Evaluating the Performance of Explainable Machine Learning Models in Traffic Accidents Prediction in California

Camilo Parra
Departamento de Informática
Universidad Técnica
Federico Santa María
Valparaíso, Chile
caparra@alumnos.inf.utfsm.cl

Carlos Ponce
Departamento de Informática
Universidad Técnica
Federico Santa María
Valparaíso, Chile
cponce@alumnos.inf.utfsm.cl

Rodrigo Salas F.
Escuela de Ingeniería C. Biomédica
Universidad de Valparaíso
Valparaíso, Chile
rodrigo.salas@uv.cl

***Abstract*—Reducing and preventing road traffic accidents is a major public health problem and a priority for many nations. In this paper, we seek to explore the performance of explainable machine learning models applied to the prediction of road traffic crashes using a dataset containing nearly three million records of this type of events and the conditions under which they occurred. To achieve this, the dataset `US Accidents – A Countrywide Traffic Accident Dataset` is used. First we will clean, standardize and reduce the data, then we will transform the time and location values using a geo-hashing library developed by Uber, later, we will increase our dataset to obtain events classified as 'not an accident' using web scraping techniques in the data sources of the original authors of the dataset. Then, we will evaluate the performance of different implementations of Random Forest and decision trees, we obtained a performance superior to 70% for the `F1` score of these models. Finally, we conclude that weather conditions are strongly related to the car accident.**

***Index Terms*—Traffic Accidents, Geolocation data, Random Forest, Decision Trees, Gradient Boosted Trees**

1. Introduction

Traffic accidents are a major public health problem and helping to predict their causes or occurrences has been a topic of interest in the field of machine learning. According to data from the World Health Organization, 1.35 million people lose their lives each year due to traffic accidents worldwide [1], during 2018 there were 6.7 million accidents in the United States [2] and in Chile, during 2019 there were 89 983 traffic accidents in which 1 617 people died [3]. It is clear that we need new tools to support the prevention of these accidents.

Previously, studies and models have been developed to predict the risk of a traffic accident occurrence. However, they have the difficulty of not having enough relevant and critical information for the prediction. In this work, we seek to explore the possibility of using a method based on the `Random Forest` classification technique to predict the probability of a traffic accident using a dataset with almost three million records of events in the United States, which contains both detailed descriptions of the environment in which the accident occurred as well as location and timing data of the event. In this research, the question that we are trying to answer is: Can we enhance the estimation of the risk of a traffic accident occurrence by extending the data set with meteorological information and with non-accident patterns?

Regarding the prediction of risk for a traffic accident occurrence, Chen [4], used 1.6 million GPS records on people's mobility along with a set of $300,000$ accident records in the city of Tokyo, Japan, to predict the probability of the occurrence of an accident in a grid with cells with $500 \times 500m^2$ as size, in one-hour time intervals through a model that extracts latent characteristics of human movement and a logistic regression based technique to make predictions. Yuan [5] used a heterogeneous dataset containing road characteristics, temperature and weather data along with demographic data to predict the probability of an accident on each road segment in the state of Iowa, USA, based on an analysis on the eigenvalues of their dataset. Lin [6] defined a model that uses decision trees to differentiate records representing the situation before an accident from records where no accidents occurred, considering information such as visibility, traffic volume, weather and speed. Wenqi [7] designed a convolutional neural network model that uses multiple state matrices to describe vehicle flow, weather, visibility, and other factors to predict an accident. Finally, Moosavi [8], uses a set of 3 million traffic accidents and information on weather, date and traffic events in the United States to implement a multi-layered model of Deep Neural Networks (artificial neural networks with multiple layers interconnected between their inputs and outputs [9] [10]) and thus predict the probability of an accident in 15 minute intervals.

To solve our problem, we will seek to develop a mechanism to use location data as one of the variables in the machine learning method along with the meteorological and temporal variables of the events in the dataset. Our main goal is to build a model that allows us to obtain the

978-1-7281-8328-2/20/$31.00 ©2020 IEEE

probability for the possible occurrence of a traffic accident by extending the dataset with meteorological information and non-accident patterns. We expect to outperform the performance obtained by the authors that published the dataset [8], where they have used Gradient Boosted Trees and Artificial Neural Networks for the predictions.

The article is organized as follows. In the next section, we briefly introduce the explainable machine learning models used in this works. In section 3, we explain the *Data Science* methodology for Knowledge Discovery in Databases used in this research. The performance results of our proposal are given in section 4. Finally, concluding remarks and future works are given in section 5.

2. Theoretical Framework

The main technique we will use to obtain the probability of the occurrence of an accident is known as `Random Forest` [11], which consists of a large number of individual decision trees that operate as an *ensemble*, that is, we use a combination of decision trees to obtain a better predictive inference than if we only used a single decision tree. This classifier, when making a prediction, evaluates each tree to obtain its class prediction based on the input variables, then the class with the most votes (the result delivered by the majority of the trees) becomes the output for this particular evaluation. A good performance for this classifier will depend on the number of trees used and a low correlation between them, so by putting all the predictions of these trees together, better results can be obtained than any of those that would be delivered by each one individually. This can be achieved because each tree protects the others from their individual errors, so as long as there is a group of trees that decide correctly the class to which the data belong, the classifier will generate good results [12], [13].

To achieve a low correlation between all decision trees, two methods can be used: `Bagging (Bootstrap Aggregation)` [14] and the `Feature Randomness` method. On one hand, the `Bagging` method consists in that, at the time of generating each tree for our forest, they obtain a random sample with the possibility of replacement from our dataset, which allows us to generate different trees, since they are very sensitive to the data with which they were trained. On the other hand, we have the `Feature Randomness` method, which consists in, when preparing the data to train each decision tree, we only give them one feature of the random dataset, to force even more variation between all the trees and therefore improve the final results when making predictions.

As an extra mention, we can review `Boosted Trees`, which are built through several iterations and modifications to the original dataset, this way we can transform low performance trees into new trees that allow better predictions by modifying both the same tree and the set it was created with, while it is trained. One of these most known methods of boosting is Gradient Boosting, which is widely used through the `xgboost` [15] library.

3. Materials and Methods

To achieve our general objective, we hae applied a *Data Science* methodology for Knowledge Discovery in Databases, where we have accomplished the following activities. i) We have obtained a dataset with multiple events and records of these accidents; ii) we have performed a preprocessing and cleaning of this data; iii) we have designed the technique to be used to manage the location and time variables; iv) we have extended the dataset with events that do not represent accidents; v) we have prepared the data for the classifier; vi) we have performed the classifier training; vii) and finally we have evaluated the generalization performance with the test set.

3.1. Dataset description

We have used the dataset called `US Accidents - A Countrywide Traffic Accident Dataset`, which was created by Moosavi [16] and it was obtained from the platform *Kaggle* in `CSV` text format.

3.2. Data cleaning

Based on what was observed in the *Kaggle* platform, we can see that the dataset contains a large number of null values in certain columns, so it is necessary to clean and eliminate data that we do not consider important. In this way, not only the required disk and RAM space to store the set is reduced, but it is also possible to speed up the operations performed on it.

3.3. Usage of geolocation and time variables

The problem of geolocation variables is that, due to the high dispersion in their values, they do not manage to generate a correlation between the occurrence of an event and its position, since the precision of a latitude-longitude tuple is too high to generate a large-scale analysis like the one we are looking for, and that is why we need to reduce the precision of this information using an alternative representation of these data. To achieve this, we will use the library `H3 - Hexagonal hierarchical geospatial indexing system` [17], developed by *Uber*, which allows us to transform a set of coordinated pairs into a string of characters representing a hexagonal area on the map, along with the ability to decide the level of accuracy and size of these hexagons. Then, in order to use this information in our model, we simply have to give the identifier of the hexagon generated from the coordinates of the original event.

For the temporal data in the events, we decided to use only the month, day of the month and the time the accident occurred in order to facilitate the prediction of new events in the future using only those three temporal parameters.

3.4. Extending the dataset

The dataset obtained only contains data on accident occurrences, but when using machine learning techniques, we need data that represents both classes that we seek to predict, however the non-occurrence of an accident is not contemplated in the original data.

To extend our set we have done the following:

1) We take an event in the original set.
2) We make two copies of the event, we edit the time of each event increasing and reducing the date field of the accident by one hour respectively. At this point we will assume that no other accident occurred in the hour before and after of the original event.
3) We queried the *API* of weather and climate data used by the authors of the original set.
4) We assigned the class "Normal" to these new events and inserted them into the extended dataset.

However, while reviewing the previously used *API*, we noticed that the service is no longer available and was terminated a year ago, so we developed a tool that allows us to obtain the necessary data for a given latitude and longitude using techniques of *web scrapping*, which will be described in the section 4.5.

3.5. Data preparation for the classifier

Once we have extended our data, we must remove those columns that we are not going to use, such as the latitude and longitude pair, the date in which the event occurred along with performing the transformation or coding of those categorical data by numerical codes, since the implementations used do not allow us to enter categorical data into the classifier.

3.6. Fitting the Machine Learning Techniques

Once these steps are completed, we can build our model and make predictions. We will build three implementations of our model and evaluate the performance of each one, these will correspond to use only one decision tree, use `Random Forest` and finally we will evaluate the performance of a `Gradient Boosted Tree`.

4. Results

4.1. Description of the dataset

The dataset to be used consists of almost three million records of traffic accidents in the United States covering 49 states [16]. The data collected in this set begins in February 2016 and ends in January 2020, all obtained from different data providers over the Internet. All records contain data such as date, city, state, time, weather, the presence of any traffic-related objects and several other attributes that describe the complete situation in which each accident took place.

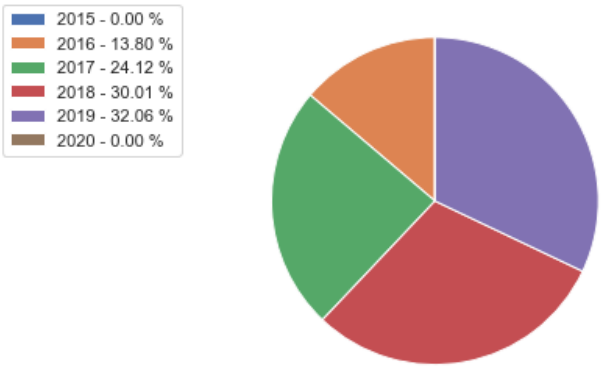

Figure 1. Pie chart with the percentage of accidents that have occurred over the years in the dataset.

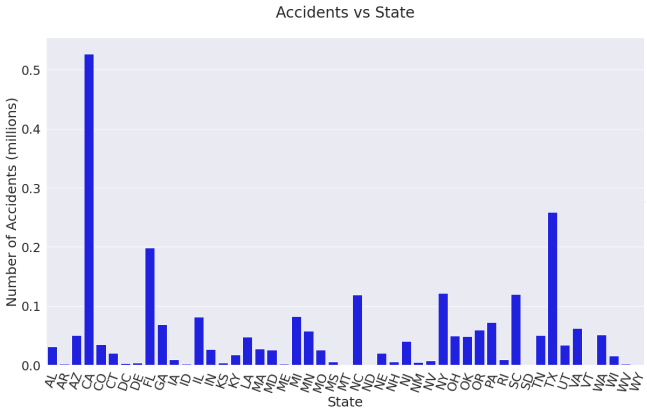

Figure 2. Bar chart with the number of accidents in each US state. California contains the most accidents in the dataset.

4.2. Descriptive analysis of the data

Using the initial dataset and the `pandas` library [18] of the `Python` programming language, we managed to obtain the following information about the dataset, figure 1 shows the distribution of accidents over the years, figure 2 shows the number of accidents by state, figure 3 shows the number of events during the different months of each year and figure 4 shows all the events according to the state where they occurred.

Based on what can be seen in the figures 2 and 4, we can notice that most of the accidents in our dataset occurred in the state of California, that is why from now on we will concentrate on these data to make our model and thus, facilitate the handling of the large volume of data in the original set.

To appreciate in more detail where these accidents occur in California, we can make a *scatter plot* of a sample with 10 000 events in figure 5 and get a better glance the distribution of these events through the territory.

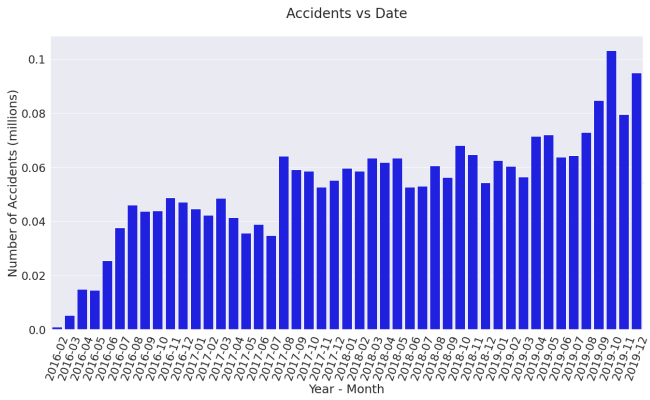

Figure 3. Bar chart with the number of accidents over the years for our dataset.

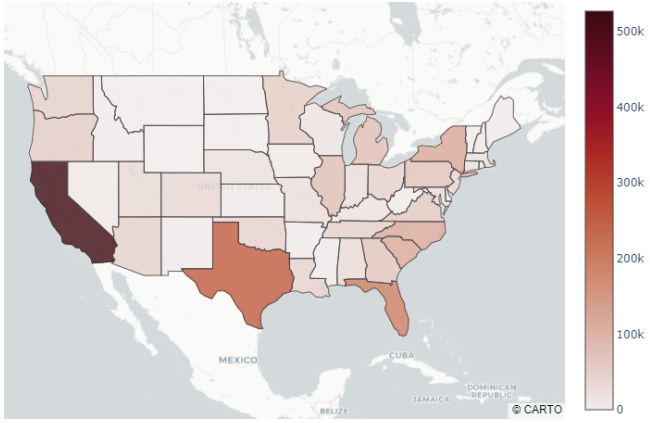

Figure 4. U.S. map colored according to the number of accidents in each state.

4.3. Implementation

In this section, we will explain how we accomplished the main goal described above.

4.3.1. Obtaining a dataset with accident records. The dataset to be used was described above and was obtained from the *Kaggle* platform in the link https://www.kaggle.com/sobhanmoosavi/us-accidents.

4.3.2. Performing data cleaning and preparation. The initial set is $1.04GB$ in size on disk and when using `pandas` to process the file, it is stored with a size of $850MB$ in RAM. Also, as explained above, there are columns whose values are mostly null or that we are simply not interesting for us to do this work, so we have decided to remove them.

Based on this, we removed 20 of the 48 columns from the original set, keeping the ones that correspond to location, weather conditions, and environmental characteristics such as the existence of intersections, crossings, traffic signs, and traffic lights among others; then we removed all the rows in which there were null values, after that, we made a

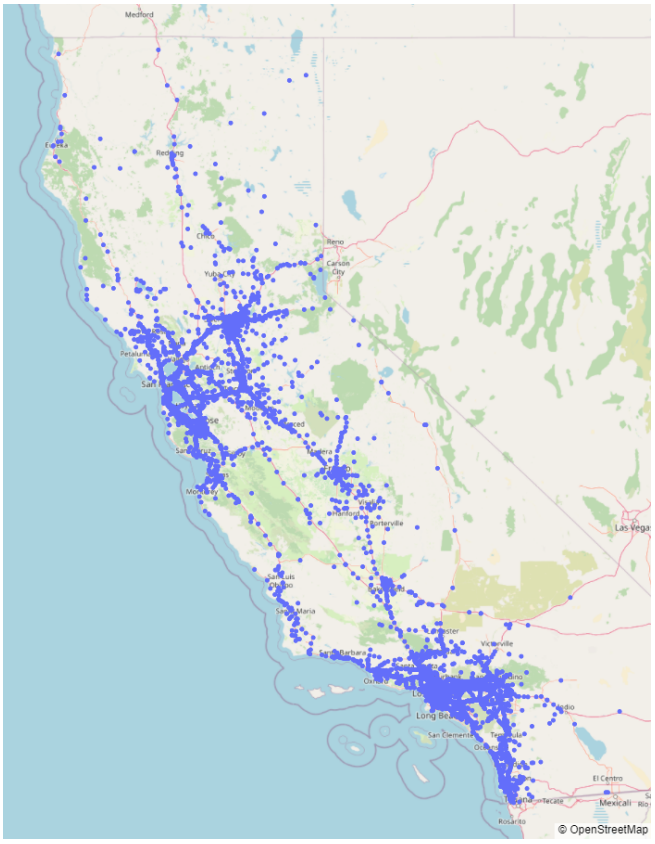

Figure 5. Scatter plot for a random sample of the events in California. $n = 10\ 000$. Each point represents the position of an accident.

transformation to those columns that are stored with a *string* data type to change them for more efficient types such as *category* or *datetime* and as a last step, we stored the new modified set using the `HDF5` format [19]. This is due to the fact that this format stores the information in binary form and not in plain text like the *CSV* format.

Thanks to these operations, we managed to reduce to $225MB$ the memory space needed to use our set, and to $228MB$ the space needed to store it on disk. In addition, the final amount of records is $2\ 506\ 618$ rows. This means that our set was reduced to 22% and 26% of its original disk and memory size respectively by removing only 16% of its original row content.

In addition, as mentioned in the previous section, for this work we will only use events that occurred in the state of California, so our new set contains $527\ 400$ events.

4.4. Using the geolocation and time variables

To attack the problem of accuracy in the location data of each event explained in the section 3.3, we have decided to use H3 [17], a library developed by *Uber* to transform a set of coordinate points into a text string, which represents a hexagon-shaped polygon and contains inside it the coordinate data used to generate it. The advantage of using this method is that we can control the accuracy and size of

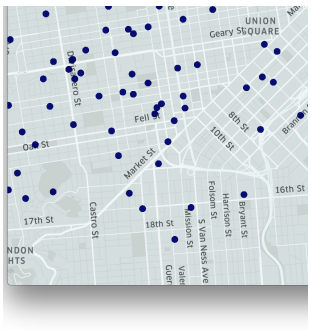

Figure 6. Scatter plot for latitude and longitude.

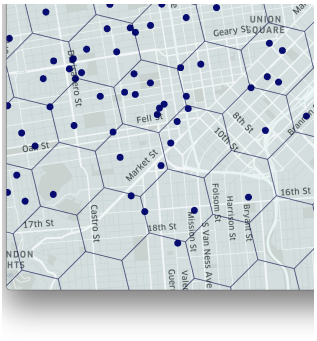

Figure 7. H3 polygons in the observed area.

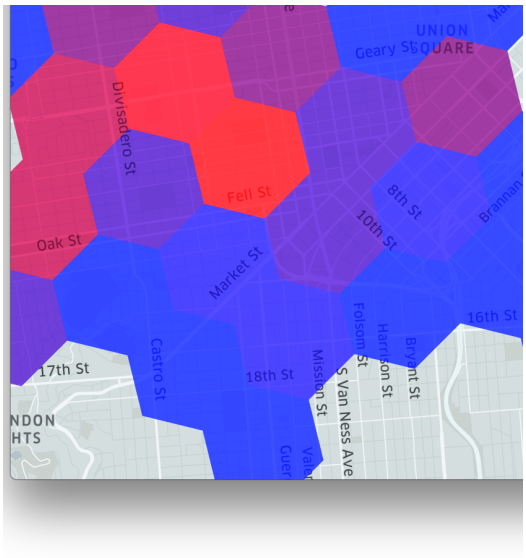

Figure 8. H3 polygons coloured according to the amount of elements they contain.

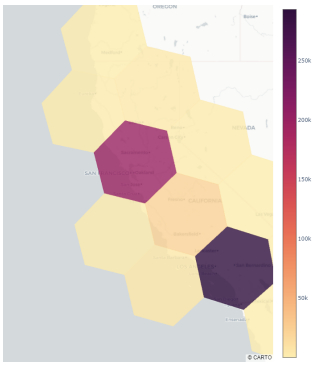

Figure 9. $P = 2$, 12 hexagons.

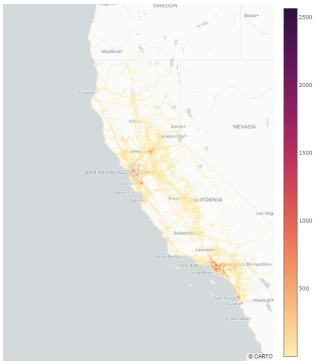

Figure 10. $P = 7$, 12 456 hexagons.

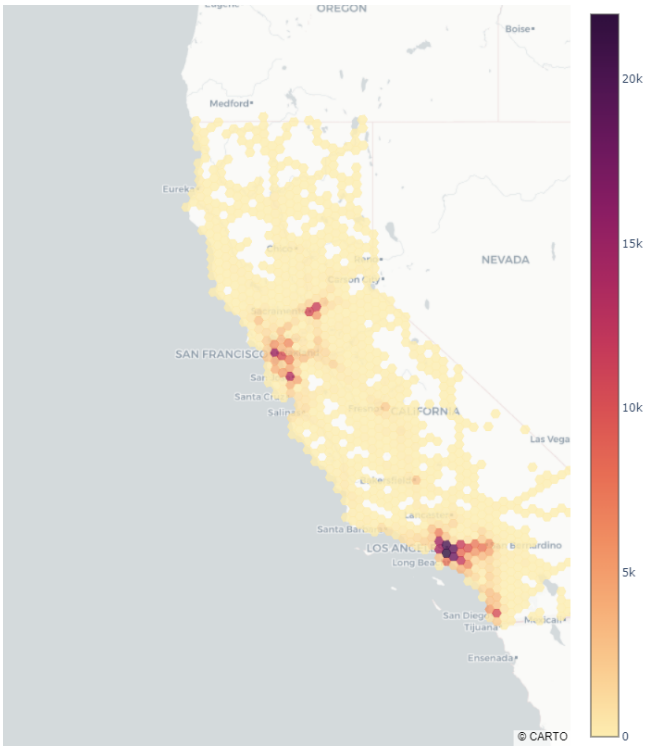

Figure 11. $P = 5$, 1 039 hexagons.

the generated hexagons, which is not possible using only coordinates. In the following figures 6, 7 and 8 you can see how the library works with an example from the *Uber Engineering Blog*.

Applying the same procedure for the geometry of our events in California we can obtain the results shown in the figures 9, 10 and 11 for different precision values, that can take values between 0 and 15, which represent the minimum and maximum precision respectively.

Based on these results, we decided to use a value of 5 for the precision parameter for `H3` which generates a total of 1 039 hexagons with an area of approximately $250km^2$ each. We can notice that using a higher value for this parameter generates a larger number of hexagons, which allows us a higher accuracy in our results, but also implies that we must process a larger amount of data, especially when extending our dataset in the next section.

To use the temporal data into our model, we decided to remove the column containing the timestamp of the event and replace it with three new columns containing the month, day of the month and time of the day when the event occurred.

4.5. Extending the dataset

We have decided to build our own extension of this set considering the time before and after the event as a negative instance, but for this, we need the information of the weather conditions of the moments before and after each event. We have developed a tool that would allow us to obtain the necessary data given a latitude, longitude and an instant of time using techniques of *web scrapping* on the site `The Weather Channel` belonging to `IBM`. For each event, we take the H3 identifier of its parent hexagon, then we

obtain the center of this polygon in the form of latitude and longitude, and finally we query the website for the necessary information in the time interval selected for that particular location. These results are then concatenated to the original dataset. In this way, we increase the size of our data to be used and add information about negative events.

4.6. Data preparation for the classifier

Since we modified the geolocation and temporal data of the original set, we decided to eliminate those columns that contained information such as the county, city, latitude, longitude and timestamp of the event.
In addition, since the Decision Tree and Random Forest implementations of the *scikit-learn* library [20] and *xgboost* do not support the use of categorical data in the construction of their classifiers, we used the data type *Categorical* of *pandas* and then obtained its corresponding numerical value using the *code* property of this type.

In the case of the `H3` identifier, we use the 64bits numerical representation instead of the original string representation.

4.7. Training and prediction

For our three models, we have separated our dataset in 70% to perform the training of the models and the remaining 30% for the validation of results and obtaining the performance metrics for each model.

4.7.1. Decision Tree. Our generated decision tree can be seen in figure 12.

The performance metrics obtained were as follows: `Accuracy 74%, Precision 71% Recall 69% y F1 70%`.

It is important to highlight the variables used by the tree, most of which correspond to weather conditions at the time the event occurred, occasionally checking the `H3` polygon identifier, which shows the important relationship between external weather conditions and the precautions to be taken, according to these, when driving.

4.7.2. Random Forest. The model created using *Random Forest* obtained the following performance indicators: `Accuracy 74%, Precision 71% Recall 69% y F1 70%`. Completely similar to the initial decision tree.

4.7.3. Gradient Boosted Tree. The model generated by the `xgboost` library managed to obtain the following performance indicators: `Accuracy 78%, Precision 79% Recall 73% and F1 74%`. Surpassing the previous models. When we visualize the tree generated in the figure 13 we can notice the same considerations as in the first model, noting the importance of the meteorological variables.

5. Conclusions

From the results of this work, we can conclude that we have achieved our main goal of enhancing the prediction performance up to 74% in the F1 measure of a possible occurrence of a traffic accident given environmental, spatial and temporal conditions. From the three models used, the one that achieved the best performance was the one that uses `Gradient Boosting` to build the decision tree. Furthermore, in all cases it was possible to observe that the climatic variables are the most important ones when it comes to determining the possibility of an accident occurring.

One way to improve the accuracy of our model is to increase the accuracy of the `H3` identifiers, but this implies a considerable increase in the amount of data to be used, this means that we need much more data and time to be able to extend our set with real meteorological data through the *scrapping* method. Moreover, it is necessary to generate a dataset that also considers the non-occurrence of accidents to obtain a better understanding of the causes of these events. To do this, we must consolidate multiple data sources that are not easy to access or that are stored in legacy systems that would not support intensive analysis with greater precision like the ones that we have used. Achieving this would also allow us to get rid of the assumption that another accident did not occur in the hour before and after an event in our extended original set.

Another way to obtain better results is to use another type of classifier that natively supports categorical data, since coding these data makes it more difficult for us to interpret the meaning of the values inferred in the leaves of the generated trees, and in addition, the effect of using H3 in the location data it's lost when we translate the identifiers to integers. Finally, based on the results of the generated models, we can explain that traffic accidents are not events caused by reasons independent of the environment in which they occur, but correspond to a set of conditions that, unfortunately, manage to be fulfilled in the right place, time and under the right conditions.

References

[1] "Global status report on road safety 2018: Summary," World Health Organization, Tech. Rep., 2018.

[2] "Traffic safety facts annual report tables. a compilation of motor vehicle crash data," National Highway Traffic Safety Administration and others, Tech. Rep., 2020. [Online]. Available: https://cdan.nhtsa.gov/tsftables/tsfar.htm

[3] "Evolución de siniestros de tránsito, consecuencias e indicadores (período 1972-2019)," Comisión Nacional de Seguridad de Tránsito, Tech. Rep., 2020. [Online]. Available: https://www.conaset.cl/programa/observatorio-datos-estadistica/biblioteca-observatorio/estadisticas-generales/

[4] Q. Chen, X. Song, H. Yamada, and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[5] Z. Yuan, X. Zhou, T. Yang, J. Tamerius, and R. Mantilla, "Predicting traffic accidents through heterogeneous urban data: A case study," in *Proceedings of the 6th International Workshop on Urban Computing (UrbComp 2017), Halifax, NS, Canada*, vol. 14, 2017.

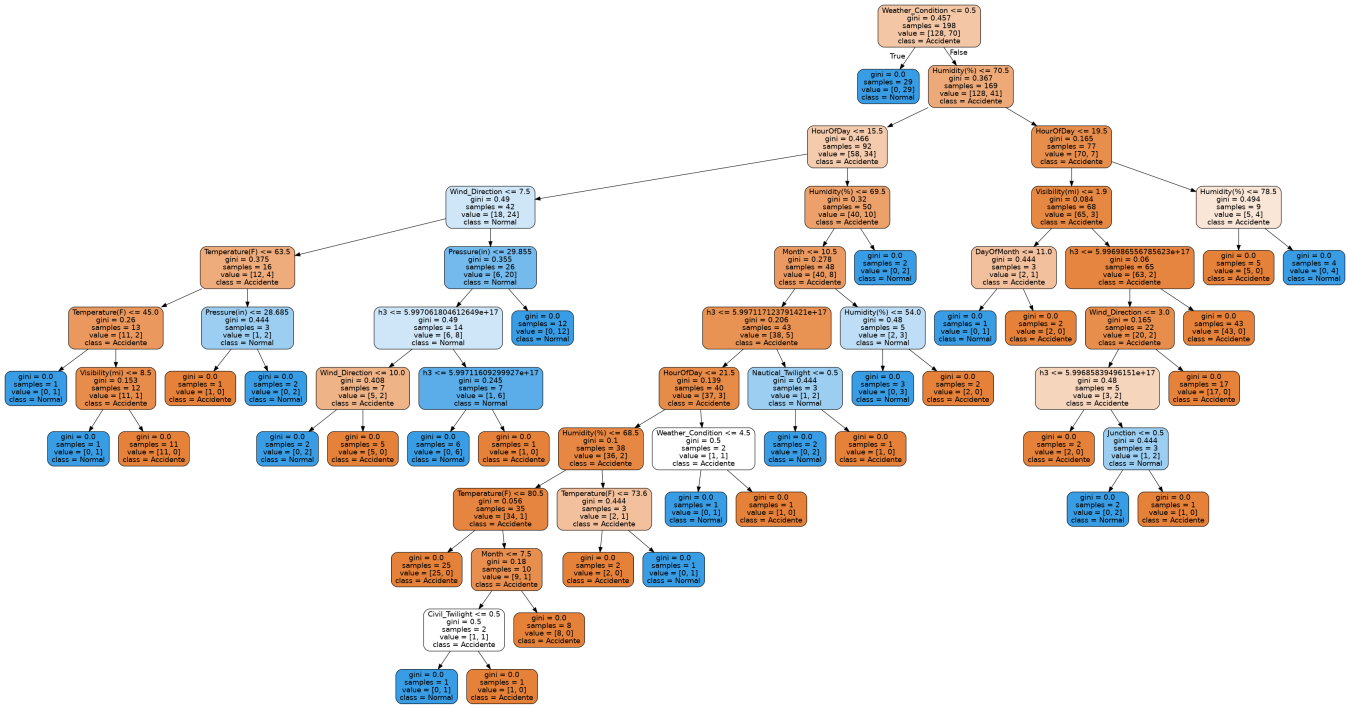

Figure 12. Generated decision tree.

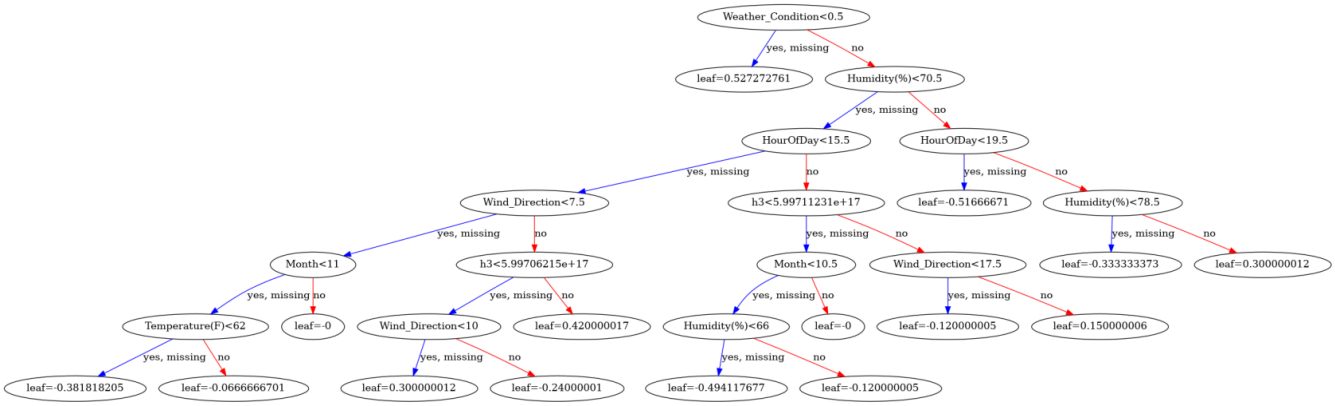

Figure 13. Generated tree with Gradient Boosting.

[6] L. Lin, Q. Wang, and A. W. Sadek, "A novel variable selection method based on frequent pattern tree for real-time traffic accident risk prediction," *Transportation Research Part C: Emerging Technologies*, vol. 55, pp. 444–459, 2015.

[7] L. Wenqi, L. Dongyu, and Y. Menghua, "A model of traffic accident prediction based on convolutional neural network," in *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. IEEE, 2017, pp. 198–202.

[8] S. Moosavi, M. H. Samavatian, S. Parthasarathy, R. Teodorescu, and R. Ramnath, "Accident risk prediction based on heterogeneous sparse data: New dataset and insights," in *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2019, pp. 33–42.

[9] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[10] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc, 2009.

[11] T. K. Ho, "Random decision forests," in *Proceedings of 3rd international conference on document analysis and recognition*, vol. 1. IEEE, 1995, pp. 278–282.

[12] H. Allende, C. Moraga, R. Ñanculef, and R. Salas, "Ensembles methods for machine learning," in *Pattern Recognition and Machine Vision – In honor and memory of prof. King-Sun Fu*, P. S.-P. Wang, Ed. The River Publishers Series in Information Science and Technology, 2010, pp. 247–261.

[13] H. Allende-Cid, R. Salas, H. Allende, and R. Ñanculef, "Robust alternating adaboost," in *Progress in Pattern Recognition, Image Analysis and Applications, 12th Iberoamericann Congress on Pattern Recognition, CIARP 2007, Valparaiso, Chile, November 13-16, 2007, Proceedings*, ser. Lecture Notes in Computer Science, L. Rueda,

D. Mery, and J. Kittler, Eds., vol. 4756. Springer, 2007, pp. 427–436. [Online]. Available: https://doi.org/10.1007/978-3-540-76725-1_45

[14] S. Leo Breiman, "Bagging predictors," Technical Report, Tech. Rep., 1994.

[15] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[16] S. Moosavi, M. H. Samavatian, S. Parthasarathy, and R. Ramnath, "A countrywide traffic accident dataset," 2019.

[17] I. Brodsky, "H3: Hexagonal hierarchical geospatial indexing system," *Uber Open Source. Retrieved from https://github.com/uber/h3*, 2020.

[18] J. Reback, W. McKinney, jbrockmendel, J. V. den Bossche, T. Augspurger, P. Cloud, gfyoung, Sinhrks, A. Klein, S. Hawkins, M. Roeschke, J. Tratner, C. She, W. Ayd, T. Petersen, MomIsBestFriend, M. Garcia, J. Schendel, A. Hayden, V. Jancauskas, P. Battiston, D. Saxton, S. Seabold, A. McMaster, chris b1, h vetinari, S. Hoyer, K. Dong, W. Overmeire, and M. Winkel, "pandas-dev/pandas: Pandas 1.1.0," Jul. 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3964380

[19] M. Folk, A. Cheng, and K. Yates, "Hdf5: A file format and i/o library for high performance computing applications," in *Proceedings of supercomputing*, vol. 99, 1999, pp. 5–33.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.