LDSSA 2018 Capstone project
Report 1

Gábor Somogyi

2018-09-16

Abstract

The objective is to build a system that predicts the seriousness of an accident based on the circumstances under which it happened.

This report describes the context, the data at hand, the models built on them, and provides steps to reproduce the deployment and summarizes the findings and possible risks.

Starting from a provided training dataset, after a quick exploration of the data, we established a baseline model based on Logistic Regression, which reached a ROC AUC score of 0.6135, with a standard deviation of 0.218. After properly encoding the NA-like categorical variables, dropping the narrowly represented and possibly confusing age column, simplifying the data, the selected XGBoost classifier delivers a ROC AUC of 0.6111, but with a lower standard deviation of 0.124.

This classifier was deployed into production, with the expectation of refining the pipeline and redeploying once more data becomes available.

Context

Insurance Inc. aims to create a new health insurance product that offers ultra-low premiums on the simple condition that deductibles are subject to change at any time. To fulfill this expectation, Insurance Inc. needs to be able to assess your risk in a traffic situation in real-time so that your deductible can be adjusted. The data for these decisions is coming from a smartphone app installed by the customers.

To able to provide the most accurate assessment, Insurance Inc. is partnering with Ethical Data Science Inc. to build a system that predicts the accidents' outcome based on the actual circumstances.

The system to be built will be reachable via a web-based service that responds to requests including the circumstances data with the probability of a serious injury, so the app can use this information to update the deductibles of the customer in real time.

Data

In this section, we analyze the dataset provided, describe the information found, and explore correlations and possibilities for feature engineering and simplification.

Data description

According to the data description, each row in the dataset corresponds to a single person, involved in a car crash, moments before it occurs. It contains information about the circumstances and the observation that it resulted in a serious injury or not. The dataset was made available without any extra information about its contents beyond the general context.

The dataset is split to 2 files, X_train.csv containing the training data, y_train.csv containing the target data. X_train.csv contains a header, identifying the data columns, while y_train.csv is a simple array of integers.

The 2 files both contain 11173 data rows, this way the dataset can be merged using the position of the rows.

Here is a detailed breakdown of the columns found in the dataset:

X_train.csv

Name	type	short description	NA values	NA %	Examples
m_or_f	categorical	person involved is male or female	1	0.009%	m, f
person_attributes	categorical	activity the person is doing	0	0.000%	driving, passenger, on_foot
seat	categorical	position of the person in the vehicle	0	0.000%	front left, front right
other_person_location	categorical	location of the possible other person	9883	88.454 %	middle_of_road
other_factor_1	categorical	extra information, if applicable	10276	91.972%	was_doing_tide_challen ge
other_factor_2	categorical	extra information, if applicable	10867	97.261%	
other_factor_3	categorical	extra information, if applicable	11095	99.302%	
age_in_years	integer	age of person involved	0	0.000%	

y_train.csv

Name	type	short description	NA values	NA %	Examples
unnamed	integer	1=serious injury, 0=serious injury	0	0.000%	0, 1

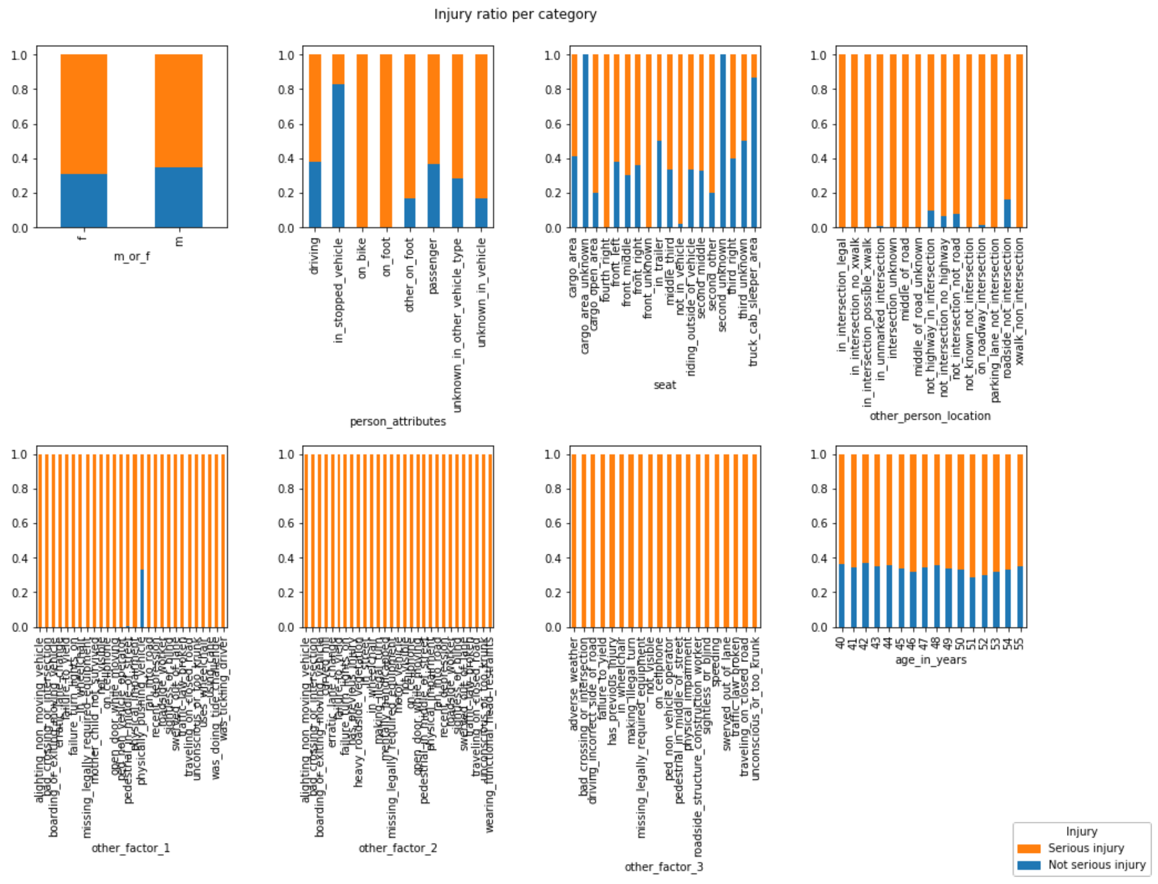
Exploration

After listing the values of the categorical columns we found that there are several values that indicate they would hold NA values: *“unknown”*, *“N/A and Unknown”* and *“N/A”*, we encoded these as NA values. Several other values include the string “unknown”, like “front_unknown among the *seat* categories, but that was kept as original.

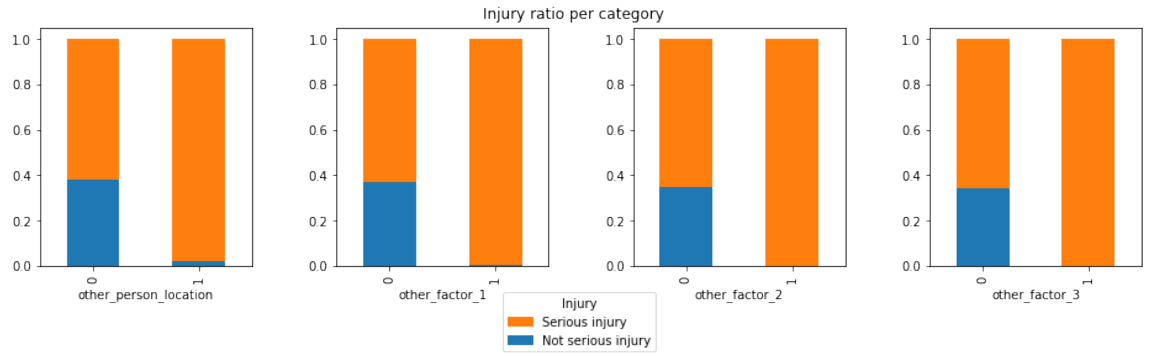
The values of age are only representing in a narrow range (40-55) even though there was no indication that the dataset would be only targeting this specific age group. Additionally, the age values show an extremely weak correlation with the injury seriousness, having a correlation value of 0.02.

The columns *other_factor_1*, *other_factor_2* and *other_factor_3* are sparsely populated (92.0%, 97.3% and 99.3% NA values, respectively), but where they hold value, the outcome was a serious injury. Similar is the case of the *other_person_location* column, where very few values do not indicate serious injury if a category is present.

The figure shows the correlation of the various categories (NA-s are omitted) to the injury seriousness:



Encoding the *other_person_location, other_factor_1, other_factor_2* and *other_factor_3* columns into NA and non-NA values (encoded as 0 and 1, respectively) is shown in the next figure:



The data distribution is very skewed, 8500 cases, 76.08% of the dataset represents a driver sitting on the front left seat, and the *other_person_location, other_factor_1, other_factor_2* and *other_factor_3* columns are not set. An additional 1153 cases, 10.32% of the data shows a passenger sitting on the front right seat, with the aforementioned 4 columns not set. The remaining categories represent less than 1/7 of the dataset.

Cleaning

The data is generally clean, categories are named concisely, no out of picture data (non-integer in integer column, non-string in categorical, no extremely similar values that could suggest typos in naming). As described in the beginning of the exploration section, "*unknown*", "*N/A and Unknown*" and *"N/A"* were encoded as NA values.

Feature engineering

As discovered in the exploration section, the values of the *other_person_location, other_factor_1, other_factor_2* and *other_factor_3* columns only hold value if they are not filled. Therefore we recoded them as 0 for the NA values and 1 for any non-NA values.

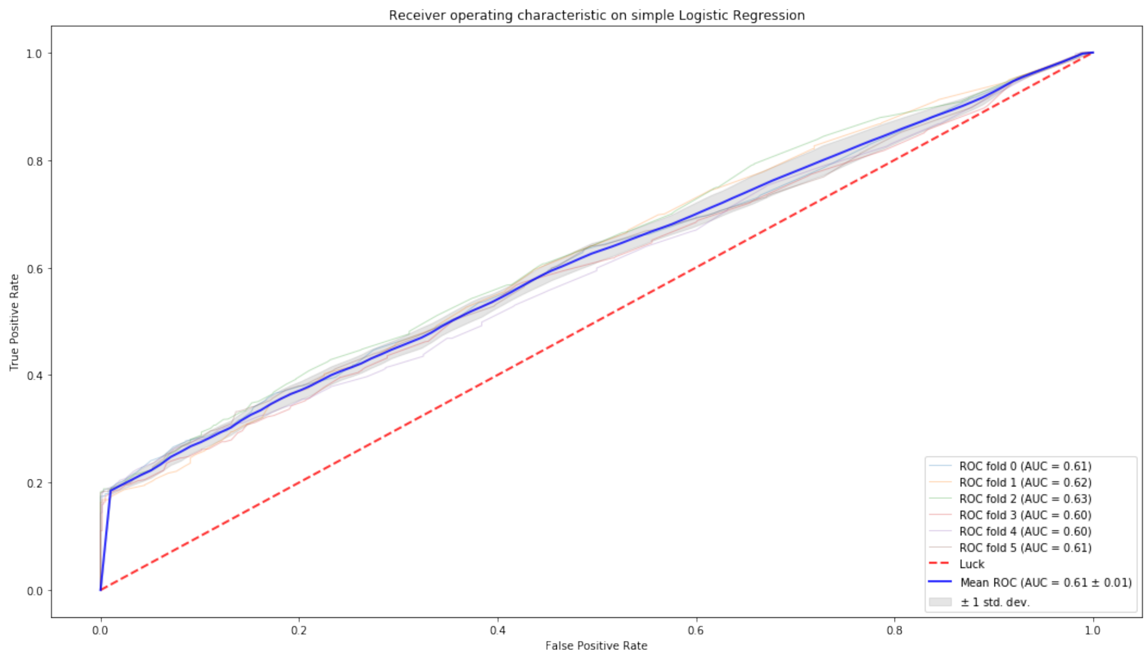
Additionally, as categorical values cannot be interpreted directly by most classifiers, one-hot encoding was used to encode the remaining categoricals.

Modelling

In this section we present the baseline model used, the further model selection and evaluation, and finally the chosen model that was deployed.

Baseline

As a baseline model, we chose a Logistic Regressor. As the Logistic Regressor doesn't handle categorical data directly, a one-hot encoder was used to encode the categories. This resulted in a cross-validated result of a ROC AUC of 0.6135, with a standard deviation of 0.0218. The following figure shows the ROC values on a 6-fold cross-validation:



After establishing the baseline we applied feature engineering, namely encoding the categorical variables of the *other_person_location*, *other_factor_1*, *other_factor_2* and *other_factor_3* columns.

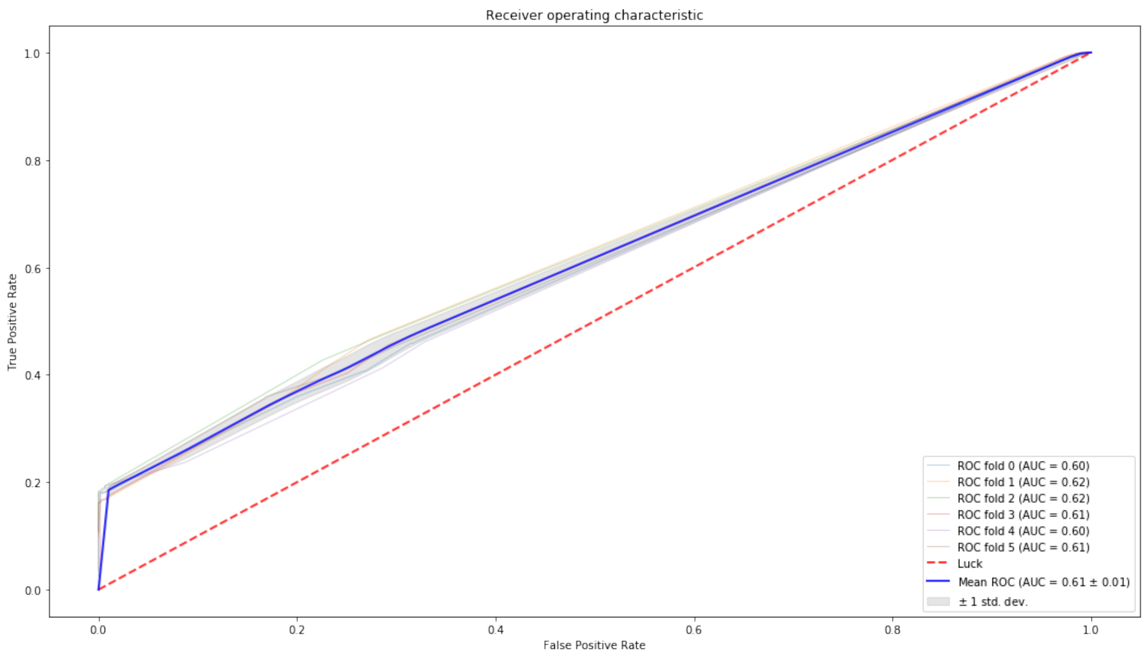
Additionally, we decided to drop the *age_in_years* column. This may lower the scores a bit, however the narrow data range brings in risks of meeting unknown data later on, offsetting the minimal correlation found in the dataset between age and injury seriousness.

Evaluating several classifiers yielded the following results:

Classifier	ROC AUC	standard deviation
Nearest Neighbors	0.582487297	0.027643108
Decision Tree	0.608550416	0.013164203
Random Forest	0.607824338	0.01186485
Neural Net	0.603417231	0.013553374
AdaBoost	0.603130026	0.017115457
Naive Bayes	0.605293626	0.01399141
XGBoost	0.610293066	0.013019572
LightGBM	0.608140869	0.012830953
Logistic Regression	0.605291016	0.017396996

From these classifiers the XGBoost scored the highest values, so we tuned it with a GridSearchCV algorithm that resulted in the ROC AUC result of 0.6111, with a standard deviation of 0.0124. Although the ROC AUC is lower than the baseline value, the significantly lower standard deviation ensures that this classifier is more robust thant the baseline. Therefore, the tuned XGBoost was chosen and deployed to the model.

Visualization of the ROC AUC of the XGBoost classifier:



Deployment

The frontend deployment happens on the Heroku platform, based on the description at https://github.com/LDSSA/heroku-model-deploy

The frontend connects to a PostgreSQL database server running on VPS on docker. The configuration for the PostgreSQL database can be found in the backend directory, and can be deployed with the *docker stack deploy –c postgres.yml* command on the desired server.

The code can be deployed to Heroku as by deploying the contents of the *deploy* folder with creating a new heroku instance, and uploading the folder via heroku git, using the *git commit heroku master* command.

The code also includes a CI configuration set in *.gitlab_ci.yml,* which can be used on GitLab runners to provide a CI pipeline. Using this runs the unit tests defined in *test_app.py* on each commit, and deploys to the heroku server set in the GitLab environment variables when tests pass, and the code gets merged to master.

If we define a staging server and production server on heroku, we can also define a pipeline, and after deploying to staging, when we feel the server is ready for production, it staging can be promoted to production manually either on GitLab, on Heroku, or from the Heroku CLI.

Expected Results and key risks

There are several key risk factors present:

The original dataset was highly skewed, 6/7 of the data represents a driver in the front left seat or a passenger in the front right seat, where we only know the age and gender besides these information, which make it difficult to accurately predict the outcomes.

The age data represents only a narrow fraction of the population, making it difficult to make predictions for unknown age ranges.

The training data didn't include typos, unknown formatting and data types, the presence of those would also be a possible source of issues.

Based on the ROC AUC scores we expect a moderate accuracy, that should improve by incorporating new data coming in after deployment.

Results and discussion

The deployed model has a ROC AUC of 0.6111, which is not exceptionally high, but the expectation is that this will raise with retrained models from the new data.

The system is deployed, and available at https://ldssa-gs.herokuapp.com, delivering predictions for the inputs.

We can make several recommendations for the future:

Collecting more and more diverse test data would ensure better model training

Incorporating more metrics, data dimensions would enable the modeling to provide more accurate results.