariant effect of predictors can be tested formally on the basis of the scaled Schoenfeld residuals (Grambsch and Therneau 1994).

Just as with standard regression models, fitting a model with 250 explanatory variables can be noisy, as it is hard to find the significant explanatory variables. Thus, I use a regularised Cox model to filter the relevant regressors.

This is done by fitting a GLM with LASSO (least absolute shrinkage and selection operator) regularisation on the data. From a mathematical point of view, lasso regression means adding a $\lambda \sum_{j=1}^p |\beta_j|$ term to the objective function of regression. Intuitively, with this additional term, the value of the objective function (which has to be minimised) is lower if more parameters are equal to zero, but the prediction error does not decrease significantly.

LASSO regression has a hyperparameter λ . Setting its value to zero leads to the unmodified Cox model. On the contrary, a lasso regression with $\lambda \rightarrow \infty$ would lead to an empty model. Finding the optimal λ hyperparameter requires estimating the model with different hyperparameters in a k -fold cross-validation framework. The λ that leads to the model having the lowest number of explanatory variables with adequate accuracy (within 1 standard error of the best model) is used for feature selection (Simon et al. 2011). This process is visualised in Figure 6. In contrast to simple regression models, the goodness of fit of the model is measured by concordance indicators, which represent the proportion of cases where a car estimated to be a better seller was actually sold earlier (concordant pair). One commonly used (and available in the R programming language) measure is the Harell C index, which is simply the proportion of concordant pairs in the total concordant and discordant (when the car estimated with the higher intensity is sold later) pairs (Schmid, Wright, and Ziegler 2016). The possible values of C-index are between 0 and 1, but 0.5 equals the accuracy of a meaningless survival model. The tuning of λ is another computationally demanding process, thus I only involve 10,000 observations for that (later for the Cox model I use all observations).

After variable selection, I fit the unregularized Cox model with the selected terms based on the methodology described above. This is because the LASSO-based model introduces bias by shrinking the coefficients of all variables towards zero, and does not provide information on the significance of the variables (confidence intervals, p-values).

4. Results

4.1 Predicting the value for money

The main results of the models used to estimate the price-to-value ratio are visualised in Figure 5, while the predictive performance on the validation set for each combination of hyperparameters is shown in more detail in Appendix 2.

In Figure 4, it can be seen that the EXtreme Gradient Boosting Training model has the best predictive capacity, and although it has a smaller average R^2 in the test set than in the validation set, it still performs significantly better than all other models (no overlapping confidence intervals). The outstanding performance of the EXtreme Gradient Boosting Training model among several procedures is not entirely surprising, as previous scientific studies have also documented (Nguyen-Sy et al. 2020, Sheridan et al. 2016).

As mentioned, the term value for money denotes the proportion of the offer price and the predicted price. With the prediction of the EXtreme Gradient Boosting Training model, these values can be calculated. Table 4 reports the quartiles and standard deviation of the values. The difference is usually quite small; 50% of the offer prices do not differ from the predicted value by more than 3%. However, the most overpriced car is offered for more than three times its worth based on the

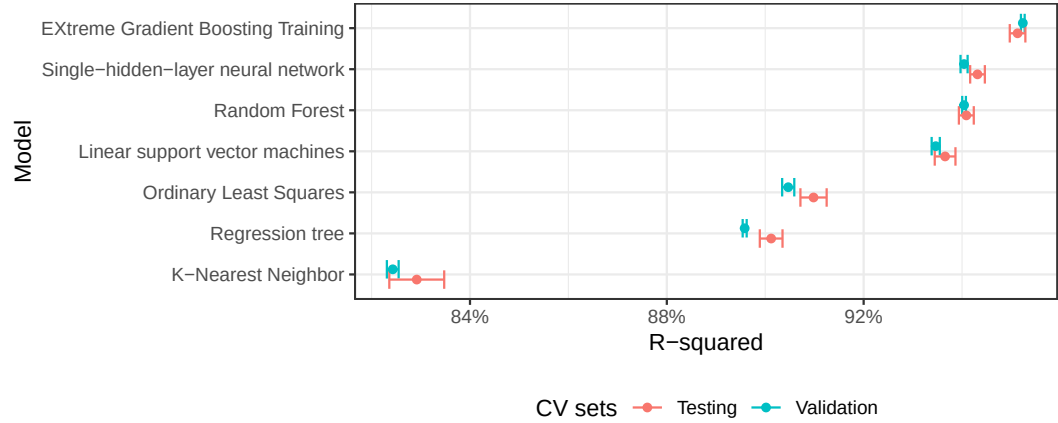


Figure 5. Comparing the estimation accuracy of the investigated models.

estimation. This value is still acceptable¹⁰, but it is a reasonable requirement to check the survival model by omitting these outlier values (robustness check).

Table 4. Descriptive statistics of value for money.

Quarter	Min (%)	Lower quartile (%)	Median (%)	Higher quartile (%)	Max (%)	St. deviation (%p)
2021/04-2021/07	-48.13%	-2.83%	-0.08%	2.66%	61.95%	5.47%
2021/07-2021/10	-51.17%	-2.69%	0.06%	2.87%	208.45%	5.71%
2021/10-2022/01	-50.43%	-2.75%	0.07%	2.96%	198.91%	5.87%
2022/01-2022/04	-47.48%	-2.63%	0.10%	2.89%	124.62%	5.63%
Total	-51.17%	-2.72%	0.06%	2.89%	208.45%	5.74%

Value for money is defined as the proportion of offer price and the predicted price from eXtreme Gradient Boosting. To report in this table I subtracted 1 from the values.

4.2 Results from survival analysis

The result of lasso-based variable selection is that the value of the lambda parameter for a model that still produces acceptable accuracy and contains the least predictors is 0.0177934. Using this to estimate all available observations, we obtain a model with a total of 55 predictors.

¹⁰. Not surprisingly, since cars with outlier offer prices are ignored in the analysis; see Chapter 2.4.

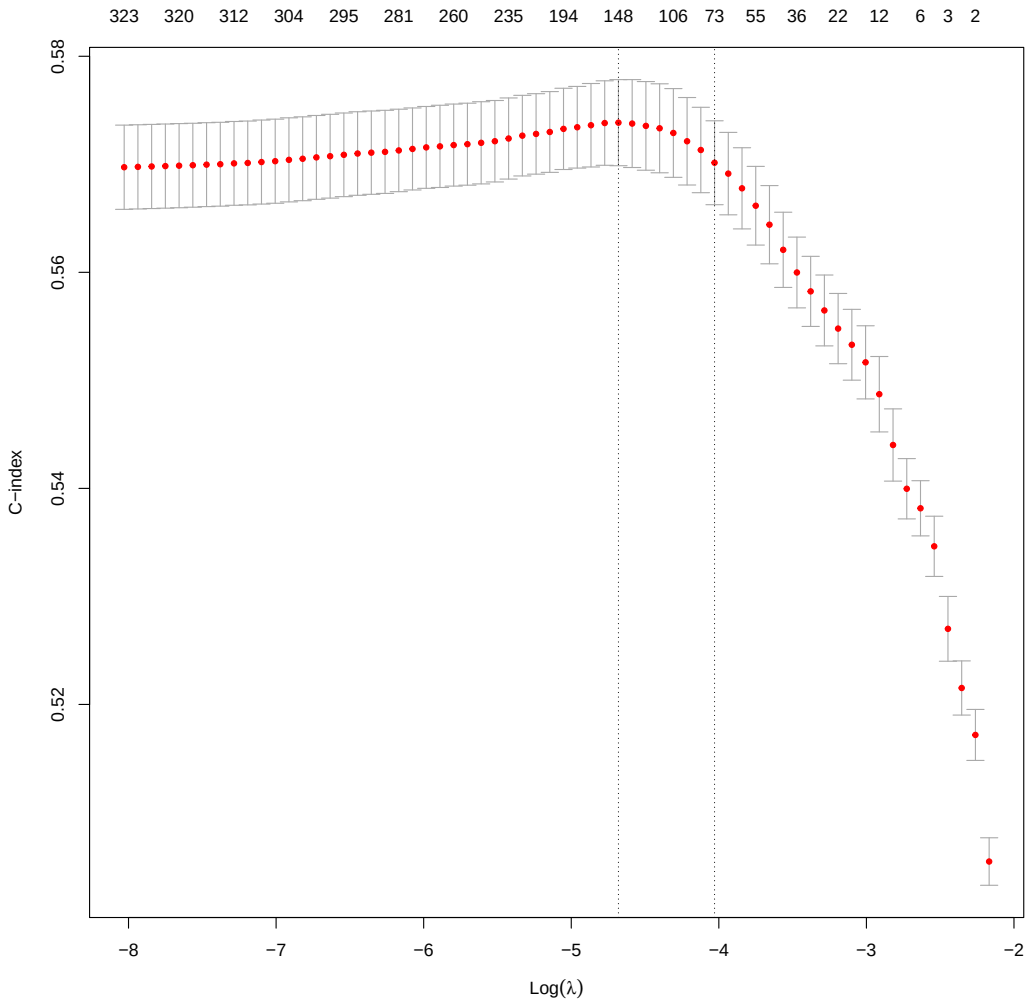


Figure 6. K-fold cross-validation for finding the optimal lambda hyperparameter.

To ensure robustness, I run the feature selection algorithm and estimate the Cox model with and without dropping the lowest and highest 5% of value for money. However, the result is almost identical, so the model is robust in this aspect. Figure 7 visualises the estimated coefficients for both modelling frameworks. The figure shows no considerable difference between the two models, the same terms were selected in the LASSO model, and the confidence intervals of the estimated coefficients overlap in the case of all variables.

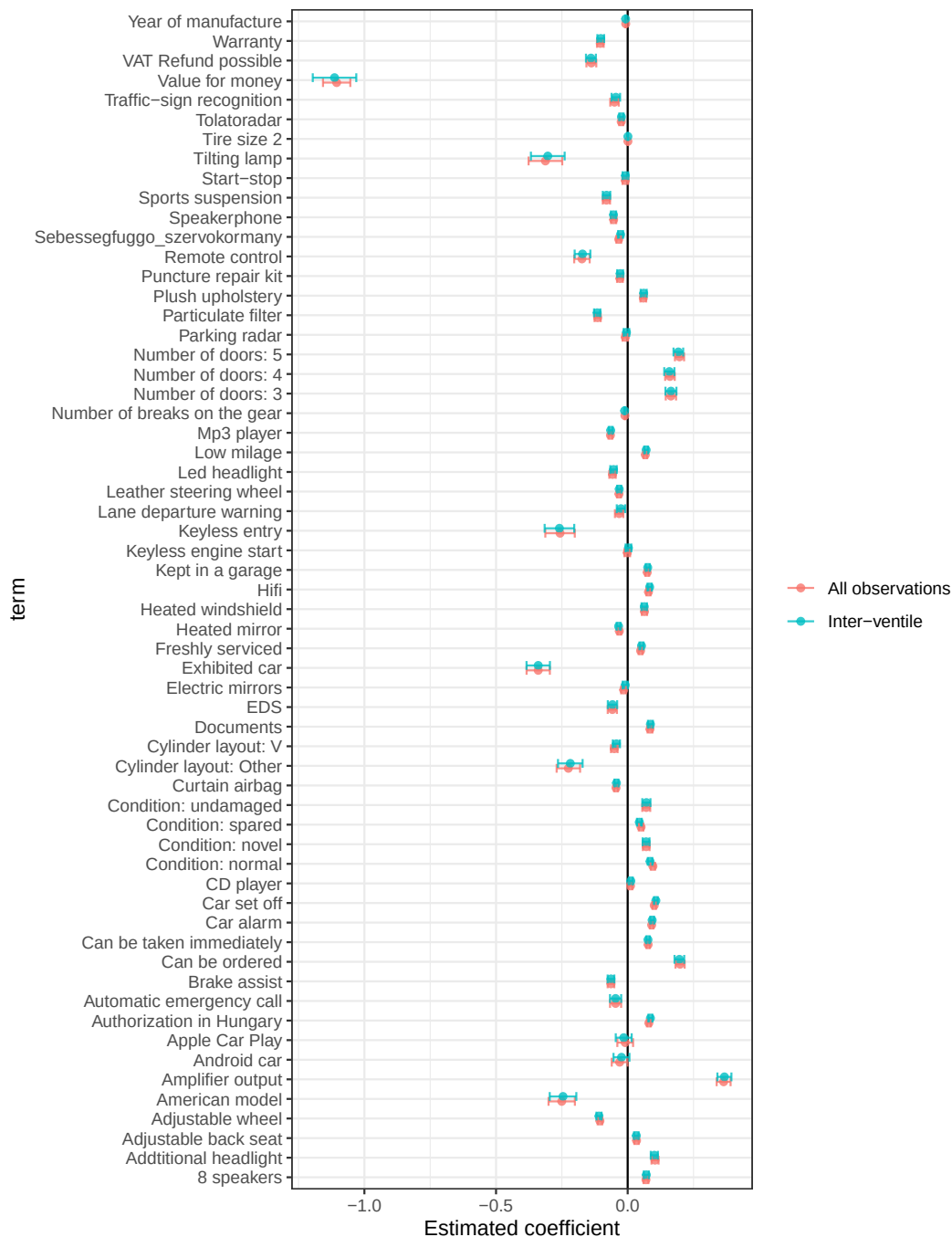
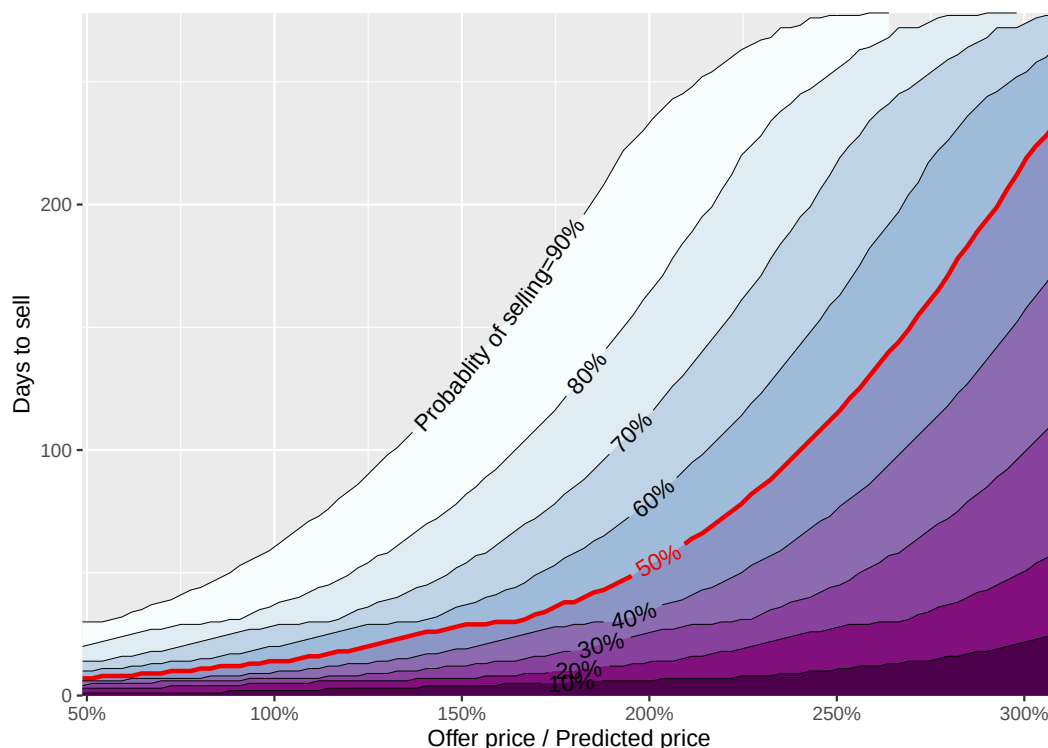


Figure 7. Estimated coefficients of the Cox models.

Since the variables are standardised in the Cox regression, value for money can be interpreted as the most important predictor to influence selling times due to its largest absolute coefficient and its sign is in line with our expectation. The logarithm of the intensity process is regressed with the predictors; thus the negative sign refers to a higher selling time if everything else is unchanged.

To demonstrate the magnitude of the effect, the model response to the change in value for money is visualised in Figure 8. The figure highlights that cars that are significantly overpriced are sold indeed very slow compared to the median selling time calculated for all cars. Although the typical cases (5% relative differences from the estimated price) are not shown well in the figure¹¹.

In the case of the typical cases, the partial effect of price can be visualised as the estimated median selling time for a car with the offer price exactly equals to the predicted price is 14 days, and it decreases to 13 if the car is underpriced with 4%, and increases to 15 days if it is overpriced with 7%. Interpreting these, the value for money itself usually does not imply an outstanding effect on the selling price, but it can if the predicted price is very far from the offered one.



In order to capture the partial effect of value for money assuming that all other variables are unchanged. Numeric predictors are fixed at the median, while categorical and logical variables are fixed at the mode. The monotone increasing red line means, that if the value for money is lower (the offer price to the predicted price is higher), then the median selling time is higher.

Figure 8. Survival model's response to the change of value for money.

If the car is an American model, it significantly increases the selling time. To describe the effect, the median selling time for American cars is 22 days, while it was only 11 in the case of non-American automobiles. The reason is that American cars are produced with different standards and it is harder to find mechanical parts in Hungary. However, this may be not a problem for some customers, thus the value for price is not lower, only the expected selling time is higher, because of the smaller possible customers. Similarly, the median selling time for exhibited cars is 33 days. Over time, it is likely that one can find a buyer who does not have a problem if the car was exhibited, but many would rather not buy it, making it harder to find a buyer for it.

11. One reason for this is that the function I wrote to calculate the partial effect can only visualise discrete values, thus the figure would not add much compared to describing the values. The source code of the functions is available in Appendix 3.

From the results of the estimated coefficients, I would also highlight the amplifier output, which correlates with a significantly shorter expected sales time. However, the causal effect of this variable is questionable, as this characteristic is likely to move along with other characteristics that may be attractive to young people (stickers, lighting, etc.).

Cylinder type 'Other' also leads to increased selling time, but only a few thousand cars have it, and most of them made by Subaru (47%), thus it is a reasonable consideration that this variable is selected because it correlates with an unobserved characteristic of the car. However, answering the exact reason for the other coefficients requires expert knowledge related to the topic of car dealership, so I do not wish to interpret additional coefficients.

5. Conclusion

In summary, the results show that: (1) eXtreme Gradient Boosting significantly outperforms other machine learning methods in estimating the value for money, (2) the value for money is indeed the most important predictor affecting the time to sell and (3) some other features of cars can be identified to influence selling price, such as whether they were American brands, exhibit cars, or had an amplifier output.

6. Discussion

The following statements should be mentioned as the limitation of the analysis. *One can not be sure if the last observed price was the actual selling price, since bargaining is possible.* Ignoring this in the analysis was a necessary assumption since I do not have the ability to control for this from online scraped data.

The assumption of proportional hazard rate functions in equation 3 means that the effect of predictors is time-invariant, which may be tested formally on the basis of the scaled Schoenfeld residuals (Grambsch and Therneau 1994). And in the current case, this *assumption is violated*, which could be remedied by extending the original Cox regression model with time-dependent covariates (Zhang et al. 2018, Fisher, Lloyd D and Lin, Danyu Y 1999). The proportional hazards assumption is routinely accepted in practice nevertheless, and extending the model in this direction is beyond the scope of this paper.

Also because I only observed 10 months, the effect of the economic environment can not be distinguished from the one-off effect of the pandemic (the restrictions were lifted in the summer). For these reasons, I am planning to continue the research in the future with a sufficient number of observations to analyse the effect of the economic environment and re-estimate the model with time-varying covariates (Zhang et al. 2018).

References

- Abdi, Hervé, and Dominique Valentin. 2007. Multiple correspondence analysis. *Encyclopedia of measurement and statistics* 2 (4): 651–657.
- Cox, David R. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 34 (2): 187–202.
- Fisher, Lloyd D and Lin, Danyu Y, journal=Annual Review of Public Health. 1999. Time-dependent covariates in the Cox proportional-hazards regression model. 20 (1): 145–157.
- Gareth, James, Witten Daniela, Hastie Trevor, and Tibshirani Robert. 2013. *An introduction to statistical learning: with applications in R*. Springer.
- Grambsch, Patricia M, and Terry M Therneau. 1994. Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika* 81 (3): 515–526.
- Hungarian Central Statistical Office. 2020. *Statistical Yearbook of Hungary, 2019*.
- . 2022. *Consumer Price Indices by COICOP classification*. Accessed on 03.04.2022. https://www.ksh.hu/stadat_files/ara/en/ara0005.html.
- Kaiser, Henry F. 1960. The application of electronic computers to factor analysis. *Educational and Psychological Measurement* 20 (1): 141–151.
- Kuhn, Max, and Kjell Johnson. 2013. Over-fitting and model tuning. In *Applied predictive modeling*, 61–92. Springer.
- Liao, Serena, Yan Lin, Yan Lin, Dongwan D. Kang, Divay Chandra, Jessica Bon, Naftali Kaminski, Frank C. Sciurba, and George C. Tseng. 2014. Missing value imputation in high-dimensional phenomic data: imputable or not, and how? *BMC Bioinformatics* 15, no. 1 (November): 346–346. <https://doi.org/10.1186/s12859-014-0346-6>.
- Nguyen-Sy, Tuan, Jad Wakim, Quy-Dong To, Minh-Ngoc Vu, The-Duong Nguyen, and Thoi-Trung Nguyen. 2020. Predicting the compressive strength of concrete from its compositions and age using the extreme gradient boosting method. *Construction and Building Materials* 260:119757.
- Schmid, Matthias, Marvin N Wright, and Andreas Ziegler. 2016. On the use of Harrell's C for clinical risk prediction via random survival forests. *Expert Systems with Applications* 63:450–459.
- Sharma, Nitika, and Kriti Saroha. 2015. Study of dimension reduction methodologies in data mining. In *International Conference on Computing, Communication & Automation*, 133–137. Greater Noida, India: IEEE, May. ISBN: 978-1-4799-8890-7, accessed April 7, 2022. <https://doi.org/10.1109/CCAA.2015.7148359>. <http://ieeexplore.ieee.org/document/7148359/>.
- Sheridan, Robert P, Wei Min Wang, Andy Liaw, Junshui Ma, and Eric M Gifford. 2016. Extreme gradient boosting as a method for quantitative structure–activity relationships. *Journal of Chemical Information and Modeling* 56 (12): 2353–2360.
- Simon, Noah, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. 2011. Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software* 39 (5): 1–13. <https://www.jstatsoft.org/v39/i05/>.
- Venables, William N, and Brian D Ripley. 2013. *Modern applied statistics with S-PLUS*. Springer Science & Business Media.
- Zhang, Zhongheng, Jaakko Reinikainen, Kazeem Adedayo Adeleke, Marcel E Pieterse, and Catharina GM Groothuis-Oudshoorn. 2018. Time-varying covariates and coefficients in Cox regression models. *Annals of Translational Medicine* 6 (7).

Appendix 1. Download dates

05/2021 – 02/2022

MAY							JUNE							JULY							AUGUST						
Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
3	4	5	6	7	8	9	1	2	3	4	5	6	7	5	6	7	8	9	10	11	2	3	4	5	6	7	8
10	11	12	13	14	15	16	14	15	16	17	18	19	20	12	13	14	15	16	17	18	9	10	11	12	13	14	15
17	18	19	20	21	22	23	21	22	23	24	25	26	27	19	20	21	22	23	24	25	16	17	18	19	20	21	22
24	25	26	27	28	29	30	28	29	30					26	27	28	29	30	31		23	24	25	26	27	28	29
31																					30	31					

SEPTEMBER							OCTOBER							NOVEMBER							DECEMBER						
Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
		1	2	3	4	5				1	2	3		1	2	3	4	5	6	7			1	2	3	4	5
6	7	8	9	10	11	12	4	5	6	7	8	9	10	8	9	10	11	12	13	14	6	7	8	9	10	11	12
13	14	15	16	17	18	19	11	12	13	14	15	16	17	15	16	17	18	19	20	21	13	14	15	16	17	18	19
20	21	22	23	24	25	26	18	19	20	21	22	23	24	22	23	24	25	26	27	28	20	21	22	23	24	25	26
27	28	29	30				25	26	27	28	29	30	31	29	30						27	28	29	30	31		

JANUARY							FEBRUARY						
Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
3	4	5	6	7	8	9	1	2	3	4	5	6	
10	11	12	13	14	15	16	7	8	9	10	11	12	13
17	18	19	20	21	22	23	14	15	16	17	18	19	20
24	25	26	27	28	29	30	21	22	23	24	25	26	27
31							28						

Starting in mid-August 2021, I was able to run the data download code on a personal server, so the daily download became more regular from then on.

Appendix 2. Hyperparameter tuning

Table 5. Top 10 hyperparameter combination for EXtreme Gradient Boosting Training

trees	min_n	tree_depth	learn_rate	loss_reduction	sample_size	stop_iter	Average R-squared
1746	16	6	0.0258	0	0.7102	14	0.9523
1211	4	10	0.0074	0	0.315	20	0.9517
1779	23	13	0.0528	0.0099	0.6565	13	0.9506
1593	8	9	0.0082	0.0026	0.8579	12	0.9505
869	12	14	0.006	0.0003	0.4159	5	0.9504
1040	37	8	0.0113	0	0.4688	17	0.95
1634	15	4	0.0823	0	0.4631	5	0.9485
653	34	9	0.0185	0.0014	0.1692	14	0.9459
595	17	3	0.05	0.0001	0.8412	16	0.9454
1272	6	15	0.0098	0	0.951	9	0.9448

Table 6. Top 10 hyperparameter combination for Single-hidden-layer neural network

hidden_units	penalty	epochs	Average R-squared
5	0.8302	644	0.9404
5	0.5594	680	0.9403
4	0.3096	851	0.9401
5	0.1741	471	0.936
4	0.0202	675	0.9359
4	0.017	721	0.9353
5	0.379	222	0.9329
5	0	592	0.9321
6	0.001	510	0.9312
5	0	405	0.9309

Table 7. Top 10 hyperparameter combination for K-Nearest Neighbor

neighbors	weight_func	dist_power	Average R-squared
2	rank	0.1465	0.8243
13	gaussian	0.2589	0.8222
9	rank	0.3331	0.8205
11	cos	1.6533	0.8194
13	optimal	1.4462	0.817
11	optimal	1.8215	0.8164
8	triangular	1.865	0.8163
12	cos	0.8301	0.8161
14	rectangular	0.1868	0.8132
9	inv	1.5549	0.8132

Table 8. Top 10 hyperparameter combination for Random Forest

mtry	min_n	Average R-squared
42	6	0.9404
35	3	0.9404
52	8	0.94
75	3	0.9398
47	16	0.9397
65	14	0.9396
28	11	0.9394
72	12	0.9393
58	19	0.9393
41	22	0.9391

Table 9. Top 10 hyperparameter combination for Linear support vector machines

cost	rbf_sigma	margin	Average R-squared
9.583	0.0025	0.1218	0.9347
13.3887	0.0018	0.159	0.9345
3.8105	0.0035	0.1333	0.9316
1.8583	0.0006	0.1903	0.9152
1.6808	0.0088	0.1661	0.9145
3.2048	0.0004	0.0201	0.9129
16.6023	0.0002	0.0103	0.9121
6.1335	0.0094	0.1731	0.9114
8.9552	0.0002	0.0002	0.9103
5.4173	0.0001	0.1546	0.9063

Table 10. Top 10 hyperparameter combination for Regression tree

cost_complexity	tree_depth	min_n	Average R-squared
0	10	39	0.8958
0	8	25	0.8954
0	9	34	0.8954
0	9	28	0.8952
0	8	36	0.8952
0	10	25	0.8949
0	10	21	0.8945
0.0002	9	24	0.8938
0	7	21	0.8925
0	11	27	0.892

Appendix 3. R function for calculating partial dependence from Cox model.

```

partial_effect_data <- function(.data, .var, n = 100, integer = TRUE,
                                cut_quantile = c(.1, .9)) {
  partial_values <- pull(total_surv_df, .var)
  if (is.numeric(partial_values)) {
    partial_quantials <- seq(from = quantile(partial_values,
                                              cut_quantile[1]), to = quantile(partial_values, cut_quantile[2]),
                             length.out = n)
    if (integer) {
      partial_quantials <- unique(round(partial_quantials))
    }
  } else {
    partial_quantials <- unique(partial_values) %>%
      na.omit()
  }

  partial_quantials_df <- enframe(partial_quantials, name = NULL) %>%
    set_names(.var)

  most_freq <- function(x) {
    x %>%
      na.omit() %>%
      enframe() %>%
      count(value, sort = TRUE) %>%
      pull(1) %>%
      first()
  }

  .data %>%
    select(- .var) %>%
    summarise(
      across(where(is.numeric), median, na.rm = TRUE),
      across(where(~ !is.numeric(.)), most_freq)
    ) %>%
    crossing(partial_quantials_df) %>%
    select(names(.data))
}

predict_quantile_prob <- function(.fit, .data, .probs = c(.25, .5, .75),
                                  .keep_predictor = TRUE) {
  probability_df <- predict(.fit, newdata = .data, response = "lp") %>%
    tibble(lp = .) %>%
    mutate(ci = exp(lp), id = row_number()) %>%
    crossing(basehaz(.fit)) %>%
    mutate(probability = exp(- hazard * ci))

  predicted_times_df <- map(.probs, function(x) {
    filter(probability_df, probability <= x) %>%
      distinct(id, .keep_all = TRUE) %>%
      # > First time where the probability of
      # surviving is less than the given prob
      select(id, time) %>%
      set_names("id", str_c("time_", x))
  }) %>%

```

```

    reduce(full_join, by = "id") %>%
    right_join(
      probablity_df %>%
        select(id) %>%
        distinct()
    ) %>%
    arrange(id)

if (.keep_predictor) {

  predicted_times_df <- bind_cols(
    .data %>%
      select_if(~ n_distinct(.) > 1),
    predicted_times_df %>%
      select(- id)
  )
}
predicted_times_df
}

```