LDSSA 2018 Capstone project
Report 2

Gábor Somogyi

2018-09-30

Abstract

The system managed to achieve a ROC AUC score of 0.5924 on the first half of the dataset, which is while below the original expectation of 0.6111, is still clearly above the random score of 0.5, that it has potential for further use.

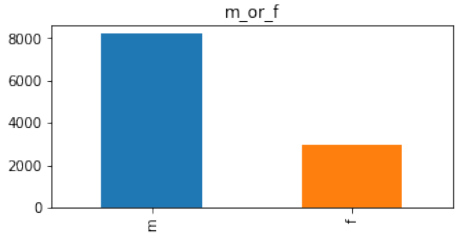
The system built has proven robust enough to manage the requests, only 7 of the 9900 prediction requests got unanswered (about 99.93% response).

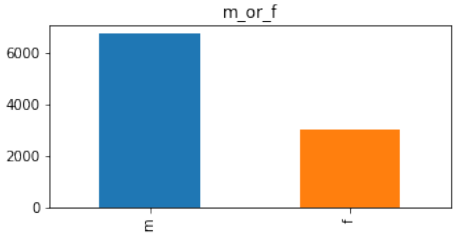
Analysis of Live Data

The live data proved to be similar to the test data, with a few notable differences. Age was represented all over the board,

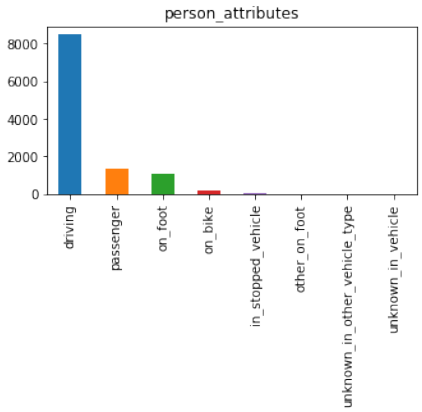
The data was generally clean, no scrambled data, no unexpected data types (e.g numbers in place of the categoricals), the main difference was in the distribution. NA values were mainly present in the other_factor variables, similarly to the training data.

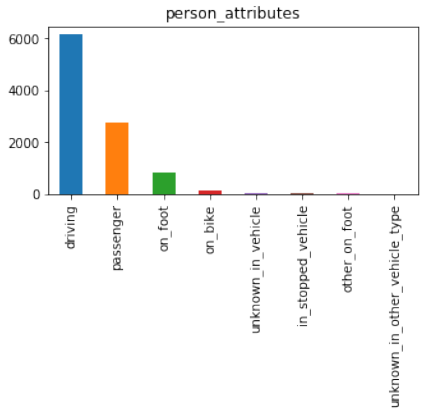
Column-by-column comparison



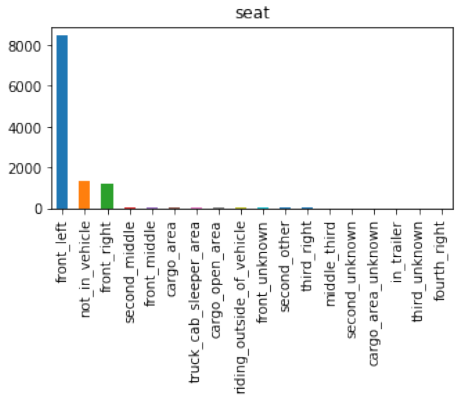


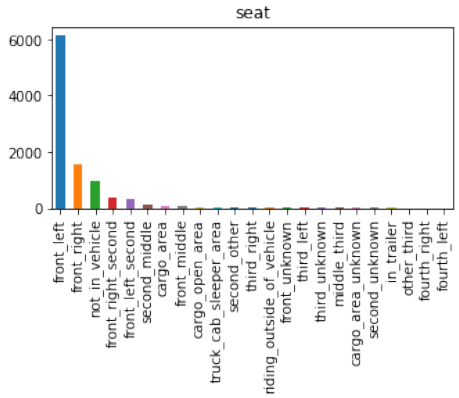
Gender didn't change much, a bit more female participants, and 123 NA values compared to the training dataset, alongside with 6731 male and 3037 female drivers.



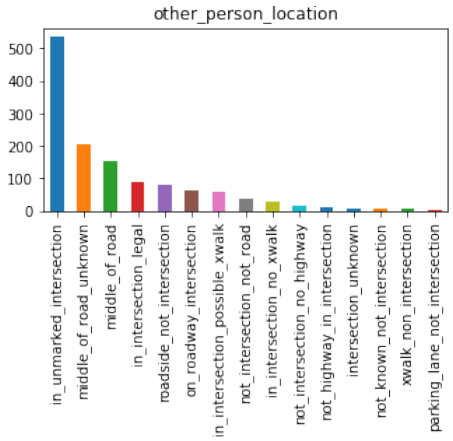


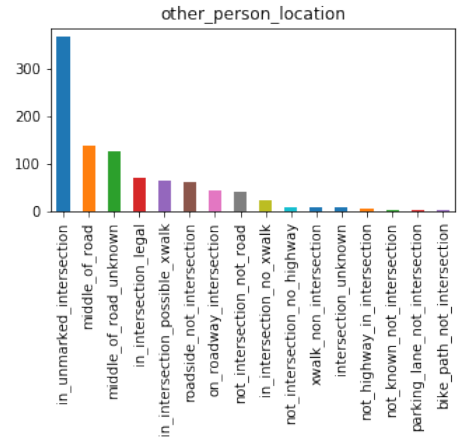
In the person attributes, the majority was still driving (6157), but passengers have greatly increased (2747, nearly doubled). On foot and on bike kept similar representation (806 and 131 cases)



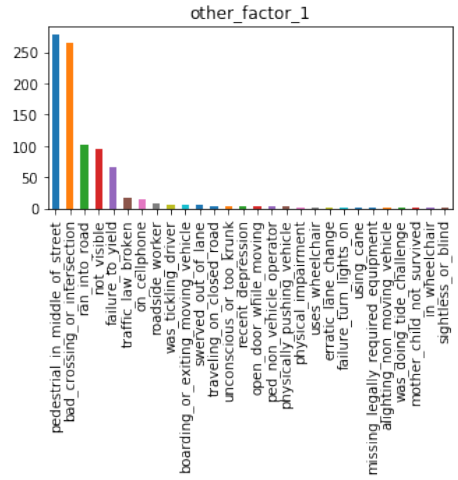


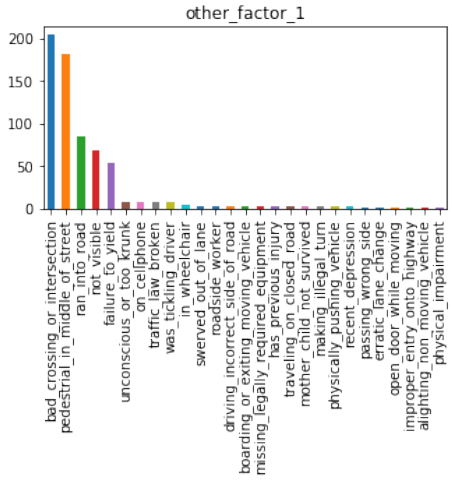
Seats had 127 NA values, and the leading location was front left in both datasets by a wide margin (6156 in test), followed by front right and not in vehicle similarly to the training set (1563 and 971 occurrences). Front right is slightly higher represented in the test set, and also the next categories have more presence (front right second and front left second) that were not present in the train data. They could either mean the front left or the front right respectively, or the middle seat. This could be explored in more detail going forward.



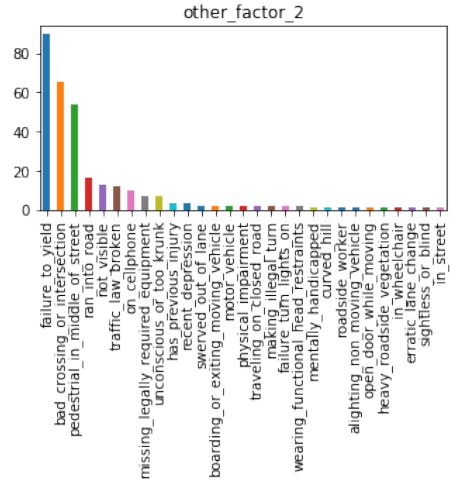


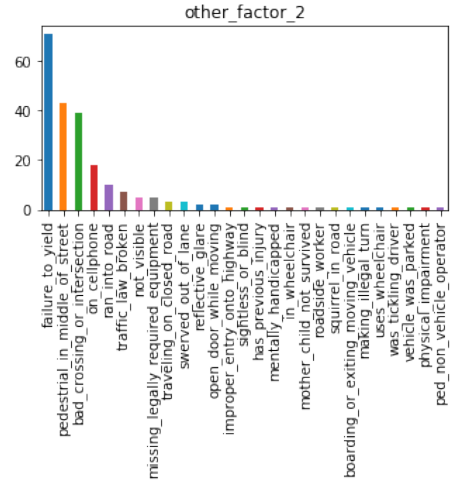
In Other person location, the vast majority of the data was NA (8938 cases), with the leading of in unmarked intersection in 367 cases (similarly to the train data), followed by 138 and 126 cases of middle of the road and middle of the road unknown, resembling the training set.



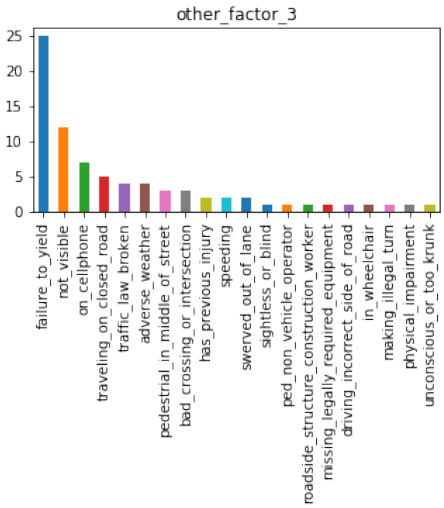


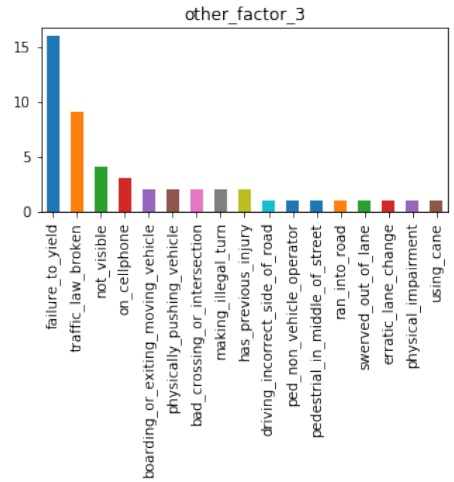
In Other factor 1, the vast majority of the data was NA (9241 cases), with the leading of bad crossing or intersection in 205 cases, followed by 182 cases of pedestrian in the middle of street, showing a similarity to the training set.



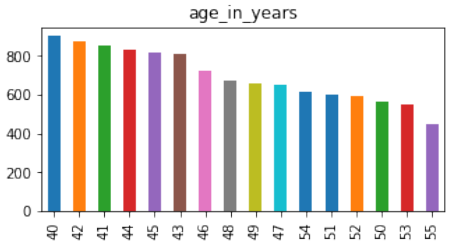


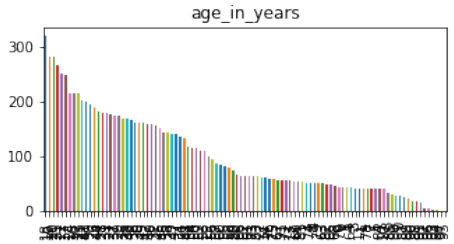
In Other factor 2, the vast majority of the data was NA (9677 cases), with the leading of failure to yield, in 71 cases, followed by 43 cases of pedestrian in the middle of street and 39 bad crossing or intersection, showing a similarity to the training set.





In Other factor 3, the vast majority of the data was NA (9850 cases), from the remaining ones still the failure to yield was the most frequent, but that was only present in 16 cases, followed by 9 traffic law broken, showing a similarity to the training set.

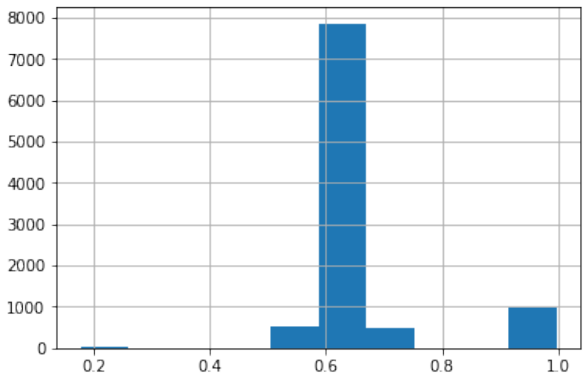




As we can see, age is represented all over the spectrum. The mean is 39.35, however 0 values are also present, so more cleaning would be required before using this data.

Analysis of Model Performance

The distribution of predictions show a pretty narrow range, which means the model was most probably overfit. This is probably originating from the high amount of training data that has both outcomes for the same subset of data (driver in front left seat with only gender and age specified)



We received a set of values containing the true outcomes for the first half of the dataset. However, cross-validating the dataset directly against the data yielded a ROC AUC score of 0.4933. After adjusting this data by shifting the value id-s by adding 1, the model score grew to 0.5924.

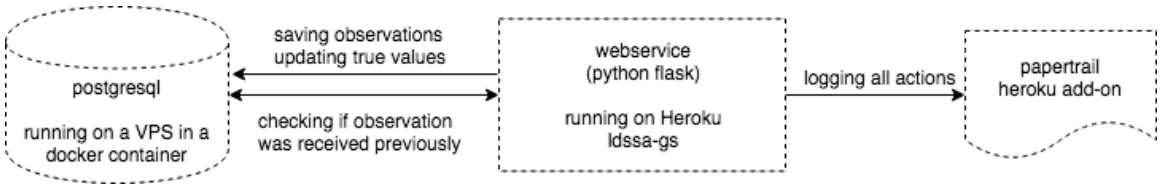
As this was a similar difference to importing the original *y_train* dataset – which had no headers – without specifying *no headers* in the import (and importing the first row mistakenly as headers), we decided this was probably an administrative error on the system side and used the updated data for evaluation.

Redeployment

Retraining didn't happen due to lack of time available. However, the server got redeployed to improve logging quality and stability. It would have been beneficial as the new data showed a much wider variety of age as this was dropped originally on the premise of it possibly disturbing the results.

Production post-mortem

There have been minor modifications to the code and the architecture after the submission of the code repository. Here is the architecture of the system that ran in production:



The model was built robustly enough to not throw errors

- all data except for age was handled as string, accepting NA values
- age was recoded as float and anything not possible to convert to float was handled as NA (since age was dropped later this was simply done for robustness and to keep the possibility to retrain and use the age data)
- to make sure no data gets lost, all incoming requests were logged before introducing them to the pipeline

Errors still occurred and addressing it

- due to an unexpected database restart, 7 predictions got lost and didn't get answered
- the database connection used from the frontend was not fault-tolerant to loss of connection, a manual restart was required
- to address this, the code got updated that even if the database connection throws an error, the prediction gets replied to the requester
- this came in handy when one of the heroku dyno lost connection to the database resulting in sporadic database errors, not saving the data to the database. However, in these cases the predictor did work and reply correctly to the client
- a manual restart itself takes about 30 seconds, and didn't result in missed predictions by itself, as long as the errors before allowed the predictions to happen

In numbers

- failed to predict and save to database: 7 items (observations 733 to 739)
- predicted and saved: 9699 items
- predicted but not saved (only recoverable from logs): 194
- total 9900 observations

Future work

There are several considerations that can be a base of future work. There could have been put more effort into feature engineering, and looking into the correlation of the various categorical values. A wider range of *age* data would have been outright useful, as in the current case it ended up being dropped.

While in the model building the NA values were identified correctly, this was upon loading the data and not upon processing, thus it was missing from the deployed pipeline, which should be corrected in a future implementation.

The model selection and tuning was cross-validated and explored in detail, however ensembling of different models might provide better performance.

The true values received also brought a sort of confusion, as they most probably needed to be shifted by 1 to properly reflect the data while they were supposed to come as verified information.

Possible technical improvements include creating a more robust database connection that would reconnect to the database in case of error. Or alternatively use a more direct database connection (e.g. with sqlalchemy and stored procedures) that doesn't require the server to maintain a persistent database connection.