Bike and Car Accident Machine Learning Report

Introduction and Motivation

The purpose of this project was to:

1. Determine if bike accidents predict car accidents, or vice versa
2. Determine other road attributes that predict accidents

The statistical analysis revealed that various road attributes did have a statistically significant effect on number of accidents.

In this document, I show the steps I took to create machine learning models to predict accidents, and which features had the most impact on predicting accidents. I used both regression and classification, and applied a wide variety of machine-learning techniques to find the most effective.

Data Re-Wrangling

The Data Wrangling performed earlier was sufficient for exploratory data analysis and visualization, but for machine learning, it needed to have missing values filled in, and to have classification variables turned into dummy columns.

First I did fill in missing values with -1, so there would be no non-null values but they would be easy to identify. For some values I was able to use contextual information to fill them in, like in places where there was no median, its width would be zero.

I used some linear regression to try to interpolate number of lanes from road width and vice versa, and also to fill in speed limits. For all other quantitative values, I filled in with the mean of the non-missing values.

For qualitative or categorical data, I left missing values as -1, and turned them into dummy columns.

For more detail, see this notebook.

Models for Regression

I calculated number and density of bike and car accidents in each region where they occurred. I hoped that I would be able to use regression to create a model to predict number of accidents based on road attributes. I used many different regression algorithms, but never produced a sufficient R^2 score, approaching 1.

Model	**Vehicle**	**R^2 Score**
LinearRegression	car	0.063
LinearRegression	bike	0.063
Ridge	car	0.063
Ridge	bike	0.078
Lasso	car	0.027
Lasso	bike	0.002
ElasticNet	car	0.027
ElasticNet	bike	0.018
KNeighborsRegressor	car	-0.007
KNeighborsRegressor	bike	0.265
SVR	car	-0.009
SVR	bike	-0.066

NuSVR	car	-0.011
NuSVR	bike	-0.113
DecisionTreeRegressor	car	-0.429
DecisionTreeRegressor	bike	0.033
RandomForestRegressor	car	0.109
RandomForestRegressor	bike	0.451
BaggingRegressor	car	0.025
BaggingRegressor	bike	0.025
GradientBoostingRegressor	car	0.099
GradientBoostingRegressor	bike	0.422
AdaBoostRegressor	car	-0.529
AdaBoostRegressor	bike	-0.157

The best these did was accounting for 45% of observed variance, which is not enough to make a very useful regressor. I did try scaling the data to see if that would help, but it did not.

However, it would still be useful to predict if an area of road is likely to have an accident in the future.

See these notebooks for more detail:

- Linear Regression
- K Nearest Neighbors
- SVM Regressors
- Tree and Ensemble Regressors

Models for Classification

Classification of roads that are likely to have accidents versus those that are not is still a useful project. Next I created a new response variable that was 0 if no accidents occured in the road area and 1 if it did. I used many different classification algorithms and found that some produced almost 90% accuracy.

Model	**Vehicle**	**Accuracy**
LogisticRegression	car	0.645
LogisticRegression	bike	0.474
KNeighborsClassifier	car	0.844
KNeighborsClassifier	bike	0.861
SVC	car	0.834
SVC	bike	0.788
LinearSVC	car	0.681
LinearSVC	bike	0.794
SGDClassifier	car	0.719
SGDClassifier	bike	0.762
DecisionTreeClassifier	car	0.886
DecisionTreeClassifier	bike	0.902
RandomForestClassifier	car	0.885
RandomForestClassifier	bike	0.908
BaggingClassifier	car	0.895
BaggingClassifier	bike	0.913
GradientBoostingClassifier	car	0.885
GradientBoostingClassifier	bike	0.895
AdaBoostClassifier	car	0.859
AdaBoostClassifier	bike	0.884

For each of these, I examined the confusion matrix as well, to make sure that the false positives and false negatives were fairly balanced, though for this problem, false positives are better than false negatives. The safety of a given road can always be improved.

For more detail see the following notebooks:

- Logistic Regression
- K Nearest Neighbors

- SVM Classifiers
- Tree and Ensemble Classifiers

Hyperparameter Tuning

Since the Decision Tree performed both fast and pretty well, and Random Forest, Bagging, and Gradient Boosting, all performed well, I decided to tune these.

Model	**Vehicle**	**Pre-Tuning Accuracy**	**Best Tuned Accuracy**
DecisionTreeClassifier	car	0.886	0.888
DecisionTreeClassifier	bike	0.902	0.905
RandomForestClassifier	car	0.885	0.897
RandomForestClassifier	bike	0.908	0.919
BaggingClassifier	car	0.895	0.895
BaggingClassifier	bike	0.913	0.922
GradientBoostingClassifier	car	0.885	0.894
GradientBoostingClassifier	bike	0.895	0.913

For more details on Hyperparameter Tuning, see this notebook.

The best performing model for predicting car accidents was a Random Forest Classifier. The confusion matrix for the best tuned model is shown here:

	Actual No Accident	**Actual Accident**
Predicted No Accident	11423	1060
Predicted Accident	1242	8512

The classification report for the car accident Random Forest Classifier:

	precision	**recall**	**f1-score**	**support**

No Accident	0.90	0.92	0.91	12483
Accident	0.89	0.87	0.88	9754
Accuracy			0.90	22237
Macro Avg	0.90	0.89	0.89	22237
Weighted avg	0.90	0.90	0.90	22237

The best model after tuning for predicting Bike Accidents was the Gradient Boosting Classifier. Here is the confusion matrix:

	Actual No Accident	**Actual Accident**
Predicted No Accident	17173	353
Predicted Accident	1586	3125

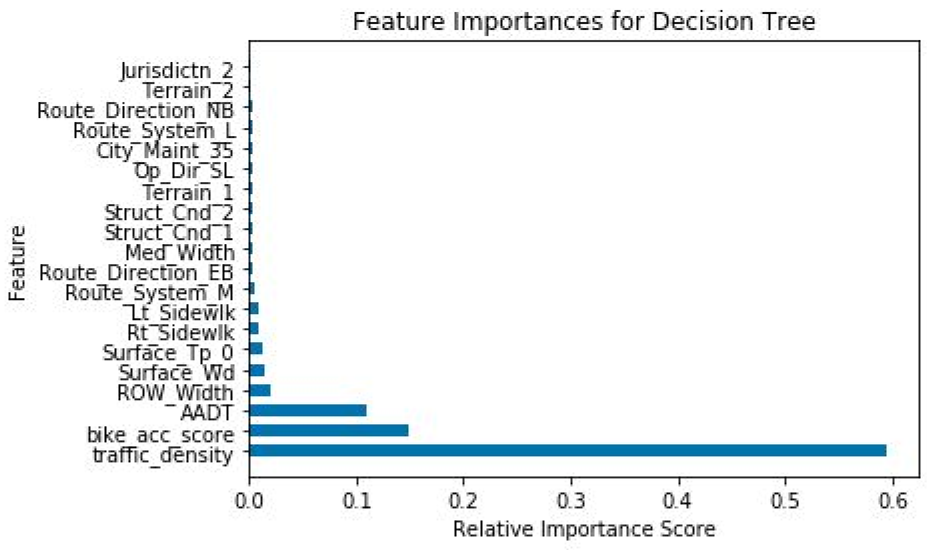
And here is the classification report. This has a low recall for accidents, and a high false-positive rate for predicting an accident. However, given that roads can always be made safer and an area with no accidents may be an area that simply hasn't had an accident yet, I think this may be a benefit to the model.

	precision	**recall**	**f1-score**	**support**
No Accident	0.92	0.98	0.95	17526
Accident	0.90	0.66	0.76	4711
Accuracy			0.91	22237
Macro Avg	0.91	0.82	0.85	22237
Weighted avg	0.91	0.91	0.91	22237

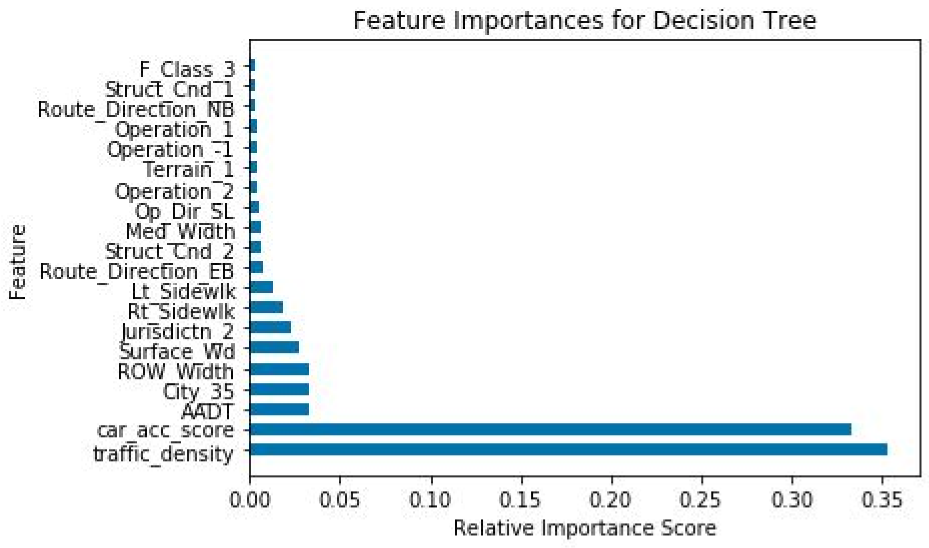
Feature Selection

I used the decision tree feature importance to identify the most important features for predicting car and bike accidents. For more information see this notebook.

Most important features for predicting car accidents:



Most important features for predicting bike accidents:



Some of the features are dummy columns for categorical features, so I kept not just the top categories, but all the dummy features. It is still a much reduced set. For cars, it went from 210 features to 77. For bike accidents it went from 210 to 54.

Below is a table comparing the accuracy of running these models on a reduced feature set. These are untuned accuracies.

Model	**Vehicle**	**Pre-Tuning Accuracy**	**Best Tuned Accuracy**	**Reduced Feature Accuracy**

				(untuned)
DecisionTreeClassifier	car	0.886	0.888	0.888
DecisionTreeClassifier	bike	0.902	0.905	0.810
RandomForestClassifier	car	0.885	0.897	0.892
RandomForestClassifier	bike	0.908	0.919	0.813
BaggingClassifier	car	0.895	0.895	0.893
BaggingClassifier	bike	0.913	0.922	0.812
GradientBoostingClassifier	car	0.885	0.894	0.883
GradientBoostingClassifier	bike	0.895	0.913	0.801

We can get almost the same accuracy with fewer features for predicting car accidents, but decreasing features reduces the accuracy for predicting bike accidents by quite a bit, from a best of 92% to 80%.

Conclusions and Next Steps

A Random Forest Classifier is best for predicting car accidents. A Gradient Boosting Classifier is best for predicting bike accidents.

The Decision Tree classifier performs so well on this data, and very quickly, it seems to make sense to use it to test road improvements. A Decision Tree is also very interpretable--at each branch one choice leads to more accidents, one to fewer, so it could easily be used to make real world decisions. One of my next steps will be to visualize simplified decision trees.

I also want to examine false positives and see what they have in common with true positives. Finally, I will fake some data with various features altered to see how it affects the model outcomes.

See the Exploring the Models section in the Bike and Car Accident Final report to see this exploration.