How Much Telematics Information Do Insurers Need for Claim Classification?

Francis Duval, Jean-Philippe Boucher, and Mathieu Pigeon
Chaire Co-operators en analyse des risques actuariels, Département de mathématiques, Université du Québec à Montréal, Montréal, Québec, Canada

It has been shown several times in the literature that telematics data collected in motor insurance help to better understand an insured’s driving risk. Insurers who use these data reap several benefits, such as a better estimate of the pure premium, more segmented pricing, and less adverse selection. The flip side of the coin is that collected telematics information is often sensitive and can therefore compromise policyholders’ privacy. Moreover, due to their large volume, this type of data is costly to store and hard to manipulate. These factors, combined with the fact that insurance regulators tend to issue more and more recommendations regarding the collection and use of telematics data, make it important for an insurer to determine the right amount of telematics information to collect. In addition to traditional contract information such as the age and gender of the insured, we have access to a telematics dataset where information is summarized by trip. We first derive several features of interest from these trip summaries before building a claim classification model using both traditional and telematics features. By comparing a few classification algorithms, we find that logistic regression with lasso penalty is the most suitable for our problem. Using this model, we develop a method to determine how much information about policyholders’ driving should be kept by an insurer. Using real data from a North American insurance company, we find that telematics data become redundant after about 3 months or 4000 km of observation, at least from a claim classification perspective.

1. INTRODUCTION

Usage-based insurance (UBI) is a type of motor insurance for which premiums are determined using information about the driving behavior of the insured. This information is usually collected using telematics technology, most often through a device installed in the vehicle or a mobile application. Because of its multiple benefits, this type of insurance is increasingly promoted by insurers. It also seems to be more and more appreciated by consumers. A survey (Willis Towers Watson) conducted by Willis Towers Watson on 1005 insurance consumers in the United States reports that four out of five drivers are in favor of sharing their recent driving information in exchange for a personalized insurance product. Among the benefits, it seems clear that the addition of telematics information into insurance pricing models improves the precision of the pure premium (see, for instance, Lemaire, Park, and Wang 2016; Verbelen, Antonio, and Claeskens 2017; Ayuso, Guillen, and Nielsen 2019; Pérez-Marin and Guillen 2019). UBI also has many positive impacts on society (see, for instance, Bordoff and Noel 2008; Greenberg 2009). Indeed, because it encourages individuals to drive less and more safely, it helps make roads safer, reducing traffic congestion, limiting greenhouse gas emissions, and making insurance more affordable, among other things.

The idea of usage-based insurance was first articulated by William Vickrey, considered as the father of UBI. In Vickrey (1968), he criticized the premium structure then used in motor insurance. He believed that the insurance premium should be modulated according to the use of the vehicle and thus appear as a variable cost to the insured. To correct the premium structure that he considered deficient, in the late 1960s Vickrey proposed a new type of insurance where the premium increases with usage. In particular, he suggested pricing through a tax applied to gasoline or tires: insureds who consumed more gasoline (or tires) would then have a higher premium. However, due to the lack of technology and organizational barriers, it was not until the mid-1990s that the first UBI program appeared in the United States. Nowadays, several major general insurance companies have their own UBI program, and this type of insurance is still growing in popularity. It is now a fact that UBI is collectively beneficial in many ways, and that is why it seems to be the future of motor insurance.

Address correspondence to Francis Duval, Chaire Co-operators en analyse des risques actuariels, Département de mathématiques, Université du Québec à Montréal, Montréal, QC H2X 3Y7, Canada. E-mail: duval.francis.2@courrier.uqam.ca

This article has been corrected with minor changes. These changes do not impact the academic content of the article.
A fairly general section of the UBI literature discusses the feasibility, implementation, costs, and benefits of UBI. Litman (2007) explored the practicability and pros and cons of different types of distance-based insurance, such as Mileage Rate Factor and Pay-at-the-Pump. Greenberg (2009) established that every mile driven insured with UBI rather than conventional insurance brings a significant benefit to the community, which he quantified at $0.016. With this in mind, he proposed benefit maximizing rules and incentives to increase the size of the UBI market. Bordoff and Noel (2008) estimated that the change from traditional insurance to UBI reduced mileage by 8% in average, resulting in yearly savings of $658 per vehicle. Indeed, the fact that the premium increases with usage is a strong incentive to drive less. They stated that most of these savings are attributable to reduced congestion and accidents.

In the recent literature on automobile insurance pricing, among all information you can get from a telematics device, several papers only focus on the distance driven, which is probably the most useful measure for ratemaking. Boucher, Côté, and Guillén (2017) simultaneously analyzed the impact of distance driven and duration on claim frequency using generalized additive models (GAMs) for cross-sectional count data. They found that neither distance nor duration is proportional to frequency but that frequency tends rather to stabilize once a certain distance or duration has been reached. The authors invoked a “learning effect” and a “highway effect” to explain the fact that a policyholder traveling 2x miles is less than twice as dangerous as an insured traveling x miles. Boucher and Turcotte (2020) went further and analyzed the link between distance driven and frequency using models for panel count data, including a GAM for location, scale and shape and a GAM with fixed effects. They refuted the idea of the learning effect to explain the nonlinearity of frequency. Instead, they found that the relationship between frequency and distance driven is approximately linear and that the apparent nonlinearity is due to residual heterogeneity incorrectly captured by GAMs.

Because an insured rarely has more than one claim per year (see, for instance, Boucher, Denuit, and Guillén 2007), assigning a probability of claiming is almost like assigning a premium. Therefore, a significant number of studies related to UBI ratemaking have focused on claim classification. Pesantez-Narvaez, J., M. Guillén, and M. Alcalá (2019) compared the predictive performance of a nonpenalized logistic regression and a boosting algorithm called XGBoost using classical and telematics information, including distance driven, fraction of driving at night, fraction of driving in urban areas, and fraction of driving above speed limit. In Huang and Meng (2019), the authors compared the performance of various classification algorithms while proposing a way to bin continuous telematics variables in order to create a finite number of risk classes and thus increase interpretability. It turns out that this discretization also increases the predictive power of the algorithms. Paefgen, Staake, and Thiesse (2013) also compared multiple classification algorithms on real claim classification data and proposed a novel way to aggregate telematics information into what they called an “aggregate risk factor.”

Moreover, the question of how to transform telematics data into useful information for pricing is a thorny issue to this day. Indeed, it is not yet clear how to use raw telematics information in an optimal way. This falls within the field of feature engineering. Some simply create features manually from the raw data, whereas others contributed using techniques from machine learning, deep learning, and pattern recognition (Weidner, Transchel, and Weidner 2016; Gao and Wüthrich 2018, 2019; Gao, Wüthrich, and Yang 2019).

Despite its many benefits, the growing popularity of UBI means that insurers are accumulating large amounts of sensitive data about their insureds. Dewri et al. (2013) have shown that it is possible to deduce users’ destinations by analyzing their driving data, without even having access to the geographical coordinates of the trips. In recent years, concerns have been arising about the use of telematics technology, particularly with respect to confidentiality and data usage. In Canada and in other countries, insurance regulators make recommendations regarding UBI products. In particular, Financial Services Commission of Ontario (FSCO) and Financial Services Regulatory Authority of Ontario (FSRA) state that telematics data fall under the definition of “personal data”, and must therefore be handled according to the relevant legislation (see Howell, 2013). They also mention that the insurer must inform the consumer in several respects, such as the type of information collected and the use made of it. In spite of these recommendations, the fact remains that large amounts of personal data are stored by insurance companies. From ethical, practical, and economic points of view, it is important for an insurer to keep a minimal amount of personal data on its insureds. Notably, collecting less data reduces storage and manipulation costs in addition to facilitating data leakage prevention.

This raises the question of how much information an insurer should collect on the driving behavior of its insureds, which should be just enough to get a good idea of how the insured drives but not too much for the reasons cited in the preceding paragraph. We explore this avenue through the binary claim classification framework, where the goal is to assign each insured a probability of claiming. In addition to data traditionally used in motor insurance pricing, we have access to telematics data in a format where we have one observation per trip, which allows us to derive interesting and representative features of the

\footnote{Table 2 of this article informs us that 99.5% of the policyholders have one claim or less.}
insured’s driving. We first build a classification model using both classical and telematics features. To this effect, we determine which classification algorithm is the most suited to our problem by comparing among a logistic regression with lasso penalty, a logistic regression with elastic-net penalty, and a random forest, while accounting for interactions between features. It turns out that the lasso model is best suited to our task. Using this classification model, we then develop a method for determining how much telematics information an insurer should collect. For this purpose, we derive several classification datasets using increasing amounts of information about the insured’s driving. We then fit a lasso model on each of these datasets and compare performance: telematics information is considered redundant when it no longer substantially improves the classification score.

In Section 2, we describe the classical and telematics data available to us in conjunction with details of how telematics features are derived. Section 3 follows, where we show how the classification datasets are built using varying amounts of information as well as other preprocessing steps. Then, in Section 4, the mathematical framework of supervised classification is introduced, in addition to explaining the functioning of the three preselected algorithms, namely, the two penalized logistic regressions and the random forest. This is followed by Section 5, in which we first make a choice among the classification models presented in Section 4. We choose the lasso logistic regression for its good performance and simplicity. We also find that adding interactions does not substantially improve the performance of the models. Using our classification model as well as the datasets built in Section 3, we then develop a method to determine the right amount of telematics information to collect based on nonparametric bootstrapping. Using the data provided by the North American insurer, we find that telematics information no longer improves substantially classification after about 3 months, or 4000 km of observation. Finally, we conclude in Section 6.

### 2. DATA

All of the data were provided by a North American insurer and are related to personal auto insurance in the province of Ontario. We have access to a telematics database consisting of 210,854,092 summaries of trips made by 67,355 vehicles between January 2016 and December 2018, for which an extract is shown in Table 1. The recording of a trip, made using an onboard diagnostics device, begins when the vehicle is turned on and stops when the ignition is turned off. Each trip is summarized in four measurements: the datetime of departure and arrival, the distance driven and the maximum speed reached. Trips are also associated with a vehicle via the vehicle identification number, but there is no column to identify the insured person, which makes it impossible to know who the driver is. Therefore, our analysis is based on vehicles rather than policyholders. For 57,671 of these vehicles, which are all observed during one or more insurance contracts, we have access to features traditionally used in motor insurance (gender, age, region, etc.) as well as claiming information. Among these features, which we will call “classical features,” 10 were selected and are described in Table 2. Distributions of the classical numeric features and those of the classical categorical features are shown in Figures A.1 and A.2 in the Appendix, respectively. Only vehicles having at least one full-year contract are kept for the analysis, which means we end up with 29,799 vehicles. In the following section, the telematics dataset will be aggregated by contract (which means the latter will go from one row per trip to one row per contract) and then merged with classical features and claim information (which are already on a contract basis). The

### TABLE 1

| VIN | Trip number | Departure datetime | Arrival datetime | Distance | Maximum speed |
|-----|-------------|---------------------|------------------|----------|---------------|
| A   | 1           | 2016-04-09 15:23:55 | 2016-04-09 15:40:05 | 10.0     | 72            |
| A   | 2           | 2016-04-09 17:49:33 | 2016-04-09 17:57:44 | 4.5      | 68            |
| A   | 3           | 2016-04-09 19:34:55 | 2016-04-09 20:22:54 | 7.0      | 60            |
| B   | 1           | 2016-05-04 06:54:00 | 2016-05-04 07:11:37 | 14.0     | 112           |
| B   | 2           | 2016-04-04 15:20:19 | 2016-04-04 15:34:38 | 13.5     | 124           |
| B   | 3           | 2016-04-04 16:10:33 | 2016-04-04 16:24:54 | 6.0      | 60            |
| C   | 1           | 2016-01-16 15:41:59 | 2016-01-16 15:51:35 | 3.3      | 65            |
| ... | ...         | ...                 | ...              | ...      | ...           |

Dates are displayed in the yyyy-mm-dd format.
resulting dataset will then be given as input to supervised learning algorithms. Because we know that supervised learning algorithms generally learn best on independent observations, we keep only the earliest 1-year contract for each vehicle, which allows us to get rid of the dependency that exists between the different contracts associated with the same vehicle. In their observed year, 99.8% of the vehicles made less than 5000 trips, for an average of 1581 trips per vehicle. The complete distribution is shown in Figure 1.

Based on the trip summaries of Table 1, we wish to derive telematics features that best depict the insured’s driving behavior by aggregating the trips for each vehicle. Indeed, we want these features, or covariates, to have a good predictive power when inputted into a supervised classification algorithm. This falls within the field of feature extraction, which is often a crucial step in machine learning. However, extracting (or creating) features from raw telematics data in an optimal way is not a simple task and is a research avenue in itself. This problem is addressed in several articles such as Wüthrich (2017) and Gao and Wüthrich (2018). Features extracted in these two studies nevertheless require second-by-second data, which we do not have at hand. We are therefore largely inspired by features of the “usage”, “travel habits”, and “driving performance” types derived in Huang

| Classical feature name       | Description                                                   | Type   |
|------------------------------|---------------------------------------------------------------|--------|
| annual_distance              | Annual distance declared by the insured                     | Numeric|
| commute_distance             | Distance to the place of work declared by the insured person | Numeric|
| conv_count_3_yrs_minor       | Number of minor contraventions in the last 3 years            | Numeric|
| gender                       | Gender of the policyholder                                   | Categorical|
| marital_status               | Marital status of the policyholder                            | Categorical|
| pmt_plan                     | Payment plan chosen by the policyholder                      | Categorical|
| veh_age                      | Vehicle age                                                  | Numeric|
| veh_use                      | Use of the vehicle                                           | Categorical|
| years_claim_free             | Number of years since last claim                              | Numeric|
| years_licensed               | Number of years since driving licence                         | Numeric|

Figure 1. Histogram of the Number of Trips for the 29,799 Vehicles in Their Observed Year.
and Meng (2019). From the telematics dataset, we thus extract a total of 14 features, all described in Table 3 and for which the distribution is shown in Figure 2. In Table 3, we also display the average value of the telematics features for two groups of vehicles, namely, the claimants (those who have claimed at least once during their observed year) and the nonclaimants (those who have not claimed during their observed year), and a two-sample \( t \)-test was conducted for each of the features to determine whether the mean differs significantly between the two groups. It turns out that the difference in the mean is significant (at a 95% confidence level) for half of the 14 features, which we have highlight in bold in Table 3. This suggests that these 7 features contain predictively relevant information. Note that claimant vehicles tend to travel more distance and make greater trips, which seems natural. They also tend to have a lower average median speed. This may be due to the fact that claimant insureds have a higher propensity to drive in the city, where average speed is lower and the risk of collision is higher than elsewhere. Claimants also tend to have a higher maximum speed reached in their observed year and drive less in the midday and more in the evening, especially during rush hour. Note that we have at our disposal two “mileage” variables. The first one, annual_distance, is declared by the insured at the very beginning of the contract and corresponds to the number of kilometers that the insured estimates he or she will drive during the following year. The second, avg_daily_distance, is measured by the onboard diagnostics device and is the actual number of kilometers the insured has driven during the year, divided by 365.25, which is the average number of days in a year. One could argue that these two features tell roughly the same thing about an insured and that we should get rid of one of them. However, insureds underestimate their distance traveled in a year by an average 1399 km. Indeed, the average of the avg_daily_distance feature times 365.25 is 16,374 km and the average of annual_distance is 14,975 km. Moreover, the Pearson correlation coefficient between these two is only 0.37, which means there is a considerable discrepancy between annual_distance and avg_daily_distance. We therefore keep both of these features in our analysis, because each one tells a different story about an insured. To determine which features (classical and telematics combined) are significant in predicting the occurrence of a claim, a nonpenalized logistic regression was fitted to the data. The coefficients obtained for the 24 features and their standard deviations are shown in Figure A.3 in the Appendix. It turns out that 10 of the 24 coefficients, among which 7 are related to telematics features, are significant (using a 5% threshold).

### Table 3
Mean Value of the 14 Features Extracted from the Telematics Dataset for Claimants and Nonclaimants

| Feature name | Description | Non-claimants (94.7%) | Claimants (5.3%) | \( p \) Value (\( t \)-test) |
|--------------|-------------|-----------------------|-----------------|-----------------|
| avg_daily_distance | Average daily distance | 44.3 | 50.1 | <.0001 |
| avg_daily_nb_trips | Average daily number of trips | 4.3 | 4.8 | <.0001 |
| med_trip_avg_speed | Median of the average speeds of the trips | 28.8 | 28.0 | <.0001 |
| med_trip_distance | Median of the distances of the trips | 5.2 | 5.2 | .9203 |
| med_trip_max_speed | Median of the maximum speeds of the trips | 69.6 | 70.0 | .2906 |
| max_trip_max_speed | Maximum of the maximum speed of the trips | 138.2 | 141.9 | |
| prop_long_trip | Proportion of long trips (>100km) | 0.0108 | 0.0103 | .4114 |
| frac_expo_night | Fraction of night driving\(^a\) | 0.0262 | 0.0276 | .1630 |
| frac_expo_noon | Fraction of midday driving\(^b\) | 0.210 | 0.198 | <.0001 |
| frac_expo_evening | Fraction of evening driving\(^c\) | 0.0845 | 0.0965 | <.0001 |
| frac_expo_peak_morning | Fraction of morning rush hour driving\(^d\) | 0.0968 | 0.0985 | .4453 |
| frac_expo_peak_evening | Fraction of evening rush hour driving\(^e\) | 0.137 | 0.143 | <.0001 |
| frac_expo_mon_to_thu | Fraction of driving on Monday to Thursday | 0.582 | 0.582 | .6786 |
| frac_expo_fri_sat | Fraction of driving on Friday and Saturday | 0.299 | 0.300 | .5475 |

Two-sample \( t \)-tests were conducted to determine whether the mean differs significantly between the two groups. Boldface indicates a \( p \)-value smaller than 0.05.

\(^{a}0h–6h\)
\(^{b}11h–14h\)
\(^{c}20h–0h\)
\(^{d}7h–9h\) Monday to Friday
\(^{e}17h–20h\) Monday to Friday
Example 1. To illustrate the exact calculation of the 14 telematics features, let us imagine that an insured, during a given contract that we assume lasts 7 days, made only five trips, summarized in Table 4. The calculation of the telematics features related to this insured would then be done according to Table 5.

One must be careful when analyzing telematics data. Indeed, insureds who have chosen to be observed telematically for insurance purposes do not correspond to the general population of insureds, which means our data cannot be considered a simple random sample of the company’s insured population. As far as we are concerned, 10% to 15% of the insureds in our North American insurance company’s portfolio are observed with a telematics device, and these are generally worse and younger drivers. This is because at the time the data were collected, to encourage policyholders to choose the telematics option, insurers were offering a 5% entry discount in addition to a renewal discount ranging from 0% to 25%. At that time, it was also not possible for insurers, at least in the region where the data were collected, to increase the premium based on telematics information: the latter could only be reduced. Because car insurance in Ontario is very expensive, any discount is welcomed by high-premium policyholders such as bad drivers and youth. As a consequence, an unexpectedly large proportion of bad/young drivers end up using the telematics option. However, this selection bias does not affect our analysis because the models and methods we develop apply only to telematically observed insureds. One must only be careful not to draw conclusions from these data and apply them to the general population of insureds.

3. DATA PREPARATION
3.1. Design of the Classification Datasets

With the data we have at hand, we wish to build several classification datasets using a varying amount of telematics information, or trip summaries. Later we perform classification on each of them and compare performance, which will allow us to determine how much telematics information is needed to obtain a proper classifier. For this purpose, we compute telematics features of Table 3 in several versions, which we do in two different ways. The first way to proceed, called the “time leap”
The (TL) method, consists of first calculating features using only 1 month of trip summaries for each vehicle and then adding 1 month’s worth of data to each subsequent version. Because vehicles are observed over 1 year, telematics features of Table 3 are derived in 12 different versions. In general, the $k$th version is calculated using the first $k$ months of telematics information related to a given vehicle, for $k = 1, \ldots, 12$. The second way to proceed, called the “distance leap” (DL) method, is quite similar to the time leap way, but uses 1000-km leaps instead of 1-month leaps to jump from version to version. For the sake of uniformity, we also derive telematics features in 12 different versions using this second method. In general, the $k$th version of a telematics feature for vehicle $i$ is calculated using the first $1000 \times k$ km of telematics information related to this vehicle, for $k = 1, \ldots, 12$. If it turns out that the vehicle drove less than 10000 km in its observed year, we simply use all available telematics information from this vehicle. Note that 37% of the vehicles have accumulated less than 12,000 km of driving during their observed year, which means that they end up with some identical versions of the telematics features for the DL method. For instance, a vehicle that has accumulated 5500 km of driving has its DL versions 6 to 12 of the telematics features calculated with the same amount of telematics information, namely, with 5500 km of trip summaries. Each version of the telematics features can be represented by a $29,799 \times 14$ matrix, where each row corresponds to a vehicle and each column to a feature. The matrix containing the $k$th version of the telematics features derived according to the TL method is denoted as $x_{TL}^k$, and the one derived according to the DL method is denoted as $x_{DL}^k$. Let us also denote by $x^c$ the $29,799 \times 10$ matrix containing the classical features of Table 2 and by $y$ the response vector of length 29,799, which indicates whether or not each vehicle had a claim in its observed year. By using TL versions of the telematics features, we then build 12 classification datasets, denoted

### Table 4

| Trip number | Departure time | Arrival time | Weekday | Distance | Average speed | Maximum speed |
|-------------|----------------|--------------|---------|----------|---------------|---------------|
| 1           | 18:20          | 18:28        | Monday  | 8        | 60            | 73            |
| 2           | 17:40          | 17:54        | Monday  | 9        | 39            | 70            |
| 3           | 09:35          | 09:48        | Tuesday | 17       | 78            | 102           |
| 4           | 07:30          | 07:37        | Thursday| 9        | 77            | 92            |
| 5           | 12:20          | 13:35        | Saturday| 109      | 88            | 120           |

Note that weekday and average speed can be easily derived from the telematics dataset.
catch-all category. The categorical features, rare categories (i.e., those associated with 5% of the observations or less) are lumped together in a

3.2. Preprocessing

Training and testing datasets need to be preprocessed every time they are being fed to the models, for either training or scoring purposes. The reasons for this are manifold. First, some of our features are categorical (gender, marital_status, pmr_plan and veh_use), and many supervised learning algorithms cannot handle this type of information, which means we need a way to encode them numerically. For this, we choose an embedding method called “target encoding,” which consists in replacing the value of each category, that is originally a string of characters, by a real number based on the response (or target) column. In the special case of mean target encoding, the value of each observation is replaced by the mean of the response variable for the category to which that observation belongs. For instance, imagine we have the feature “gender” with categories “woman” and “man” and that the mean of the response variable is 0.09 for women and 0.11 for men. Women would then be encoded with the value 0.09 and men would be assigned the value 0.11. What we use is similar to mean encoding, except that the encoded values are derived using a generalized linear model rather than using the mean. Suppose we are in the context of supervised binary classification and we want to encode the feature x, which is categorical with k categories. Generalized linear model target encoding consists of first fitting a logistic regression without intercept using only x to explain the binary response variable y, which yields coefficient estimates \( \hat{\beta}_1, \ldots, \hat{\beta}_k \). Then, each of the k categories is encoded with its corresponding coefficient. Hence, category j is encoded with the value \( \hat{\beta}_j \), for \( j = 1, \ldots, k \). This means that the k-category categorical feature x becomes a numerical feature with k unique values. To perform target encoding, the step_lencode_glm function of the embed package in the R programming language (R Core Team, 2020) is used. Note that prior target encoding the categorical features, rare categories (i.e., those associated with 5% of the observations or less) are lumped together in a catch-all category.
Secondly, the \texttt{commute\_distance} feature, which is numerical, is missing for 22.3\% of the observations. Because many classification algorithms cannot handle missing values, we need a way to impute them. Virtually any prediction algorithm could be used to perform imputation, but some are more suitable than others. We chose an algorithm based on bagged decision trees (see Breiman 1996) because they are known to be good at imputing missing data, partly because they generally have good accuracy and because they do not extrapolate beyond the bounds of the training data. Bagged decision trees are also preferable to random forest because they require fitting fewer decision trees to have a stable model. The idea is to first consider the feature to be imputed, namely, \texttt{commute\_distance}, as a response variable. Then, a bagged decision tree model is trained on all observations for which \texttt{commute\_distance} is not missing, using all features except the latter as predictors. This fitted model is then scored on all observations with a missing \texttt{commute\_distance} value, and the prediction is used as a replacement value. To implement bag imputation, we use the \texttt{step\_bagimpute} function of the \texttt{recipe} package in \texttt{R}.

Thirdly, supervised learning algorithms generally learn best when the data are preprocessed in a certain way. For instance, they generally benefit from normalized and rather symmetrical feature distributions. To normalize features, we use the \texttt{z-score} normalization, which is a quite popular choice. Consider the numerical feature vector $x = (x_1, \ldots, x_n)$. The \texttt{z-score}-normalized version of this vector is

$$x' = \left(\frac{x_1 - \bar{x}}{s}, \ldots, \frac{x_n - \bar{x}}{s}\right),$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2}$ are the empirical mean and standard deviation of vector $x$, respectively. In order to obtain more symmetric feature distributions, we use the Yeo-Johnson transformation (see Yeo and Johnson 2000), which is similar to the Box-Cox transformation except that it allows for negative input values. Yeo-Johnson transformation also has the effect of making the data follow a more Gaussian-like distribution. The Yeo-Johnson transformation $\psi$ applied on a real value $x$ is defined as follows:

$$\psi(x, \lambda) = \begin{cases} 
\frac{(x + 1)^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0, x \geq 0 \\
\ln(x + 1) & \text{if } \lambda = 0, x \geq 0 \\
-(\frac{(-x + 1)^{2-\lambda} - 1}{2-\lambda}) & \text{if } \lambda \neq 2, x < 0 \\
-\ln(-x + 1) & \text{if } \lambda = 2, x < 0,
\end{cases}$$

where $\lambda$ is a parameter that is optimized by maximum likelihood so that the resulting empirical distribution resembles a Gaussian distribution as closely as possible. Note that the preprocessing steps are performed in a specific order that depends on the feature type. Figure 3 illustrates the data preprocessing “recipe.”

4. BINARY CLASSIFICATION ALGORITHMS

4.1. Binary Classification Framework and Classification Algorithm Preselection

Let us be in the context of binary supervised classification, in which we have at our disposal $n$ labeled examples $\{(x_i, y_i)\}_{i=1}^{n}$, where $x_i = (x_{i1}, \ldots, x_{ip})$ and $y_i \in \{0, 1\}$ are respectively the $p$-dimensional vector containing the features and the label (or response variable) for observation $i$. The goal is to estimate $E[Y|x]$, the conditional mean of $Y$ given the features, which can also be seen as the conditional probability $P(Y = 1|x)$. Many algorithms have been developed to estimate this probability, including logistic regression, random forest, artificial neural networks, support vector machines, etc. For the analysis, we retain two supervised learning algorithms; that is, penalized logistic regression and random forest. The reason for this choice is that the latter often give excellent classification results while requiring little preprocessing of the data. In fact, these two could be qualified as off-the-shelf algorithms because they perform implicitly feature selection and do not require much data preprocessing, unlike other algorithms such as neural networks. They are also quite easy to tune because they do not have too many hyperparameters. Note that some algorithms that require more care before inputting the data into the model can probably lead to better classification performance, but our goal is not really in that way. For the penalized logistic regression, we consider two specifications, namely, the lasso penalty (also called $\ell_1$-penalty) and the elastic-net penalty.
4.2. Logistic Regression

In logistic regression, the goal is to approximate the conditional probability of having a positive case \((Y = 1)\) by applying the sigmoid function over a linear transformation of the features. The model can therefore be expressed as

\[
p_i := P(Y_i = 1 | x_i) = \sigma \left( \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} \right),
\]

where \(\sigma\), the sigmoid function, ensures that the output is a real number between 0 and 1. The model is parameterized by the unknown parameter vector \(\beta = (\beta_1, ..., \beta_p)\) and the intercept \(\beta_0\). These parameters are often estimated by maximum likelihood, which leads to asymptotically unbiased estimates. Maximizing likelihood is equivalent to minimizing cross-entropy loss, given by

\[
L(\beta_0, \beta) = -\frac{1}{n} \sum_{i=1}^{n} y_i \ln(p_i) + (1-y_i) \ln(1-p_i).
\]

There is no closed formula for maximum likelihood parameter estimates in the logistic regression framework, but a variety of numerical optimization methods can be used. Most of the time, the method of iteratively reweighted least squares is used.

4.3. Lasso Logistic Regression

The maximum likelihood estimator is the asymptotically unbiased estimator with the smallest variance. However, it is rarely the best for prediction accuracy. Indeed, although it has a low bias, it has a rather large variance. In 1996, Tibshirani (1996) proposed a new method called least absolute shrinkage and selection operator (lasso) for estimating parameters in linear regression that reduces the variance of the parameters at the cost of increased bias. In practice, this decrease in variance more than offsets the increase in bias, thus improving predictive performance. Although this method was originally used for models using the least squares estimator, it generalizes quite naturally to generalized linear models. In the case of logistic regression, a penalty proportional to the sum of the absolute values of the parameters is added to the cross-entropy loss of Equation (2). In lasso logistic regression, the optimization problem thus becomes

\[
\min_{\beta_0, \beta} \left\{ L(\beta_0, \beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right\},
\]

where \(\lambda \geq 0\) is the penalty hyperparameter. In the special case where \(\lambda = 0\), the penalty term disappears and we recover the conventional nonpenalized logistic regression. This penalty hyperparameter is not directly optimized by the algorithm and must therefore be chosen by the user; for instance, with cross-validation and grid search. Equation (3) is called the Lagrangian formulation of the lasso logistic regression optimization problem. It can be useful to rewrite this in the constrained form,

\[
\min_{\beta_0, \beta} \left\{ L(\beta_0, \beta) \right\} \quad \text{s.t.} \quad \sum_{j=1}^{p} |\beta_j| \leq s.
\]

Note that there is a one-to-one correspondence between \(\lambda\) and \(s\). This formulation allows the realization that the model gives itself a “budget” of parameters. Indeed, the sum of the absolute values of the coefficients, which is the \(\ell_1\) norm of the parameter vector \(\beta\), must always be less than or equal to the constant \(s\) set by the user. This has the effect of shrinking and even zeroing some of the logistic regression coefficients. The lasso logistic regression fits into the more general framework where the constraint on the parameter vector is given by

\[
\sum_{j=1}^{p} |\beta_j|^q \leq s,
\]

where \(q \geq 0\) is a fixed real number. In particular, setting \(q = 1\) retrieves the lasso constraint, whereas \(q = 0\) and \(q = 2\) correspond respectively to best subset selection and ridge logistic regressions. Best subset selection and ridge have both their pros
and cons. Ridge regression only shrinks coefficients; it never sets them to zero, which makes interpretation more difficult. In general, one prefers to have a sparse model: according to Occam’s razor, a simple explanation of a phenomenon is to be preferred to a more complex one. Best subset selection leads to sparse models, but it involves resolving a nonconvex and combinatorial optimization problem and becomes infeasible above about 50 features. Lasso regression attempts to retain good features from both subset selection and ridge; it leads to sparse models while being a convex optimization problem. In fact, we can show that \( q = 1 \) is the smallest value that leads to a convex problem, and this partly explains why lasso regression is so popular. The loss function to be minimized in Equation (3) is not differentiable due to the absolute values in the penalty term, but it is convex, and a wide range of methods from convex optimization theory have been developed to compute the solution, including coordinate descent, subgradient, and proximal gradient-based methods. In this article, lasso logistic regression is fit using the \texttt{glmnet} library of the \texttt{R} programming language. This library uses a proximal Newton algorithm, which consists of making a quadratic approximation of the log-likelihood part of the loss function and then applying a weighted coordinate descent iteratively. For more details on lasso logistic regression, we refer to Friedman, Hastie, and Tibshirani (2010), Hastie, Tibshirani, and Wainwright (2015), and Hastie, Qian, and Tay (2021).

### 4.4. Elastic-Net Logistic Regression

Even though lasso regression often performs very well on tabular data, it has a few drawbacks. Among other things, when there is a group of features that are highly correlated with each other, the lasso tends to select only one feature in the group and does not care which one is selected. This can be a problem for us, because some of the telematics features we have created (or even classical features) may be highly correlated with each other (e.g., \texttt{avg_daily_distance} and \texttt{avg_daily_nb_trips}). Zou and Hastie (2005) addressed this problem by proposing a new regularization and variable selection method called “elastic-net.” The elastic-net regression combines the penalties of ridge and lasso regressions and thereby retains the best of both worlds. Ridge regression is known to share parameters among highly correlated features, which often improves performance, whereas lasso yields sparse models and thus performs feature selection, which is desirable. Elastic-net regression thus yields sparse models while improving the treatment of highly correlated features. More precisely, elastic-net regression includes a penalty term in its loss function that is a linear combination of the \( \ell_1 \) (lasso) and \( \ell_2 \) (ridge) penalties. In the particular case of binary classification, elastic-net coefficients are therefore obtained by solving

\[
\min_{\beta_0, \beta} \left\{ L(\beta_0, \beta) + \lambda \left[ \frac{(1-z)}{2} \| \beta \|_2^2 + z \| \beta \|_1 \right]\right\}, \tag{6}
\]

where a new “mixing” hyperparameter \( 0 \leq z \leq 1 \) appears. Ridge and lasso regressions are in fact special cases of elastic-net regression, when \( z = 0 \) and \( z = 1 \), respectively. If \( z \) and \( \lambda \) are known, Criterion (6) is convex and can be solved by a variety of algorithms such as coordinate descent.

### 4.5. Random Forest

The random forest classifier was first formalized by Leo Breiman in 2001 and enjoys of great popularity for its many strengths, including a usually high accuracy on structured data. It consists of building several decision trees on slightly modified versions of the original dataset. The final prediction is then obtained by aggregating all of the trees, often by taking the mean on the individual predictions. The trees built are usually very deep and therefore have little bias but have a large variance. Aggregating them allows drastically decreasing the variance and obtaining a much more flexible prediction function, increasing the predictive power compared to a single tree. The main advantage of random forest over logistic regression is that it can approximate a wider range of functions because it is a nonparametric algorithm, making fewer assumptions about the form of the underlying data generating function. Another benefit is that it automatically takes into account the interactions between features due to the tree structure of its components. Like lasso and elastic-net logistic regressions, random forest has an embedded feature selection mechanism. On the other hand, random forest is harder to interpret, so for equal performance, logistic regression is preferred. Note that because logistic regression assumes a linear relationship between the features and the log-odds of a positive case, random forest usually outperforms it when this relationship is rather nonlinear. In order to train a random forest model, one first generates bootstrap samples of the training dataset on which decision trees will later be built. This is done by drawing observations with replacement, and usually as many observations as there are in the original training set are drawn. Then, for each of these bootstrap samples, \( 1 \leq p^* \leq p \) features are randomly picked, \( p^* \) being a previously chosen hyperparameter. This last step allows the decision trees to be built on different subsets of features, which has the effect of decorrelating the predictions, thus improving performance. Finally, a CART-like classification tree (see Breiman et al.
is built upon each of these datasets, each one yielding an estimated probability of having a positive case for every point of the feature space. The criterion we use to build the trees is the impurity of the nodes, measured by the Gini index. Every time the feature space is split in two, we thus choose the splitting point that decreases the Gini index the most. Other criteria are also possible. For a given point, a final prediction is obtained by averaging the individual predictions of all trees. More details about the general procedure are given in Algorithm 1. For more information about random forest, we refer to Breiman (2001), Hastie, Tibshirani, and Wainwright (2015), and Hastie, Qian, and Tay (2021).

Algorithm 1: Random forest binary classifier

**Inputs:**
- Training dataset $\mathcal{T} = \{(x_i, y_i)\}_{i=1}^n$ containing $p$ features
- Number of features to pick $1 \leq p^* \leq p$
- Number of trees (or bootstrap samples) $B$ for $b = 1, \ldots, B$

**for** $b = 1, \ldots, B$ **do**
1. Generate a bootstrap sample $\mathcal{T}^*$ by drawing with replacement $n$ observations from $\mathcal{T}$.
2. Pick at random $p^*$ of the $p$ features.
3. Build a CART classification tree on $\mathcal{T}^*$ using only the $p^*$ features previously picked, yielding the prediction function $\hat{T}_b(x)$.

**end**

**Output:** Random forest classifier $\hat{f}_{RF}(x) = \frac{1}{B} \sum_{b=1}^B \hat{T}_b(x)$

5. **ANALYZES**

5.1. **Claim Classification Model**

In order to choose among the three classification models presented in Section 4 (lasso logistic regression, elastic-net logistic regression, and random forest), we make them compete on the dataset whose telematics features are computed with all available trip summaries of the observed year for each vehicle, namely, $\mathcal{D}_{12}^{TL}$.

5.1.1. **Hyperparameter Tuning**

First of all, the three models need to be tuned because they all involve hyperparameters that are not directly optimized by their respective algorithm. The general idea behind tuning is to test several combinations of hyperparameters and evaluate the out-of-sample performance for each of them, which is often done using a validation set or cross-validation on the training set. One then usually chooses the combination of hyperparameters yielding the best performance. We choose to use five fold cross-validation with the area under the receiver operating characteristic curve (AUC) as the metric for evaluating classification performance. This metric is often used in binary classification, notably because it does not depend on the threshold used for classification and because it works well on unbalanced datasets (our dataset is highly unbalanced because there are many more nonclaimant vehicles than claimant ones). Different methods have been developed to choose which combinations of hyperparameters to try out, including grid search, random search, Bayesian optimization, gradient-based optimization, and evolutionary optimization. For penalized logistic regression, it is generally not necessary to use a sophisticated tuning algorithm, and one usually proceeds with a simple grid search, which we do. For the random forest, we choose a more refined method, namely, a Bayesian optimization method, which has been shown to yield better results than grid search and random search (see, for instance, Snoek, Larochelle, and Adams 2012).

The lasso requires the tuning of only one hyperparameter, which is the penalty parameter $\lambda$ of Equation (3). We first create a grid of 100 penalty values ranging from $10^{-10}$ to 1 uniformly distributed on a logarithmic scale, namely, the set

$$G_\lambda = \{10^{-10+i/90}\}_{i=1}^{100}.$$  

Then, five fold cross-validation AUC is assessed on the training set $\mathcal{T}_{12}^{TL}$ using each of the values in $G_\lambda$ as a candidate. It turns out that the best value for $\lambda$ is 0.000231 (which is the value in $G_\lambda$ associated with $i = 64$), yielding an AUC of 0.6373. For the elastic-net model, the mixing parameter $\alpha$ must also be tuned in addition to the penalty parameter (see Eq. [6]). With a grid search, one usually uses a coarse uniform grid of values for $\alpha$. We thus choose to use five values uniformly distributed between 0 and 1 inclusively, namely, the grid $G_\alpha = \{0, 0.25, 0.5, 0.75, 1\}$. For $\lambda$, we use the same grid as for lasso; that is, $G_\lambda$. The
performance of the \( |\mathcal{G}_2| \times |\mathcal{G}_2| = 100 \times 5 = 500 \) possible combinations of hyperparameters is thereafter evaluated and it turns out that \( \lambda = 0.00298 \) (which is the value in \( \mathcal{G}_2 \) associated with \( i = 75 \)) with \( z = 0 \) is the best choice, with an AUC value of 0.6377, slightly better than lasso.

Regarding the random forest, two hyperparameters are tuned, namely, the number of features drawn every time a tree is built \( p^* \) (see Algorithm 1) and the minimum number of observations required to make a further split in any leaf \( n^* \). Note that for simplicity, the latter does not appear in Algorithm 1. The total number of trees \( B \) must also be chosen, but it is not, strictly speaking, a hyperparameter. It only needs to be large enough for the performance to stabilize. We choose \( B = 1000 \), which is plenty for the number of observations we have. Note that \( B \) cannot be too large because a random forest can never overfit due to too many trees. However, increasing the number of trees obviously increases the computation time. Bayesian optimization is used to find the best possible pair \((p^*, n^*)\). Basically, it consists of treating the unknown function that maps hyperparameter values to the loss function evaluated on a test set (or with cross-validation) as a random function. An a priori distribution, which captures beliefs about the behavior of this function, is defined. Then, as combinations of hyperparameters are tested and evaluated, the a priori distribution is updated, yielding the a posteriori distribution. The latter is thereafter used to find the next combination of hyperparameters to try out. So unlike grid and random search, Bayesian optimization leverages past evaluations to find the most promising candidates faster. To implement this procedure, we use the \texttt{tune_bayes} function of the \texttt{tune} package in \texttt{R}, which uses a Gaussian process to model the probability distribution over the function. One can think of the Gaussian process as a generalization of the normal distribution concept to functions. It turns out that \((p^*, n^*) = (1, 39)\) is the best pair that has been tested, yielding an AUC value of 0.6004. This means only one feature is picked every time a tree is built and that the growth of a tree stops when all of its leaves have less than 39 observations.

5.1.2. Interactions

A limitation of logistic regression is that it does not naturally take into account interactions between features, unlike random forest. Fortunately, this can be overcome by manually calculating interactions. According to the interaction hierarchy principle (see Kuhn and Johnson 2019), lower-level interactions are more likely than higher-level ones to explain variation in the response variable. For instance, level 2 interactions are more likely to be predictive than level 3 interactions, which are more likely to be predictive than level 4 interactions, and so on. Therefore, in order to keep computation time reasonable, we only consider level 2 (or pairwise) interactions. We also limit ourselves to calculating the interactions between the 10 classical features of Table 2 as well as telematics features whose mean value is significantly different between claimant and nonclaimant vehicles (i.e., the seven bolded features in Table 3). These seven features are presumed to have good predictive power, and according to the principle of heredity (see Kuhn and Johnson 2019), they have a higher probability than other features of creating interactions that also have good predictive power. This entails calculating \( \binom{17}{2} = 136 \) pairwise interactions. Next, lasso and elastic-net regressions are tuned on the training dataset \( \mathcal{T}_1^{12} \) expanded with the 136 interactions as new columns. For this purpose, the same grid search procedure described in Subsection 5.1.1 is used.

5.1.3. Out-of-Sample Performance Comparison

Optimal hyperparameter(s) found as well as the five fold cross-validation AUC value for each of the five tuned models are shown in Table 6. The interaction-free elastic-net model has the best cross-validation score, with an AUC of 0.6377. However, it does not outperform the lasso model, which has an AUC of 0.6373, enough to justify the extra complexity. Indeed, an elastic-net model takes more time to tune, because it has an additional hyperparameter, and also takes longer to fit. Note also that with or without interactions, the two logistic regressions perform similarly considering the variability of the AUC. Because one always prefer the simplest model for equal performance, we reject the two models including interactions. Finally, the random forest is the worst model, with an AUC of only 0.6004, which is considerably lower than the penalized logistic regressions. We therefore reject the latter. Note that the Bayesian optimization algorithm has found that the optimal value for the hyperparameter \( p^* \) is 1, reinforcing the belief that interactions between features do not carry useful information for classification. Indeed, the fact that \( p^* = 1 \) means that the decision trees that make up the random forest are built with only one feature at a time, thus eliminating the possibility of including interactions.

Ideally, in order to properly estimate performance in supervised learning, a model should be evaluated on samples that have not yet been used to build or fine-tune it. We therefore use the testing dataset \( \mathcal{T}_1^{12} \) to assess the performance of the five tuned models. The models were first trained on the full training dataset \( \mathcal{T}_1^{12} \) before being scored on \( \mathcal{T}_1^{12} \), which allowed us to compute an AUC value for each of them, shown in Table 6. The AUC values on the testing set are slightly lower than those obtained by cross-validation, which is normal. In fact, the relatively close values indicate that we did not leak too much information into the models. The lasso model with interactions has the best testing set AUC value (0.6214), but we believe it is not
Training datasets should collect from its policyholders. The method we propose consists of first tuning and training a lasso model on each of the 5.2. Classification Performance Assessment on the 13 Classification Datasets

that the difference in the risk of an accident between men and women is mainly due to the difference in driving intensity, which is captured by two of our telematics features, namely, avg_daily_distance and max_trip_max_speed. As pointed out by Ayuso, Guillen, and Pérez-Marín (2016), it seems that the difference in the risk of an accident between men and women is mainly due to the difference in driving intensity, which is captured by two of our telematics features, namely, avg_daily_nb_trips and avg_daily_distance.

5.1.4. Feature Importance

Once the models are trained, in addition to the performance, it is interesting to look at which features contributed the most to classifying observations. For the three models compared, it is possible to calculate an importance score for each feature. For lasso and elastic-net logistic regressions, because the models are trained with normalized versions of the features, the absolute value of the estimated parameter may be used for this purpose. For instance, if the estimated parameter associated with \( \text{avg}_-\text{daily_distance} \) is \( \hat{\beta}_1 \), its importance score is \( |\hat{\beta}_1| \). For the random forest, there are many ways to compute feature importance. We chose to use the mean decrease of the Gini index. This method consists of assessing for each feature how much it has contributed to decrease the impurity of the tree nodes, measured with the Gini index. Once the importance score is obtained for all features, we can order them from the most to the least important, which we did in Figure 4 for each of the three models. Looking at this figure, we can notice that the lasso and elastic-net models have a high degree of agreement because they consider the same 12 most important features and the links intersect very little. This is not so surprising because these two logistic regression models work in a similar way. On the other hand, the random forest has a lower degree of agreement with the latter two because the links intersect a lot. This is probably because a random forest makes no assumptions about the nature of the relationship between the features and the response variable, whereas logistic regression assumes that this relationship is logistic-linear. The random forest therefore probably leverages better the features with rather nonlinear relationships with the response variable, in contrast to logistic regression. However, all three models agree that \( \text{avg}_\text{daily_nb_trips} \), \( \text{avg}_\text{daily_distance} \), \( \text{max_trip_max_speed} \), \( \text{frac}_\text{expo_evening} \), and \( \text{frac}_\text{expo_mon_to_thu} \) are important features for prediction, all ranked in the top 10 most important features. One last important thing to note is that telematics features are considered more important by the models than their classical counterparts. Indeed, telematics features have an average ranking of 9.45, whereas for classical features this value is 16.77. This makes us believe that telematics data tell a better story about an insured’s risk than traditional ratemaking information, which is consistent with the literature. For instance, Verbelen, Antonio, and Claeskens (2017) showed using GAMs that telematics features have better predictive power than their classical counterparts. Indeed, their GAM model, which only uses telematics, outperforms their GAM using only classical factors, at least according to some criteria. In traditional pricing—that is, in pricing without driving data—gender is an important factor in determining an insured’s premium. Note that here gender only ranks 10th, and this is probably due to the fact that gender is a proxy for other factors related to driving habits. As pointed out by Ayuso, Guillen, and Pérez-Marín (2016), it seems that the difference in the risk of an accident between men and women is mainly due to the difference in driving intensity, which is captured by two of our telematics features, namely, \( \text{avg}_\text{daily_nb_trips} \) and \( \text{avg}_\text{daily_distance} \).

5.2. Classification Performance Assessment on the 13 Classification Datasets

Remember that our main objective is to develop a method to estimate the amount of telematics information that an insurer should collect from its policyholders. The method we propose consists of first tuning and training a lasso model on each of the training datasets \( T_m^0, ..., T_m^{12}, m \in \{\text{TL, DL}\} \) derived in Section 3. The lasso model tuned and trained on dataset \( T_m^k \) is denoted by \( M_k^m \), for \( k = 0, ..., 12 \) and \( m \in \{\text{TL, DL}\} \). We then assess the performance of these fitted models on their corresponding

| Models                          | \( \lambda \)   | \( \alpha \) | \( p^* \) | \( n^* \) | AUC (five-fold cross-validation) | AUC (testing set) |
|--------------------------------|-----------------|-------------|----------|----------|---------------------------------|------------------|
| Lasso                          | \( 2.31 \times 10^{-4} \) | —           | —        | —        | 0.6372^{(0.0052)}              | 0.6189           |
| Elastic-net                    | \( 2.98 \times 10^{-3} \) | 0           | —        | —        | 0.6377^{(0.0049)}              | 0.6176           |
| Random forest                  | —               | —           | 1        | 39       | 0.6004^{(0.0064)}              | 0.5889           |
| Lasso (with interactions)      | \( 1.18 \times 10^{-3} \) | —           | —        | 1        | 0.6350^{(0.0050)}              | 0.6214           |
| Elastic-net (with interactions)| \( 1.52 \times 10^{-2} \) | 0           | —        | —        | 0.6359^{(0.0046)}              | 0.6198           |

Numbers in superscript indicate standard deviations.
testing dataset. For instance, the fitted model $M_m^k$ is assessed on the testing dataset $V_m^k$, for $k = 0, \ldots, 12$ and $m \in \{\text{TL, DL}\}$. The performance is evaluated using the AUC, and in order to obtain a distribution of the latter, a nonparametric bootstrap strategy is used. Nonparametric bootstrap is a method used to estimate the distribution of any statistic and consists of generating new samples (or datasets) called “bootstrap samples.” A bootstrap sample is simply obtained by drawing with replacement as many observations as there are in the original sample. An empirical distribution is then obtained by calculating the desired statistic on each bootstrap sample. Therefore, we generate $b = 500$ bootstrap samples for each of the 13 testing datasets $V_m^0, \ldots, V_m^{12}$, $m \in \{\text{TL, DL}\}$. Then, in order to obtain an AUC distribution for model $M_m^k$, we score the latter on each of the 500 bootstrap samples related to $V_m^k$, denoted as $(\delta j)^{500}$, and we derive the 500 corresponding AUC values, which form the empirical distribution. Once the empirical distribution of the AUC has been obtained for each of the models $M_m^0, \ldots, M_m^{12}$, $m \in \{\text{TL, DL}\}$, it is thereafter possible to inspect them and determine at what point telematics information becomes redundant or, in other words, at what point the addition of telematics information in the lasso model no longer meaningfully improved the classification performance. The AUC distributions are shown in Figure 5 for the 13 lasso models and for both approaches, namely, the TL approach (top panel) and the DL approach (bottom panel). The first thing that can be noticed when looking at the box plots is that the addition of telematics features into the model significantly improves its performance, which is in line with the literature. Indeed, the first box plot from the left in top and bottom panels, which corresponds to the classical model, is lower than all the other ones. Looking at the top panel, we realize that with the addition of as little as 1 month’s worth of trip summaries, we can drastically improve classification performance. Similarly, adding just 1000 km of telematics data into the classical model also improves performance substantially (bottom panel). After 1 month or 1000 km, although the marginal improvement is less substantial, the AUC slightly increases but stabilizes fairly quickly, after about 3 months or 4000 km. After this point, telematics information collected seems redundant and no longer improves classification. This suggests that telematics features calculated with 3 months of trip summaries have about the same predictive power as those calculated with one complete year of data.
FIGURE 5. Distribution of the Classification Performance (AUC) Obtained with Nonparametric Bootstrap for the 13 Models and for Both Approaches. *Note:* Black stars show the AUC value obtained on the original testing set.
6. CONCLUSIONS

In this article, we first created several relevant telematics features from trip summaries in order to incorporate telematics information into supervised classification models. Then, using these features in conjunction with features traditionally used in automobile insurance, and by adequately preprocessing the data, we compared the performance of three popular classification algorithms, namely, a lasso logistic regression, an elastic-net logistic regression and a random forest. We found that random forest, which often gives good results in classification tasks, performs the worst, whereas the two logistic regressions are on equal footing. However, we chose the lasso as our classification model because of its greater simplicity. We also considered interactions between features and found that they contain little or no predictive power because their addition into the models does not improve out-of-sample classification performance. Then, based on the lasso model and thus remaining within the framework of supervised classification, we developed a novel method for determining when information on the insured’s driving becomes redundant. A great strength of our method is that it requires little computational time and does not require second-by-second trip data, which are large and therefore time-consuming to process and difficult to manipulate. Also, it can be used by any insurance company that has access to a dataset similar to the one used in the analysis, namely, a telematics dataset where each observation corresponds to a trip. Using real data from a North American insurance company, we found using nonparametric bootstrapping that after about 3 months or 4000 km of observation, telematics no longer helps achieve a better classification performance, at least if measured with the AUC. This means it is probably not worth it for this insurer to observe its policyholders during long periods of time. Rather, it is better off observing them during a predetermined short period. Indeed, people generally do not enjoy being tracked, and telematics data are both costly and risky to store and manipulate. In addition, collecting less telematics information can help to meet the recommendations of insurance regulators and facilitate the prevention of data leakage.

As reported in Bolderdijk et al. (2011), policyholders tend to adopt better driving habits in the early months of telematics observation when a financial incentive is given. Therefore, it is probably not a good idea to observe an insured’s driving habits for 3 months (or 4000 km), determine a discount (or surcharge) and then apply that same discount (or surcharge), ad vitam aeternam. This could indeed result in pure premiums that are too low, because the observed behavior would be biased due to a lack of financial incentive to drive well after the observation period. A simple solution to this issue would be to monitor policyholders at random times during their contract to eventually collect 3 months (or 4000 km) of data. Guillen, Nielsen, and Pérez-Marín (2021) addressed this issue by making pricing dynamic. Basically, they proposed a pricing scheme in which the premium is updated weekly, which drives the insured to adopt good driving habits on an ongoing basis. They combined a baseline premium related to traditional risk factors with extra charges related to driving behaviors deemed unsafe captured by what they call “near-miss events,” such as hard braking and smartphone use when driving. Moreover, they provided an alternative way to solve the problem of minimizing data storage because their method only requires keeping 1 week of telematics data. However, unlike our article, they did not use information related to driving habits such as the proportion of night driving, average speed, proportion of long trips, etc., which have a demonstrated link to claim experience. Their method also requires collecting good near-miss data, which is not trivial. In this analysis, only collision coverage claims were considered; that is, the target column given as input to the classification models is the indicator of a collision claim, at-fault or not. One could repeat the analysis by considering at-fault and non-at-fault collision claims separately and see whether either type needs more or less telematics history to have a good estimate of the claiming probability. The analysis could also be performed again using collision-free claims, including theft, vandalism, and fire, and see whether we come to similar conclusions. Lastly, we could generalize the approach for count data. Therefore, instead of having the indicator of a claim as the response variable, we would have the number of claims, moving us into a counting regression context.

ACKNOWLEDGMENTS

The authors gratefully acknowledge The Co-operators for both financial support and for providing the data used in this paper through the Co-operators Chair in Actuarial Risk Analysis. The authors also thank the editors and the three anonymous reviewers for providing feedback and helping them.

FUNDING

The authors thank The Co-operators and the Natural Sciences and Engineering Research Council of Canada for funding.
REFERENCES
Ayuso, M., M. Guillén, and J. P. Nielsen. 2019. Improving automobile insurance ratemaking using telematics: Incorporating mileage and driver behaviour data. Transportation 46 (3):735–52. doi:10.1007/s11116-018-9890-7
Ayuso, M., M. Guillén, and A. M. Pérez-Marín. 2016. Telematics and gender discrimination: Some usage-based evidence on whether men’s risk of accidents differs from women’s. Risks 4 (2):1–10. doi:10.3390/risks4020010
Baekke, P., and L. Bocca. 2017. The value of vehicle telematics data in insurance risk selection processes. Decision Support Systems 98:69–79. doi:10.1016/j.dss.2017.04.009
Bolderdijk, J. W., J. Knockaert, E. Steg, and E. T. Verhoef. 2011. Effects of pay-as-you-drive vehicle insurance on young drivers. Transportation Research Part D: Transport and Environment 16 (2):139–48. doi:10.1016/j.trd.2010.12.002
Bordoff, J. E., and P. J. Noel. 2008. The impact of pay-as-you-drive auto insurance in California. Washington D.C.: Brookings Institution.
Boucher, J.-P., S. Côté, and M. Guillén. 2017. Exposure as duration and distance in telematics motor insurance using generalized additive models. Risks 5 (4). doi:10.3390/risks5040054
Boucher, J.-P., M. Denuit, and M. Guillén. 2007. Risk classification for claim counts: A comparative analysis of various zero-inflated Poisson and hurdle models. North American Actuarial Journal 11 (4):110–31. doi:10.1080/10920277.2007.10597847
Boucher, J.-P., and R. Turcotte. 2020. A longitudinal analysis of the impact of distance driven on the probability of car accidents. Risks 8 (3). doi:10.3390/risks8030091
Breiman, L. 1996. Bagging predictors. Machine Learning 24 (2):123–40. doi:10.1007/BF00058655
Breiman, L. 2001. Random forests. Machine Learning 45 (1):5–32. doi:10.1023/A:1010933404324
Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen. 1984. Classification and regression trees. Boca Raton, FL: CRC Press.
Dewri, R., P. Annadata, W. Eltarjaman, and R. Thurimella. 2013. Inferring trip destinations from driving habits data. In Proceedings of the 12th ACM workshop on privacy in the electronic society, 267–72.
Dewri, R., T. Hastie, and R. Tibshirani. 2010. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software 33 (1). doi:10.18637/jss.v033.i01
Gao, G., and M. V. Wüthrich. 2018. Feature extraction from telematics car driving heatmaps. European Actuarial Journal 8 (2):383–406. doi:10.1007/s13385-018-0181-7
Gao, G., and M. V. Wüthrich. 2019. Convolutional neural network classification of telematics car driving data. Risks 7 (1). doi:10.3390/risks7010006
Gao, G., M. V. Wüthrich, and H. Yang. 2019. Driving risk evaluation based on telematics data. Insurance: Mathematics and Economics 88, 108–119.
Greenberg, A. 2009. Designing pay-per-mile auto insurance regulatory incentives. Transportation Research Part D: Transport and Environment 14 (6):437–45. doi:10.1016/j.trd.2009.01.003
Guillen, M., J. P. Nielsen, and A. M. Pérez-Marín. 2021. Near-miss telematics in motor insurance. Journal of Risk and Insurance 88 (3), 569–589. doi:10.1111/jori.12340
Hastie, T., J. Qian, and K. Tay. 2011. An introduction to glmnet. CRAN R Repository.
Hastie, T., R. Tibshirani, and M. Wainwright. 2015. Statistical learning with sparsity: The lasso and generalizations. Boca Raton, FL: CRC Press.
Howard, P. 2013. Usage-based automobile insurance pricing in Ontario (2013). https://www.fsrao.ca/industry/auto-insurance-sector/guidance/archived/usage-based-automobile-insurance-pricing-ontario-2013
Huang, Y., and S. Meng. 2019. Automobile insurance classification ratemaking based on telematics driving data. Decision Support Systems 127:113156. doi:10.1016/j.dss.2019.113156
Kahn, M., and K. Johnson. 2019. Feature engineering and selection: A practical approach for predictive models. Boca Raton, FL: CRC Press.
Lemaire, J., S. C. Park, and K. Wang. 2016. The use of annual mileage as a rating variable. ASTIN Bulletin 46 (1):39–69. doi:10.1017/asb.2015.25
Litman, T. 2007. Distance-based vehicle insurance feasibility, costs and benefits. Victoria, British Columbia, Canada: Victoria Transport Policy Institute.
Paelgen, J., T. Staake, and F. Thiesse. 2013. Evaluation and aggregation of pay-as-you-drive insurance rate factors: A classification analysis approach. Decision Support Systems 56:192–201. doi:10.1016/j.dss.2013.06.001
Pérez-Marín, A. M., and M. Guillén. 2019. Semi-autonomous vehicles: Usage-based data evidences of what could be expected from eliminating speed limit violations. Accident Analysis & Prevention 123:99–106. doi:10.1016/j.aap.2018.11.005
Pesantez-Narvaez, J., M. Guillén, and M. Alcañiz. 2019. Predicting motor insurance claims using telematics data—xgboost versus logistic regression. Risks 7 (2). doi:10.3390/risks7020070
R Core Team. 2020. R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/
Snook, J., H. Larcheolle, and R. P. Adams. 2012. Practical Bayesian optimization of machine learning algorithms. arXiv preprint arXiv:1206.2944.
Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological) 58 (1):267–88.
VerbeLEN, K., T. Antonio, and G. Claeskens. 2017. Unraveling the predictive power of telematics data in car insurance pricing. Journal of the Royal Statistical Society: Series C (Applied Statistics) 67 (5):1275–1304.
Vickrey, W. 1968. Automobile accidents, tort law, externalities, and insurance: An economist’s critique. Law and Contemporary Problems 33 (3):464–87. doi:10.2307/1199038
Weidner, W., F. W. Transchel, and R. Weidner. 2016. Classification of scale-sensitive telematic observables for risk individual pricing. European Actuarial Journal 6 (1):3–24. doi:10.1007/s13385-016-0127-x
Willis Towers Watson (2017). Infographic: How ready are consumers for connected cars and usage-based car insurance?. https://www.wtwco.com/en-US/Insights/2017/05/infographic-how-ready-are-consumers-for-connected-cars-and-usage-based-car-insurance
Wüthrich, M. V. 2017. Covariation selection from telematics data car driving data. European Actuarial Journal 7 (1):89–108. doi:10.1007/s13385-017-0149-2
Ye, J.-K., and R. A. Johnson. 2000. A new family of power transformations to improve normality or symmetry. Biometrika 87 (4):954–959. doi:10.1093/bio- met/87.4.954
Zou, H., and T. Hastie. 2005. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67 (2):301–20. doi:10.1111/j.1467-988X.2005.00503.x
Discussions on this article can be submitted until July 1, 2023. The authors reserve the right to reply to any discussion. Please see the Instructions for Authors found online at http://www.tandfonline.com/uaaj for submission instructions.

APPENDIX

FIGURE A1. Distributions over the 29,799 Vehicles of the Six Classical Numeric Features Selected for the Analysis.
FIGURE A2. Distributions over the 29,799 Vehicles of the Four Classical Categorical Features Selected for the Analysis.
FIGURE A3. Nonpenalized Logistic Regression Coefficients and Their Standard Deviations for Classical and Telematics Features Obtained with the 29,799 Vehicles.