Characterizing Vehicle Crash Locations Through Machine Learning

Link to GitHub Repository

Introduction

- This project aims to build a predictive model in order to determine if a car crash would occur for a combination of factors.
- The real-world application would be for first-responder agencies to better assess the area surrounding them and be able to preemptively deploy personnel and resources to locations with high crash likelihood.
- The final model was trained with 127 features/variables of boolean nature that encompass groupings such as road physical description (ex. intersection type, lane count), driving conditions (ex. light levels or if the road had any slippery substances), and timing factors (ex. day of the week or if it was a holiday).
- The final goal is to build a classification model with a focus on firstly being able to accurately predict if a combination of features would have a high chance of crash (interpreted as yes/no for a crash) and secondly being able to extract what features contributed the most to the chances of a crash occurring.

Data Source

The dataset used in the models explored was "Accidents in France from 2005 to 2016" from Kaggle

Link to Kaggle Dataset

Note: All data files are available in CSV format through this project's Github link. It is suggested to use those files as the filenames were edited to avoid problems with escape character errors when loading into the Jupyter Notebook.

Data Cleaning & Preprocessing

The final dataset after cleaning has 827,742 observations and 19 columns.

To ensure clean data prior modeling, unnecessary features were removed. Features removed were any feature that was information that would not be known or could not be controlled by the agency before a crash occurred (ex. driver age). In addition features from the original data files that were purely informational such as the id of the accident (which was helpful in merging the datasets before being removed).

Cleaning:

- The data was checked for null values, outliers, duplicate observations, value consistency within each feature, and detectable irregularities.
- Extensive investigation was required for the road lane count feature as there were many observations containing zero lanes also containing other road characteristics that would not be present unless there were road lanes. In addition there were many observations with road lanes above 8. I was unable to find any road in France with more than 8 lanes, making the conscious choice to include exit and entrance ramps to highway equivalent roads. High lane count observations had a range of 9-99.
    - The zero lane issue was addressed by investigating any associated feature that was commonly tied to them, then working through each road description feature to determine the common lane count associated with that feature would be & imputing that lane count in place of the zero lanes originally recorded.
        - ex. if the road type was "autoroute" I imputed 2 lanes or if the road type was one-way I imputed 1 lane
    - Observations with >8 lanes were dropped as they represented less than 0.5% of the total data (leaving multiple hundred thousand observations left to analyze)
- Null values were checked and observations with any null value were dropped as they represented less than 1.32% of the total data (leaving multiple hundred thousand observations left to analyze).

Created Features:

- "sep_present" to denote whether a separator was present between opposing traffic directions. Created based on if the recorded width of the separator was >0
- "ped_present" to denote if pedestrians were present; which is not normally known beforehand or controllable by agencies it can inform on areas where pedestrians would be present. Created based on if the count of pedestrians involved in the accident was >0
- "date" edited from original data to be rounded to nearest hour as agencies are more likely to deploy personnel for hour increments than smaller ranges.
- "day_of_week" (i.e. Mon-Sun), extracted from the date of the accident
- "is_weekend" (0 or 1), extracted from the date of the accident
- "is_holiday" (0 or 1), using the date of the accident and check if the date corresponded to a french holiday (2005-2016 record of French holidays contained in the holidays.csv file)

Preprocessing:

- Categorical features (all features) were one-hot encoded
- Standardization was not performed as all features were categorical with values of 0 or 1
- A target feature "crash" was created with value of 1 for each observation (as only crashes were recorded in the data)
- In order for the model(s) to be able to perform binary classification, the combinations of features not already present in the data were generated and labeled with crash=0
    - This was done generatively with randomness as producing the cartesian product of all features would result in 2^127 observations and require more memory than is available worldwide.
    - A number of combos was generated equal to the number of unique combos already present (about 415,503) with each generated combo compared to the pre-existing set to ensure no duplicates
    - The generated combos were then labeled with "crash"=0 and resampled with replacement to equal the count of crash=1 observations, resulting in a 1:1 balanced dataset.
- As the final step before modeling, the dataset was split into training and test sets with an 80:20 ratio.
    - The train & test sets were also ensured to have similar ratios of crash = 0 & 1

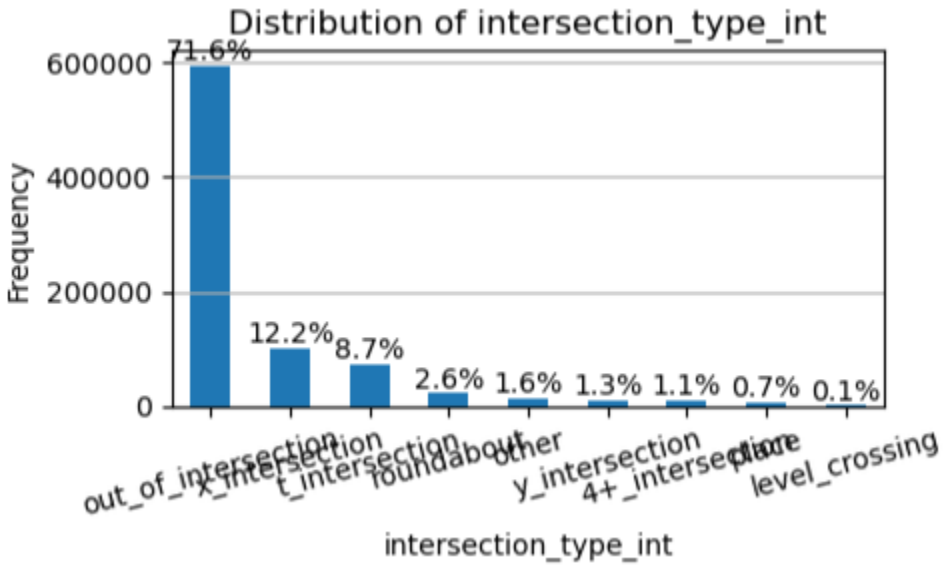
EDA: Highlights

- Municipal (local) roads account for about 50% of crashes
    - highways (departmental, national, autoroutes) account for about 47% of crashes
- About 64% of crashes occur on two-way roads
- About 72% of crashes occur outside any intersection
    - Disproving an early hypothesis that most crashes occur in an intersection
- About 78% of crashes occur on a dry road
- About 83% of crashes occur with no pedestrians present
- About 81% of crashes occur during normal weather (light out, sparse clouds, no precipitation)
    - About 10% of crashes occur during rain
- About 69% of crashes occur during the day
    - About 17% of crashes occur at night with artificial lighting
    - Disproving an early hypothesis that more crashes occur with minimized light
- Hour of day range for elevated crashes:
    - 8am-7pm
    - peak at 6pm (10.4%)
- Friday is most common day of week at 16.7%
    - Other days (except Sunday) hover around 14%
    - Sunday about 12%

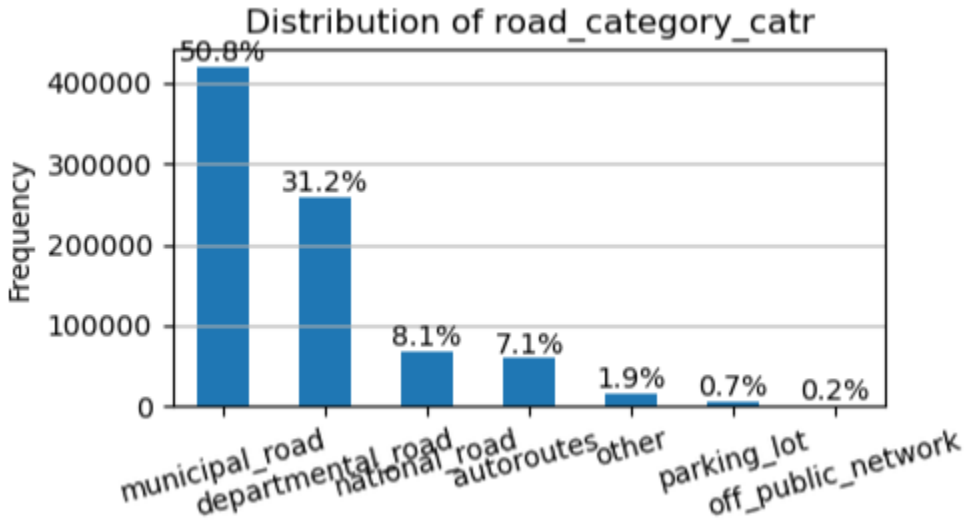
These highlights can be noted in the distribution plots in the project jupyter notebook. A selection of plots that greatly contribute to understanding are included in this report.

Featured Distributions

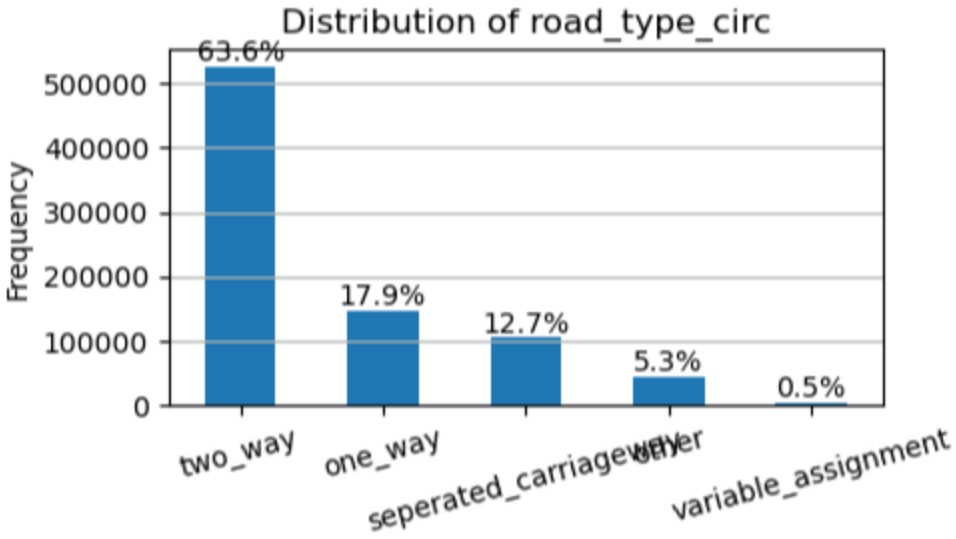
- Intersection Type:



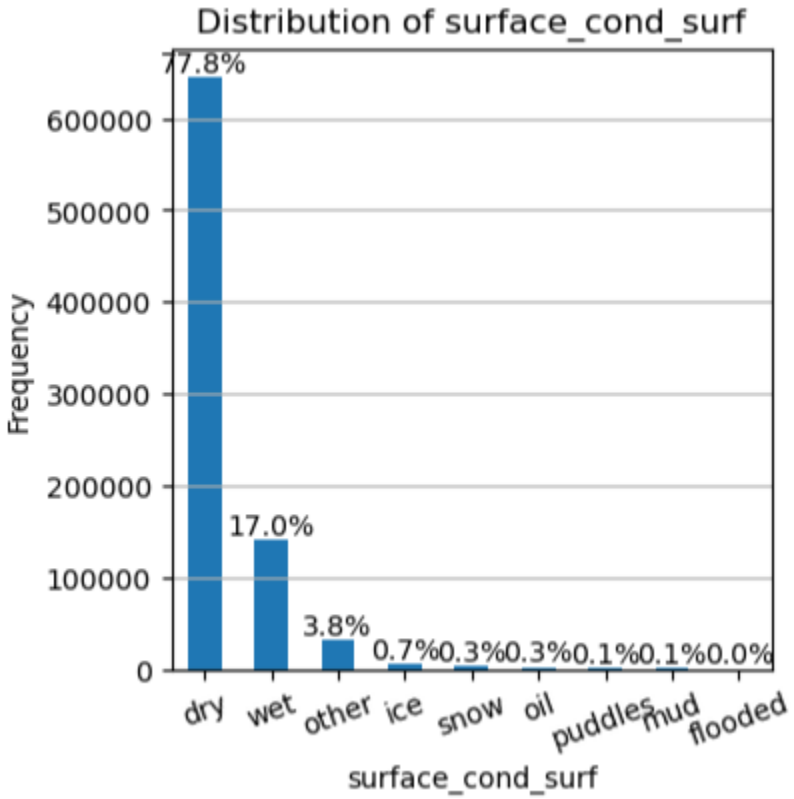
- Road Category:



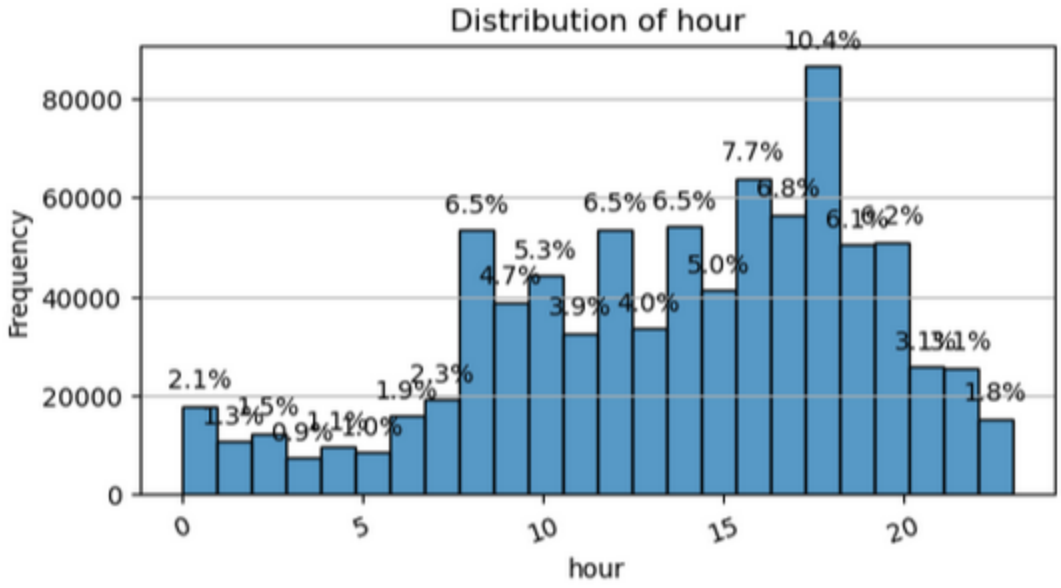
- Road Type:



- Surface Conditions:



- Time of day (range of 0-24)



Interactions Between Features

- Each feature within the groupings (road physical description, driving conditions, weather conditions, and time conditions) were correlated to each other
    - Such as the road type and category (autoroutes always are two-way) or the weather could only be one possibility at a time.

Model Results

3 classification models were chosen for their use in binary classification

- The models also had the requirement of being able to extract the feature importances in predicting a crash
- Logistic Regression (SciKit Learn)
- Random Forest (SciKit Learn)
- Sequential Neural Network (Keras/Tensorflow)

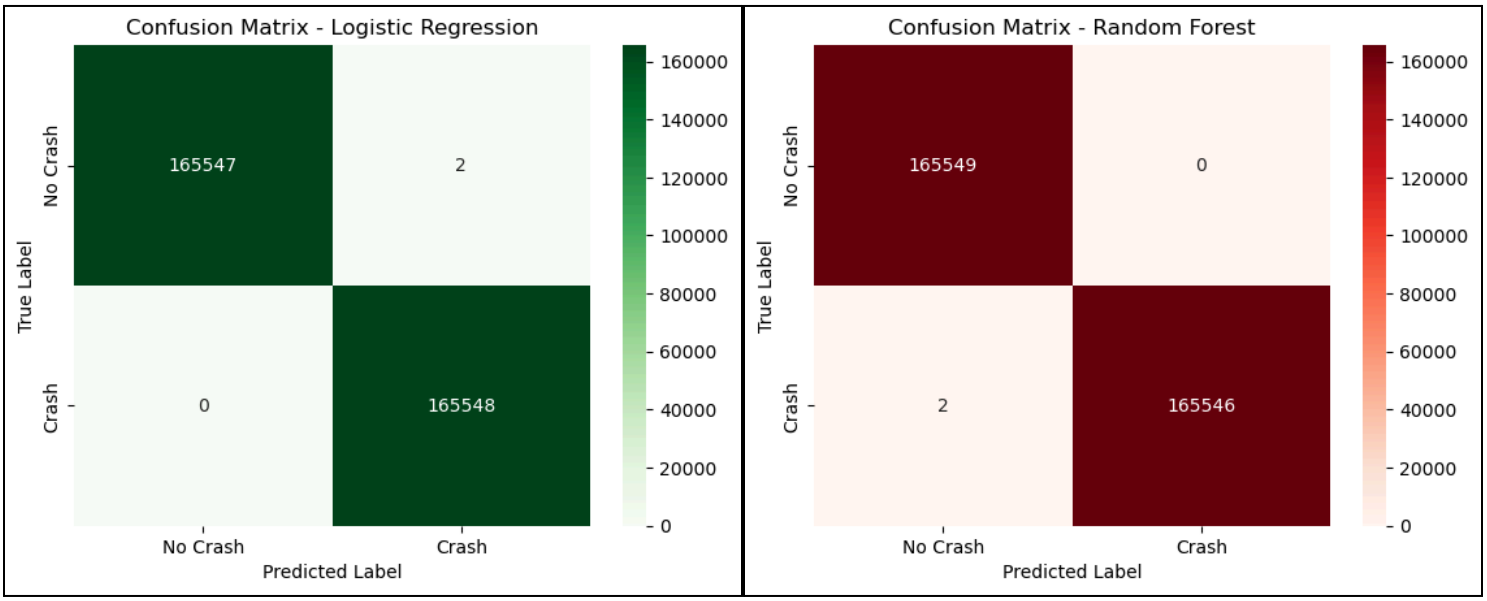
Preliminary Modeling

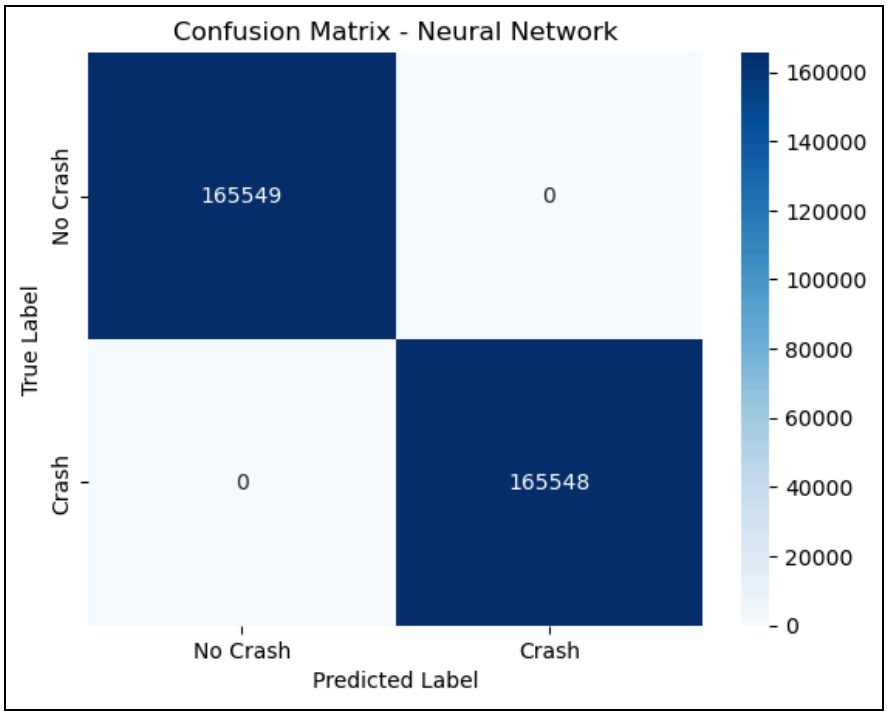
- 3 classification models were trained and scored based on their predictions
- The random forest model started with max_features='sqrt' in order to speed up code execution time.

- Each model achieved 100% accuracy, recall, precision, and f1-score
    - Understandably due to no overlap in feature combinations for crash = 0 or 1
    - This is acceptable and desired as the goal is to increase understanding of where & when these crashes occur
- The neural network model then underwent hyperparameter tuning to determine optimal parameters
    - This was performed more as an instructional exercise as the original model had near perfect metrics already.
    - Hyperparameters tuned specifically include: number of layers and learning rate
        - Other hyperparameters were checked and tuned automatically through the keras function Hyperband, with direction to optimize accuracy
- The hyperparameters determined during the neural network model tuning were saved to a .txt file and reloaded into the jupyter notebook for the model to use those parameters instead of the original
    - The cells containing the code for the original model and the tuning process were kept in-state and converted to raw format cells to prevent code execution
    - This was implemented in order to save substantial time when running the code of the notebook

Models Metrics & Confusion Matrices:

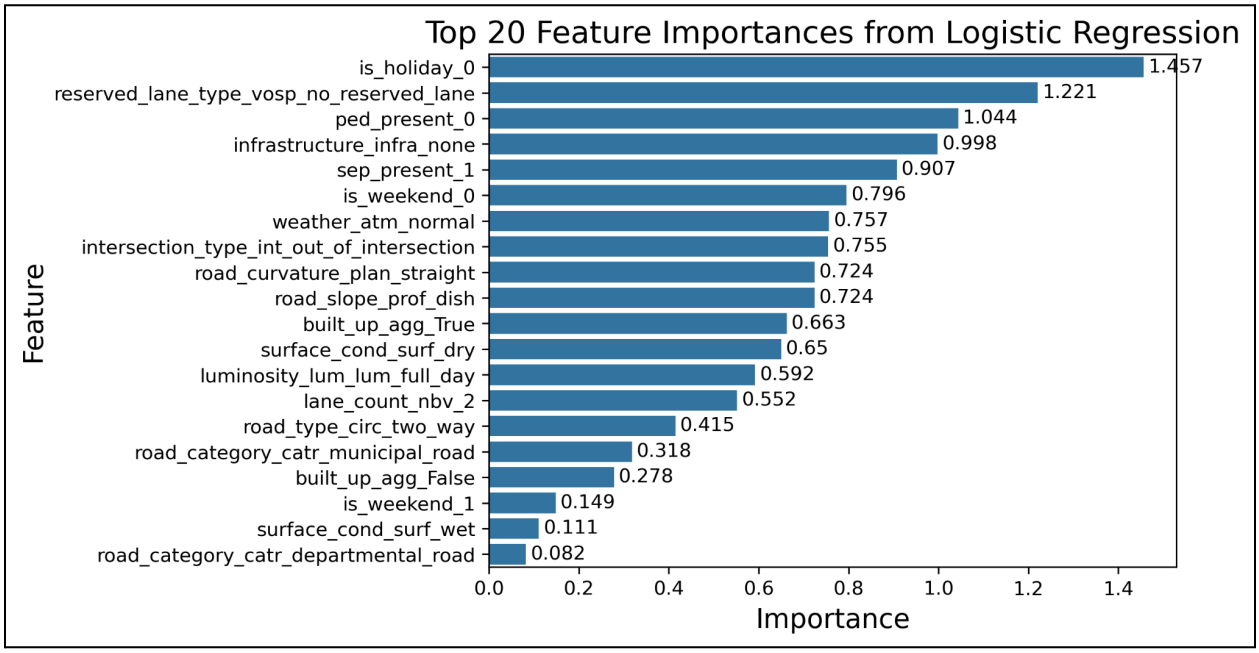
- Each model scored 1 (100%) for each score (accuracy, recall, precision, and f1-score)
- The only information of note is that both the logistic regression and random forest models had 2 false predictions each, as shown in confusion matrices below

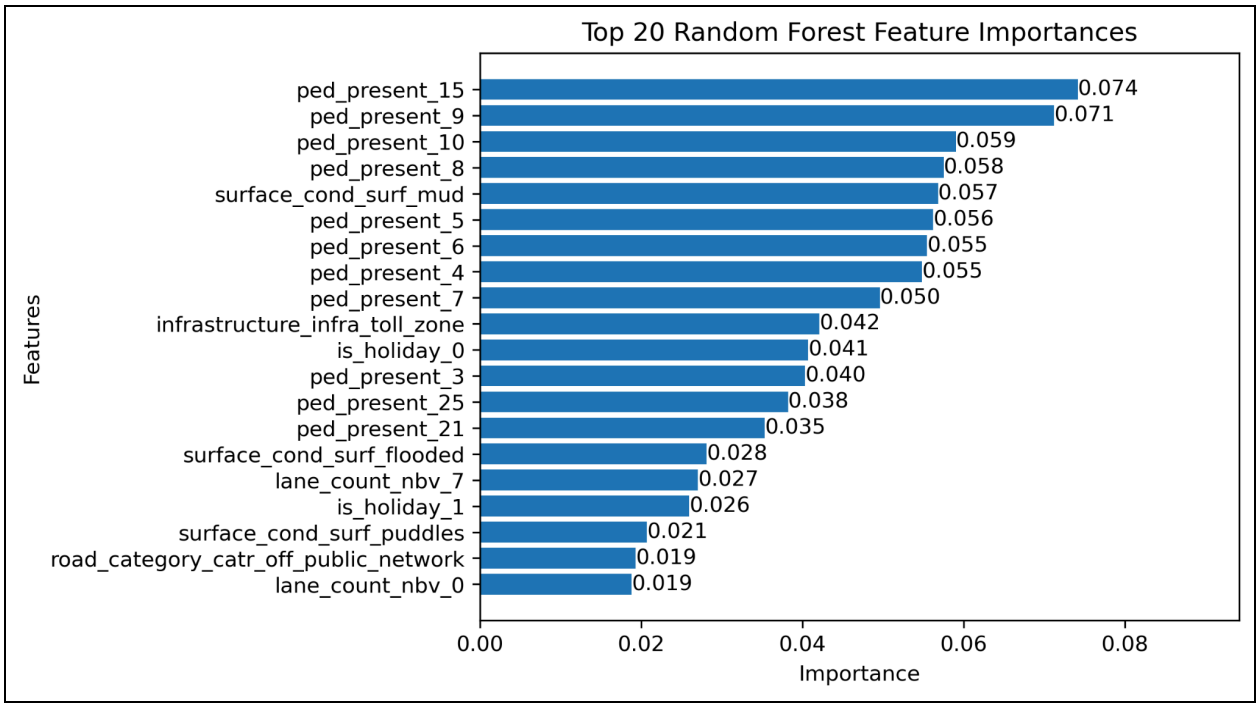




Feature Importances

The top 20 features were extracted from each model after training and prediction. The logistic regression and random forest have straightforward ways of extracting. For the neural network, a SHAP analysis was performed using the LIME python library.





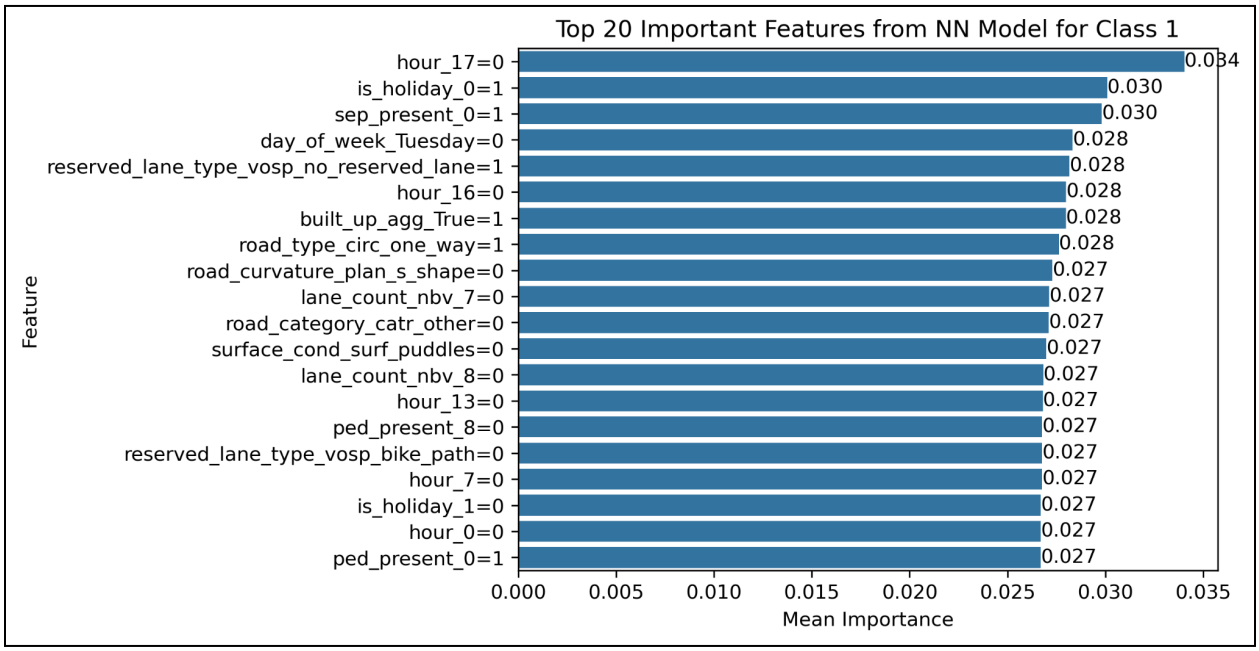


Table Format of top 20 feature importances per model:

	rank_NN	feature_NN	importance_NN	rank_LR	feature_LR	importance_LR	rank_RF	feature_RF	importance_RF
0	1	ped_present_0=1	0.0584	1	is_holiday_0	1.4567	1	ped_present_15	0.0742
1	2	ped_present_5=0	0.0531	2	reserved_lane_type_vosp_no_reserved_lane	1.2213	2	ped_present_9	0.0712
2	3	is_holiday_0=1	0.0530	3	ped_present_0	1.0441	3	ped_present_10	0.0590
3	4	reserved_lane_type_vosp_no_reserved_lane=1	0.0526	4	infrastructure_infra_none	0.9980	4	ped_present_8	0.0575
4	5	hour_6=0	0.0524	5	sep_present_1	0.9070	5	surface_cond_surf_mud	0.0568
5	6	infrastructure_infra_toll_zone=0	0.0518	6	is_weekend_0	0.7958	6	ped_present_5	0.0562
6	7	ped_present_10=0	0.0514	7	weather_atm_normal	0.7567	7	ped_present_6	0.0554
7	8	hour_22=0	0.0514	8	intersection_type_int_out_of_intersection	0.7547	8	ped_present_4	0.0548
8	9	hour_9=0	0.0513	9	road_curvature_plan_straight	0.7244	9	ped_present_7	0.0496
9	10	hour_15=0	0.0512	10	road_slope_prof_dish	0.7240	10	infrastructure_infra_toll_zone	0.0421
10	11	road_category_catr_off_public_network=0	0.0511	11	built_up_agg_True	0.6627	11	is_holiday_0	0.0407
11	12	built_up_agg_False=1	0.0509	12	surface_cond_surf_dry	0.6501	12	ped_present_3	0.0403
12	13	infrastructure_infra_pedestrian_area=0	0.0502	13	luminosity_lum_lum_full_day	0.5919	13	ped_present_25	0.0382
13	14	weather_atm_fog_smoke=0	0.0497	14	lane_count_nbv_2	0.5521	14	ped_present_21	0.0354
14	15	ped_present_6=0	0.0495	15	road_type_circ_two_way	0.4150	15	surface_cond_surf_flooded	0.0281
15	16	weather_atm_storm=0	0.0494	16	road_category_catr_municipal_road	0.3183	16	lane_count_nbv_7	0.0271
16	17	surface_cond_surf_ice=0	0.0490	17	built_up_agg_False	0.2780	17	is_holiday_1	0.0260
17	18	built_up_agg_True=1	0.0490	18	is_weekend_1	0.1488	18	surface_cond_surf_puddles	0.0207
18	19	hour_14=0	0.0489	19	surface_cond_surf_wet	0.1110	19	road_category_catr_off_public_network	0.0193
19	20	hour_20=0	0.0484	20	road_category_catr_departmental_road	0.0822	20	lane_count_nbv_0	0.0188

Feature Importance Notes:

- Top features for both the neural network and random forest models change every time the model is trained and predictions made.
- The logistic regression model top features seem consistent across new trainings

Final Model Recommendation

Logistic Regression Classifier

- default values as of scikit Learn version 1.5.2
- random_state=9 for reproducibility

This model is chosen as the final use model due mainly to lower resource usage and time for code execution when compared to random forest and the neural network.

Another very important factor is the feature coefficients are much more informative of how each feature affects the chance of a crash occurring versus the black box nature of a neural network whose top most important features change each time the model is trained.

Conclusion

- The most likely location/time for a crash according to the top features of the logistic regression model would be:
    - non-holiday
    - no reserved lane (ex. an emergency lane)
    - no pedestrians
    - no infrastructure in immediate area
    - a separator is present
    - weekday (Mon-Fri)
    - normal weather
    - away from an intersection
    - straight road
    - dish slope (ex. bottom of two hills)
    - in built-up areas (i.e. town or city)
    - dry roads (matches for normal weather)
    - full daylight
    - 2 lanes
    - two way road
    - municipal (local) road
    - Note: this is taking the top features that do not conflict, this does not necessarily confirm this specific combination has a 100% accident rate.
    -
- The goal of this project was to characterize (describe) the locations where vehicle crashes were most likely to occur.
- This was performed through building a binary classification model (Yes crash or No crash) and examining the importance of the characteristics for predicting the occurrence of a crash.
- A hypothesis set at the beginning of this project was that crashes would be more likely to occur for bad driving conditions (i.e. weather, more cars crossing paths, etc.).
    - This was proven false by the top features indicating non-inclement conditions
- This does not mean that adverse conditions do not carry risk, just that agencies may want to monitor these areas more than others in order to increase efficiency in which they respond to an emergency.

Possible Improvements

Possible Limiting factors & Improvements:

- Non-crash data not present
    - This could potentially be collected by automatic systems that count cars that trigger a detector. Matching data to the crashes dataset would also need to be

collected but except for weather and date/time those features should remain static and can be hard coded such that they are included when a car is counted.

- Using non-crash data would improve model quality in that an actual probability could be determined for a crash rather than simply treating a combination of features as Yes-crash or No-crash