Toronto cycling accident rates – Predicting collision rates from intersection characteristics

Christian Farfan Centeno

INTRODUCTION

Engineers in the automobile industry have tried to design and build safer automobiles, and most North American cities are completely designed around the ubiqitous car. However, a growing number of individuals choose to make regular use of bicycles in major cities, despite the fact that hopsital statistics show that cyclists are the highest percentage of severly injured road compared to other groups (http://www.trafa.se/en/road-traffic/road-traffic-injuries/). In Toronto, for example, 1.7% of the people in Toronto rode their bikes to work in 2006, according to the Cenus. This share of regular users is under reported however, as it does not include those who bike for fitness, to run errands, or for leisure. It also may exclude those who use a bike to reach public transportation for their commute, as the Census only allowed one choice for travel method to work. Thus the number of active cyclists in the city is higher than 1.7% of the population. From 2001-2006 there was over 30% increase in the number of cyclists commuting to work (http://goo.gl/sMLsfs) and in 2011 the proportion was reported higher again, despite mode of commuting being removed from the mandatory part of the Census, and becoming voluntary information. This change in transportation habits cannot be ignored, and transportation offices need to consider them when making future decisionsm, particularily as there isn’t much inherent safety that can be added to bike designs when cyclists share the road directly with motor vehicles. Instead, cities have to rethink how they design new infracsture that better accomodates this growing class of road user.

The city of Toronto has developed a Ten Year Plan to develop the area's cycling network and become a more bicycle friendly city. Meant to outline the future investments in infrastructure between 2016-2025, the city has published maps of the proposed expanded cycling network. These maps were produced mostly by data collection on traffic, estimating potential demand by non-cycling trips and non-walking trips less than 5km (http://www.torontocyclingnetwork.info/studying-toronto/). Identifying where to improve or add bike lanes via traffic is only partly accomplished by looking at accicident numbers and bike traffic though. A more instructive approach would be identifying why cycling collisions happen, or at least what factors increase the liklihood . Identifying which areas or intersections require intervention or perhaps should be avoided by cyclists altogether can be accomplished by analyzing cycling collision data and applying machine learning methods to make predictions on accident rates. This should allow us to predict collision rates for intersections throughout Toronto based on a few chosen features, and predict how changes to infrastructure or traffic rates may affect the risks Cyclists face in the city.

This paper tackles this problem by using Toronto cycling accidents to perform a regression analysis on intersection accident rates. Evidence suggests that around 75% of cycling accidents occur at, or near, a road junction (http://www.trafa.se/en/road-traffic/road-traffic-injuries), so focusing on intersections rather than stretches of individual roads makes sense from a domain-specific view.

The data contains the date/time, severity of injury, as well as GPS and street intersection locations of reported cycling collisions, along with other relevant data. Grouping by equivalent intersections, each junction becomes a data point with a target (the number of yearly collisions), and associated features. Various regression approaches can then by run on the data, of which OLE regression, OLE with L1 and L2 regularization (Lasso and Ridge), and random forest are all presented in this report.

The best performing model was the ensemble random forest method, which on average managed to explain 62% of the variation in collision rates at intersections located in Central Toronto.

DATA

The main data used is a Bicycle collision data set released by the Toronto Traffic Safety Unit a few years ago (https://github.com/farfan92/SpringBoard-/blob/master/cycling_collisions_toronto_1986-2010.xls.csv). It contains 31480 cycling accidents in a time period from 196-2010.

The secondary data are the vehicular, bicycle and pedestrian counts from the Toronto Open Data website: http://www1.toronto.ca/wps/portal/contentonly?vgnextoid=1a66e03bb8d1e310VgnVCM10000071d60f89RCRD

Complementary data was the centreline database of all intersections in the City, also found on the Open Data webstite: http://www1.toronto.ca/wps/portal/contentonly?vgnextoid=1a66e03bb8d1e310VgnVCM10000071d60f89RCRD

DATA WRANGLING

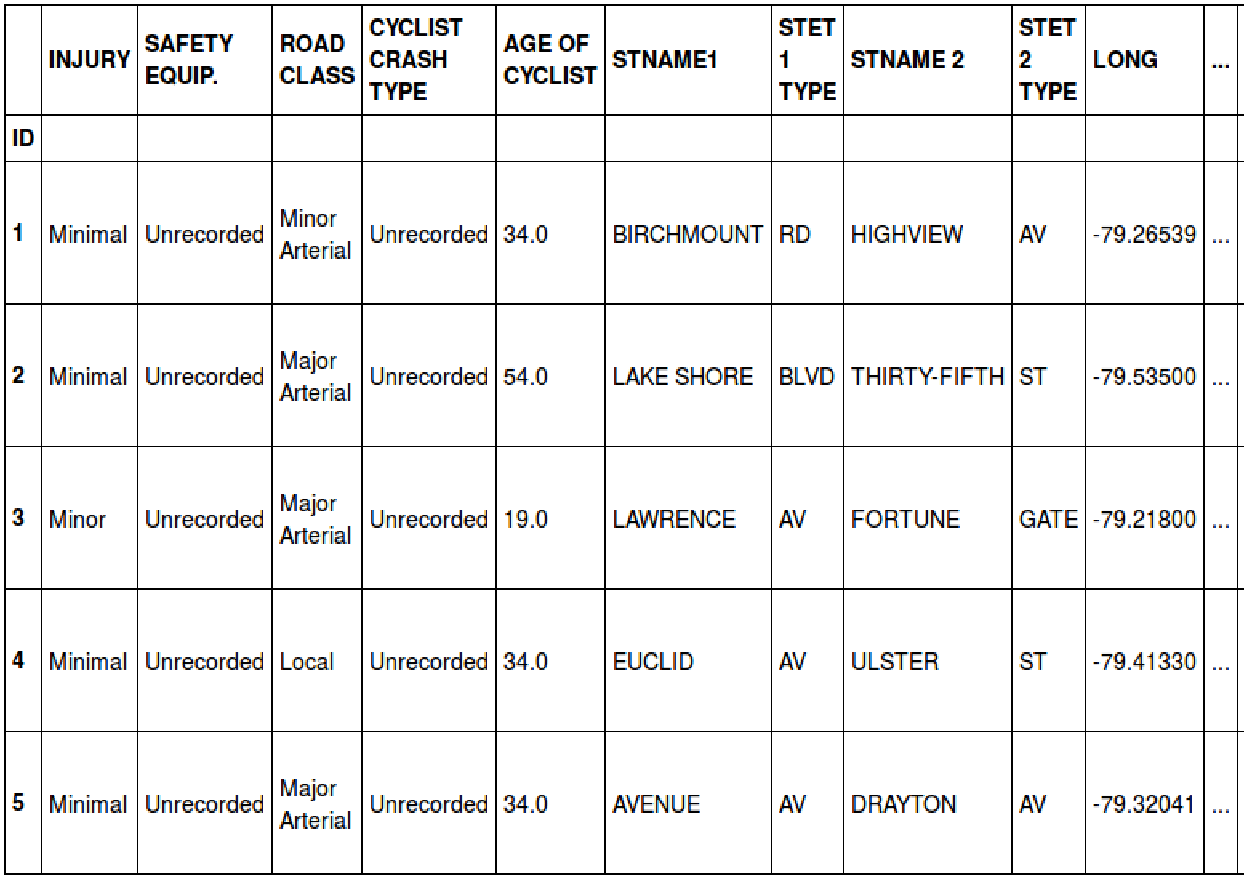
We chose to use the 2011 vehicular count data, as this would be most relevant to our accident data. The file is in .csv format and each contains column entries for a street 1, street 2, latitude, longitude, as well as both eight and twenty-four hour vehicular and pedestrian counts.

The bike counts are contained in a zip file, containing spreadsheets for various intersections. The format for these varies greatly, with many having multiple tabs, some divided by date, while others by direction through intersection. The different formats and small amount of files (~50), meant that this information could be scrapped manually, and put into a new spread sheet with the intersections and corresponding counts. When possible, the weekday average of the counts was used otherwise, simply the single reported number was taken as the count.

The main dataset contains cycling accidents recorded by the police from 1986-2010. Each entry contains columns for the resulty injury severity, road class, streets, coordinates, date, and various other details.

Cleaning the main data was done by loading up the file in a spreadsheet program, and ordering entries by the various columns. It was quickly made apparent that a large portion of the unacceptable entries (strings in numerical columns and vice versa), occured due to an entry “skipping” a column. These entries could simply be manually copy and pasted over into the correct format. However, there were also “mixed” entries, where values for dates and coordinates were mixed with strings for driver actions’ or other labels, in the same column. These seem to be an artifact of errors from an initial data output from the Police’s database into .csv format. These cannot be adjusted as easily. For these cases, wildcard regular expressions were used to filter out the correct numerical values in the columns.

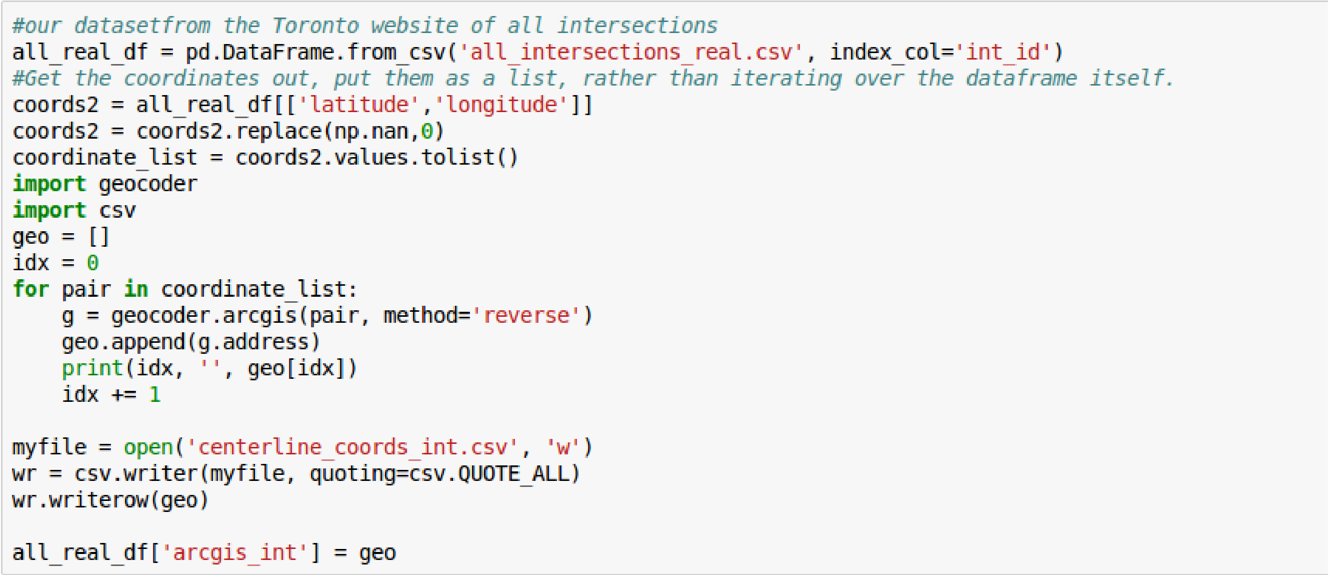
We then read this dataset into a Jupyter notebook through Pandas, and droped alternate format columns, resulting in a dataframe as shown in Table 1. In order to properly analyse the collision rates at various intersections, the data set needed to be geocoded to specific intersections. The collisions in the dataframe were all first geocoded based on the cross on the crosstreets listed, rather than the GPS coordinates, since there were many coordinates that just were defaulted to the center of Toronto. After this first pass, there were still a few "impossible" intersections remaining after examing the output. These were then geocoded using their coordinates in a second pass. The geocode API used was that from ArcGis.



	INJURY	SAFETY EQUIP.	ROAD CLASS	CYCLIST CRASH TYPE	AGE OF CYCLIST	STNAME1	STET 1 TYPE	STNAME 2	STET 2 TYPE	LONG	...
ID											
1	Minimal	Unrecorded	Minor Arterial	Unrecorded	34.0	BIRCHMOUNT	RD	HIGHVIEW	AV	-79.26539	...
2	Minimal	Unrecorded	Major Arterial	Unrecorded	54.0	LAKE SHORE	BLVD	THIRTY-FIFTH	ST	-79.53500	...
3	Minor	Unrecorded	Major Arterial	Unrecorded	19.0	LAWRENCE	AV	FORTUNE	GATE	-79.21800	...
4	Minimal	Unrecorded	Local	Unrecorded	34.0	EUCLID	AV	ULSTER	ST	-79.41330	...
5	Minimal	Unrecorded	Major Arterial	Unrecorded	34.0	AVENUE	AV	DRAYTON	AV	-79.32041	...

Table 1: Cleaned dataframe of Toronto cycling collisions from 1986-2010

The dataset containing all the intersections in Toronto was filtered out by selecting only those which were not classified as “pseudo” intersections (overpasses and underpasses), and highway intersections where cyclists would not be found. The complementary dataset of all intersections in Toronto was also geocoded, using the latitude and longitude coordinates. These also required a second pass using a different geocoding API (Google Maps) on about 10 or so entries, as ArcGis would not place these and return only NaN values. This geocoding was done to later on cross reference with the collisions database by joining the two and then dropping all the duplicates, so as to include road junctions that never had a collision between 1986 and 2010.



```
#our datasetfrom the Toronto website of all intersections
all_real_df = pd.DataFrame.from_csv('all_intersections_real.csv', index_col='int_id')
#Get the coordinates out, put them as a list, rather than iterating over the dataframe itself.
coords2 = all_real_df[['latitude','longitude']]
coords2 = coords2.replace(np.nan,0)
coordinate_list = coords2.values.tolist()
import geocoder
import csv
geo = []
idx = 0
for pair in coordinate_list:
    g = geocoder.arcgis(pair, method='reverse')
    geo.append(g.address)
    print(idx, '', geo[idx])
    idx += 1

myfile = open('centerline_coords_int.csv', 'w')
wr = csv.writer(myfile, quoting=csv.QUOTE_ALL)
wr.writerow(geo)

all_real_df['arcgis_int'] = geo
```

Figure 1: Code to geocode intersection entries after filtering for real intersections only.

Figure 1 is a code example of how the geocoding was done. It shows how the coordinates for the entire list of intersections in the city was geocoded from the given latitude and longitude.

EXPLORATORY DATA ANALYSIS

Preliminary investigation invovled working only with the collisions dataframe after cleaning. First we look at some of the raw numbers of collisions throughout the years shown in Figure 2, grouped by reported collision type. We note that there was a steady decline in numbers until 1994, when a substantional drop occurred until 1997 when it picked up again. I have been unable to identify any particular reason why this would be the case, as there weren't any major traffic affecting news I could find. After that the numbers pick back up and hold generally steady.

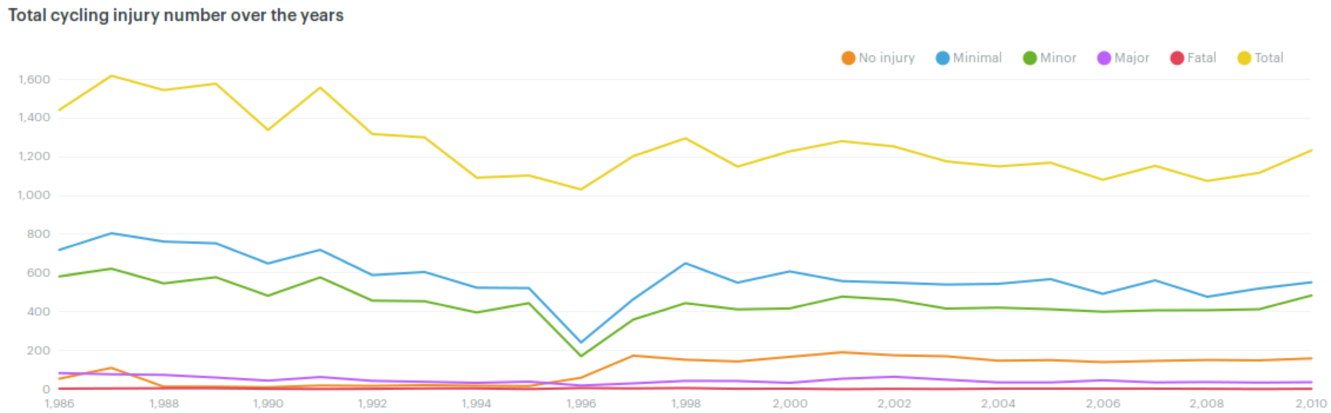


Figure 2: Yearly injuries grouped by injury type

Looking at Figure 2, we note that the majority of injuries are reported as Minmal or Minor. Interestingly, the incident of collisions which result in a ‘no injury’ has actually increased significantly around the time we see a large dip in minmal and minor collisions. We suppose then that around this time there was a general shift in how Police were classifying injuries. This seems more likely than a mass change in cycling behavior, as we see that the total numbers themseles didn't change much. Only the proportions that were classified under 'No injury'. However in 1996 there were two cycling fatalities that attracted a lot of public attention. A press conference was held and the Toronto mayor accounced a cooperative effort would be put forward to examine safety issues for cyclists in the city. This seems to mark the beginning of the various cycling movements and safety studies within the city that now are reported on every few years. It is around this time frame that legisltion mandating that minors use helmets came into existence for the province of Ontario.

Therefore, we use 1996 as a benchmark to differentiate the two eras. The obvious change in the pattern of reports and known domain knowledge makes this a good cutoff for data fed into a machine learning model. Data older than this point represents a rather different time for cycling in Toronto, and would likely only hurt any practical applciations for the project.

Continuing our exploratory work, we would like to look at what the raw data tells us about the relative danger to cyclists at each intersection. Merely looking at the raw totals at each intersection would not be very intructive however, as different bicycle traffic levels would invalidate direct comparison. Therefore, we require a normalisation for each intersection in our dataset. Since attempts to gain access to more extensive data from the Toronto Cycling app yielded no results, we were required to make due with the limited dataset of bike counts which were previously discussed. The resulting files only spanned a few dozen points in Toronto, and more importantly, focused on the Downtown area for the most part. Figure 3 shows the locations for which bike traffic counts are available.

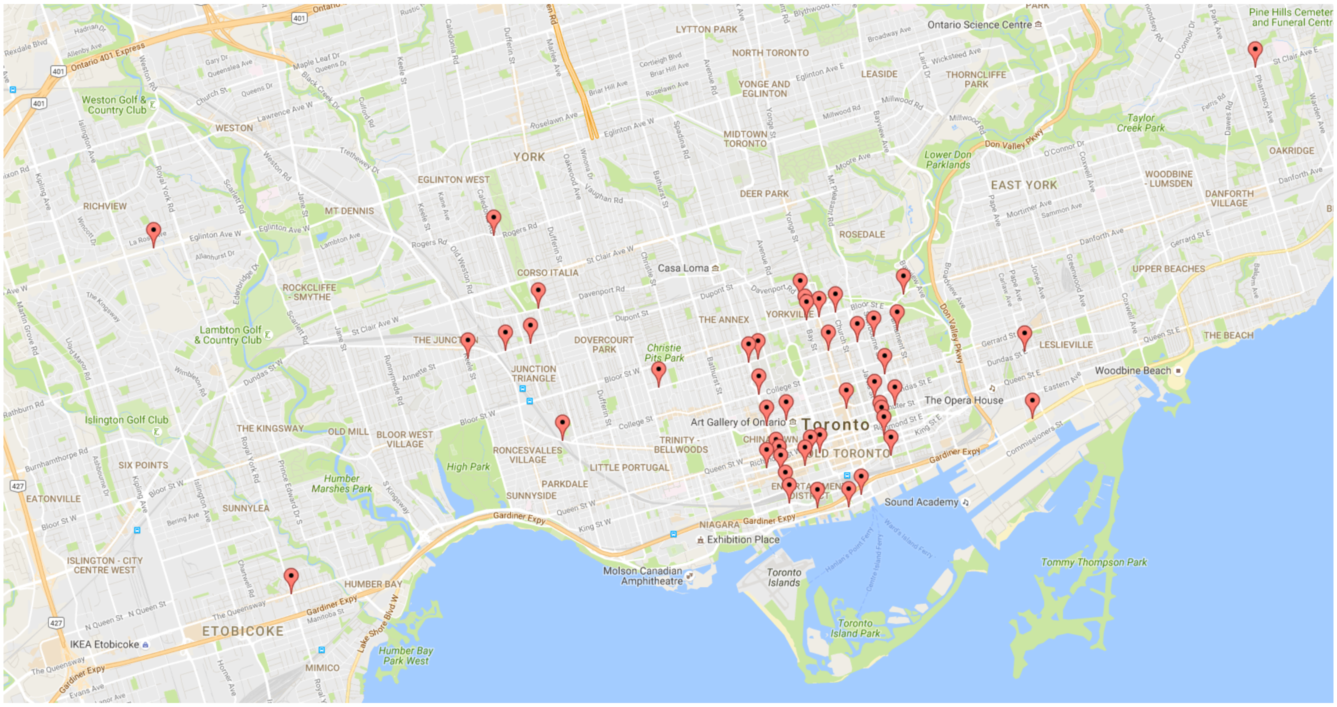


Figure 3: Intersections with recorded bike counts on Toronto Open Data site

In order to create a normalisation procedure that could be propogated to the rest of the our intersections, we compared these bike counts to the corresponding vehicle and pedestrian traffic counts. These are far more wide ranging, and by looking for some sort of relationship, one could make predictions on bike traffic in the missing points.

While there was unfortunately no obvious direct relationship between traffic numbers (something like a constant ratio, or a linearly increasing one), we do get an acceptable power fit when comparing the bike-to-pedestrian traffic ratio, once we filter out a few areas. The University of Toronto campus has a much higher bike-to-pedestrian ratio than any other available bike count locations. This isn't suprising, as students favour cycling to and from campus. The other area is the waterfront right by the financial and entertainment districts. These points had a far lower traffic ratio than given by this fit. So we proceed by treating them as seperate special cases, mapping the available bike counts directly to any nearby collisions, as long as they fit within a 'box' that we'll use to define those areas. The power-fit made on the resulting “filtered” bike intersections is shown in Figure 4.

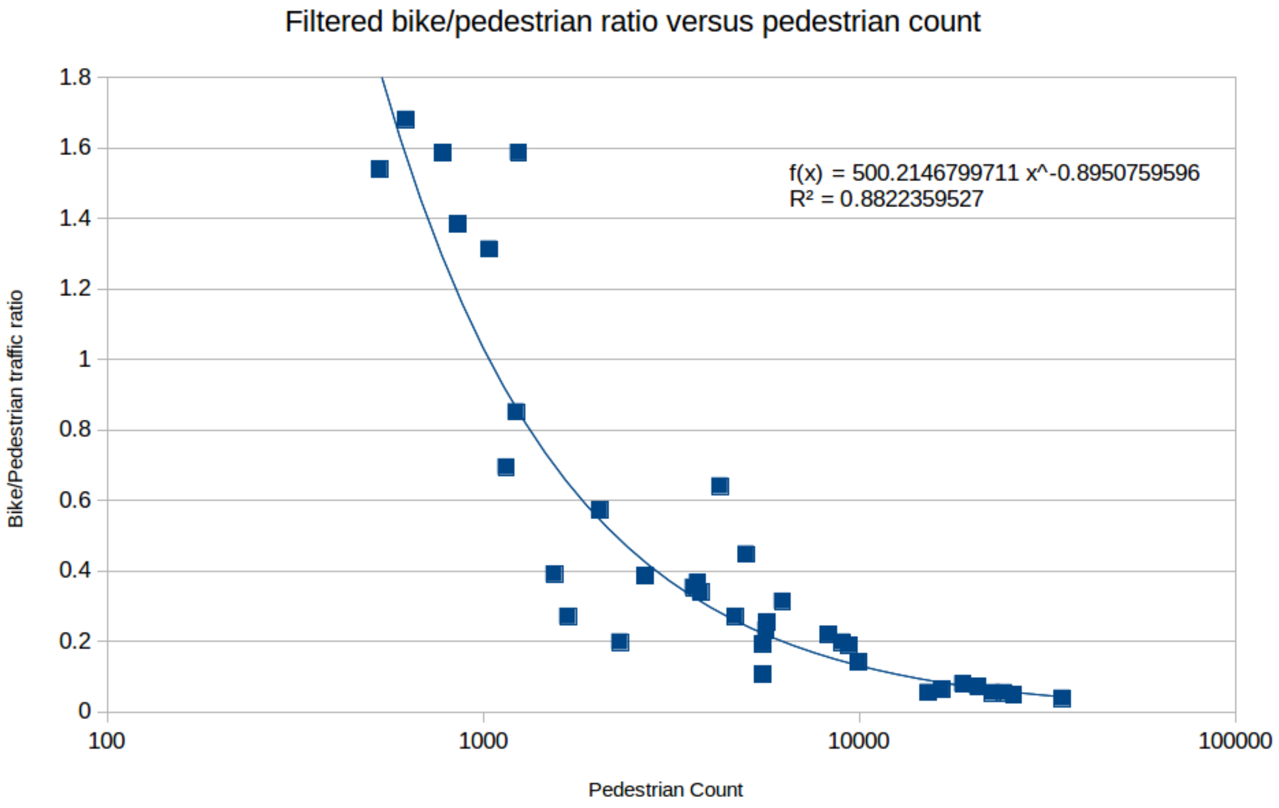


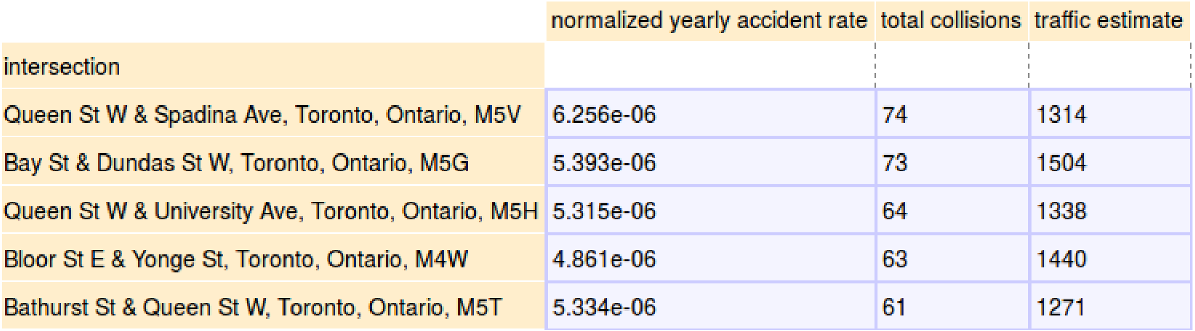
Figure 4: Traffic count ratio excluding special case areas

We note that the resulting fit gives a rather good R-squared value of 0.88. However, two issues give us pause. The first is that the fit is made with a small data set. Secondly, the points are all centraly located, mostly downtown. Our confidence in this relationship holding up in the outer city is not strong, as cycling and pedestrian behavior is markedly different in any city when moving out towards the suburbs. Thus we restrict our geoceded dataframe of collisions to those that have a postal code associated with the areas roughly east of the Junction, West of carlaw, and south of Dupont, with the neigbourhoods near Exhibition place also thrown out. This isn't a perfect system by any means, but we have no reason to believe that our nice power-fit relationship will hold outside this rough area.

Once we have restricted our collisions, this reduces our data to 10722 recorded cycling accidents, from the 31480 orginally in the data set. We map the coordinates of each of these entries to the closest intersection in our dataset of pedestrian counts. We use the associated pedestrian count, along with the power fit function from Figure 4 to then add a column of bike traffic estimates to the collisions dataframe. Rememebering that we have a few areas that we're treating apart, we need boxes so we can check if points fall inside those areas. Using the Path calss from matplotlib, polygons representing the exception areas previously discussed were created. When mapping the collisions to the closest pedestrian count, first we check whether the point falls into one of these exception areas. If it does, we map the point to the closest bike count contained in the polygon, instead of the generalized bike traffic estimate derived from the power fit. The same process is repeated for the entire intersection dataframe.

Having derived traffic numbers for both our collision and intersection dataframes, we then join the two dataframes, using an outer join on the intersection name columns. The dataframe was then grouped by intersection, and a normalized accident rate was calculated using the bike traffic numbers, as well as the total collisions at that road junction. After sorting by this normalized accident rate, the duplicate intersections which were already contained in the collisions dataframe originally (the non-zeros), were dropped.

We thus have a dataframe of intersections sorted by the traffic normalized yearly accident rate. The first 5 entries, or most ‘dangerous’ intersections, are displayed in Table 2. We would like to account, however, for the fact that collisions are very rare events. Without a very long period of observation time, with many chances for each intersection to produce collisions, it is unwise to assume our sample distribution properly reflects the actual probability of collisions. The noise in the observation means that both uncharacterisatically high and low traffic incidences will skew our results, as traffic accidents are actually quite rare events, even over 25 years of data. We will deal with this using Bayesian statistics. We create a prior estimate on what the distribution actually is, using the samples that we have available. In a sense, we're using the data twice. First to estimate our prior. Then we'll use it along with this prior to create a posterior distribution. This is what is known as empirical Bayesian analysis.



	normalized yearly accident rate	total collisions	traffic estimate
intersection			
Queen St W & Spadina Ave, Toronto, Ontario, M5V	6.256e-06	74	1314
Bay St & Dundas St W, Toronto, Ontario, M5G	5.393e-06	73	1504
Queen St W & University Ave, Toronto, Ontario, M5H	5.315e-06	64	1338
Bloor St E & Yonge St, Toronto, Ontario, M4W	4.861e-06	63	1440
Bathurst St & Queen St W, Toronto, Ontario, M5T	5.334e-06	61	1271

Table 2: Top 5 interesections for cyclist accident rate in central Toronto

There is a lot of discussion revolving around full Bayesian analysis versus empirical Bayesian analysis. In some sense, if you know enough about the system you are studying, you should be able to develop a good prior beforehand, rather than using your limited dataset to do so. On the otherhand, imposing beliefs about your system beforehand can be equally controversial to depending on the data twice. A histogram of the collisions at intersections, with integer bin widths is shown in Figure 5, with a gamma fit overlayed from the Seaborn plotting API. Seeing this plot, it looks like a gamma distribution would be a good choice to fit our prior. We know that we are dealing with sums of poission distributions where each intersection is basically a poisson process, with the mean accidents for a year being the parameter. So the histogram in Figure 5 is really summing up the number of accidents each year, over these poisson processes. While we could try and fit each intersection individually, this is a whole lot of work. And it's not clear how one would use these estimated parameters to compute the prior distribution for the ensemble.

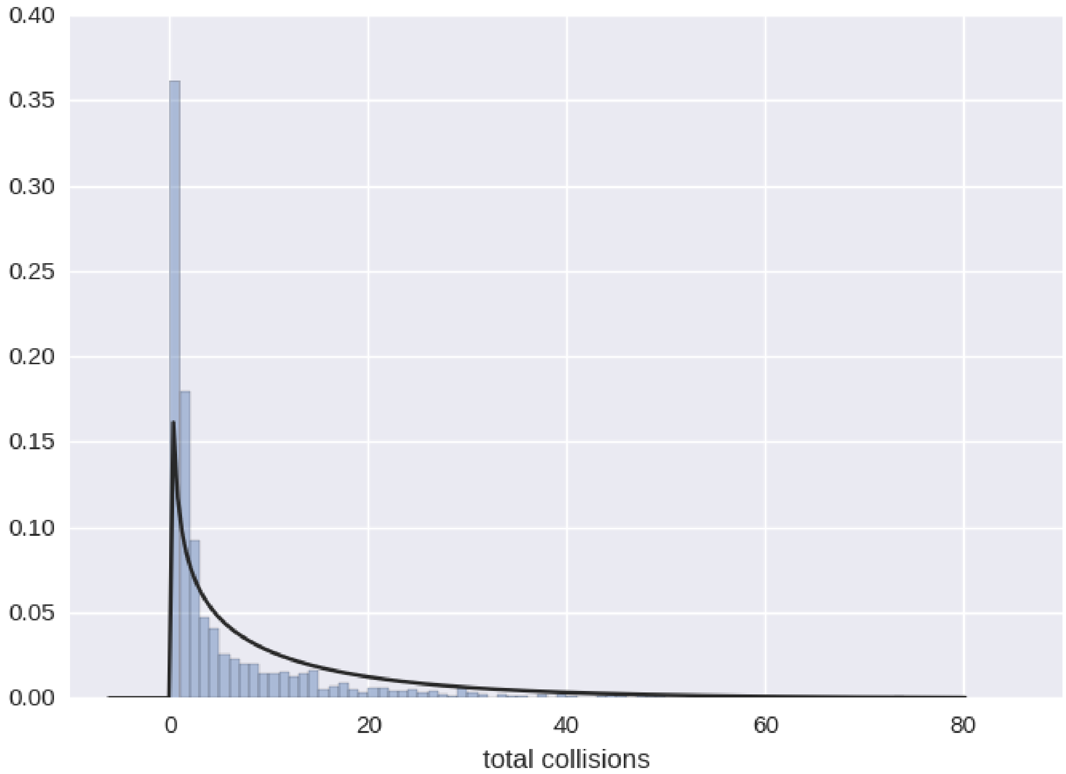


Figure 5: Histogram of total collisions for central Toronto intersections. Gamma distribution overlaid.

A gamma distribution looks nice, and intuitively makes sense since the prior for a poisson distribution is a gamma. There has been some very involved work on Bayesian inference for traffic problems. An overview of 3 models which extend our basic method is found in (Miranda-Mereno et al 2005): http://www.civil.uwaterloo.ca/itss/papers%5C2005-2%20(Alternative%20risk%20models%20for%20ranking%20sites).pdf, without directly touching on machine learning regression methods. We won't go that deep for our exploratory work. Suffice to say, there are extensions to the above Poisson model, which attempt to account for the heterogenity of various intersections. The same features one would choose to use in a linear regression for instance, would show up as parameters in the exponential distribution one tries to fit, and the poisson parameters for each intersection is assumed to be a random variable, drawn from a chosen distribution. Methods such as Gauss-Hermite quadrature for a Poisson lognormal model, and Newton–Raphson algorithm or expectation maximization for negative binomial regression are used to derive the fit parameters.

We proceed with our more naive approach, and fit a gamma distribution using Scipy’s stats module. The resulting fit is plotted over a zoomed in view of the same interger width histogram of total collisions in Figure 6. It’s a good looking fit. The histogram shows the normalized counts of integer collision totals for our merged dataframe. We then take the fit paramters of our model, and use them to adjust our estimate of the accident rate.

The mathematics of why the calculation to update our accident rate estimate is straight forward to derive from Bayesian probablity theory, but intuitively we just look at it like a weighted average of the prior and sample estimates, where more weight is given to the sample data when the time period is larger, or when the traffic numbers are greater. This makes sense, as in those cases we are more confident in our data providing us with a good idea of probability distibution for that intersection.

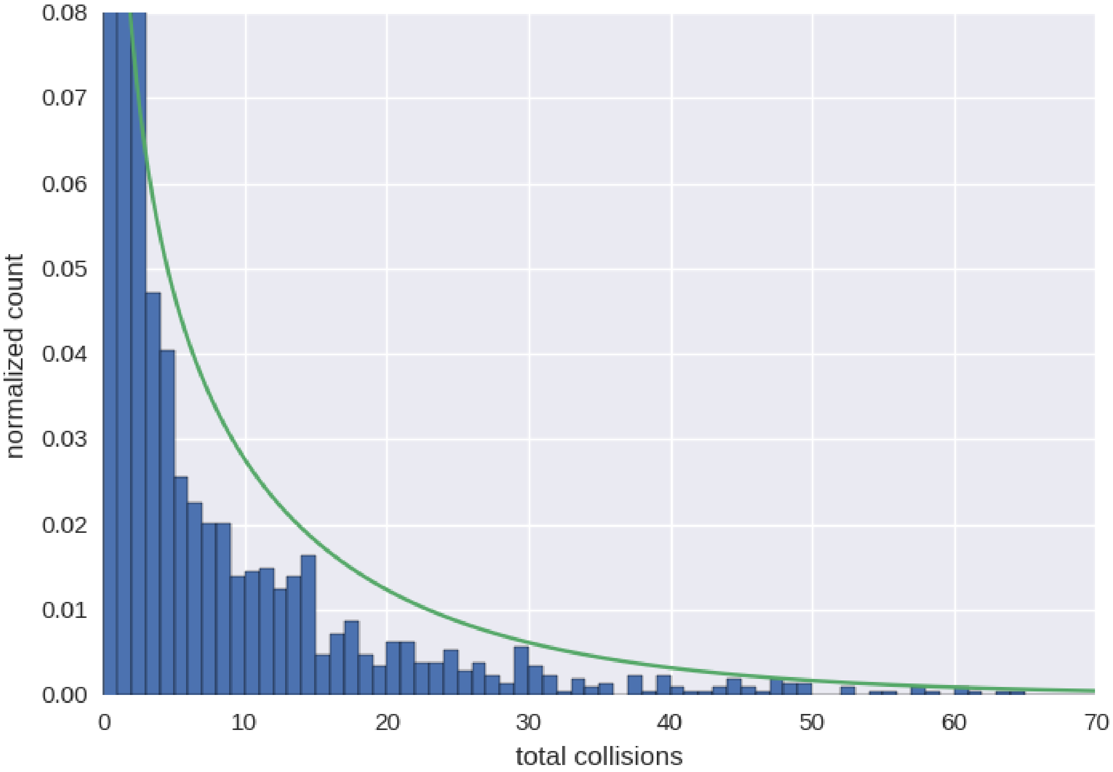
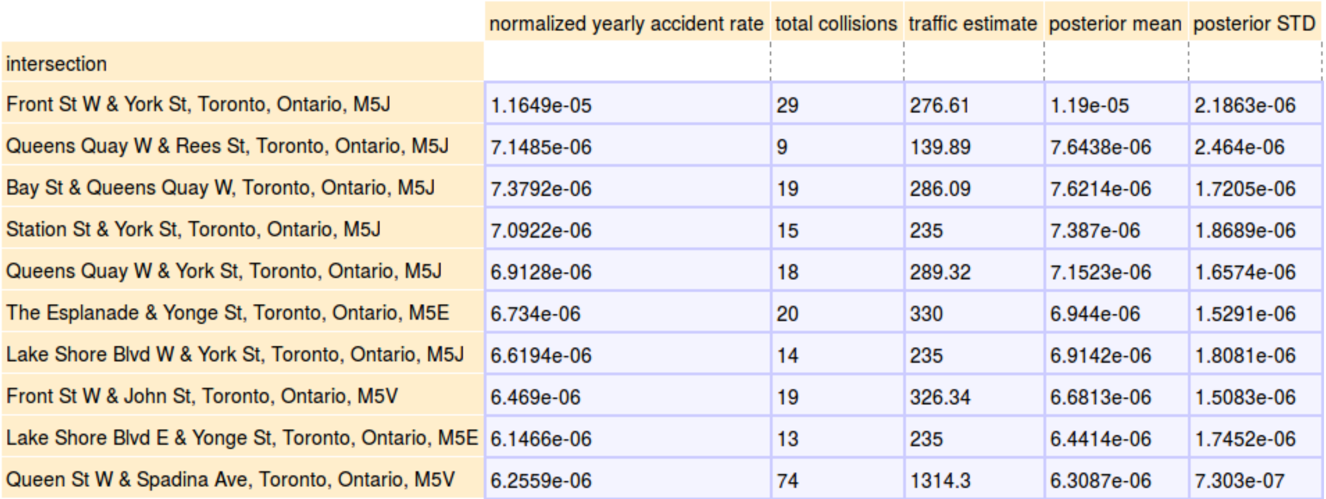


Figure 6: Gamma distribution fit to cycling collisions data for central Toronto

Table 3 shows the ten highest accident rate intersectinos for cyclists as given by the posterior distribution. The standard error and previous columns have been included as well. We see that our previous highest accident rate junction, at Queen and Spadina, is now at tenth, with much lower total collision intersections ahead now. If one looks deeper throughout the dataset we see that much of the order changes slightly, but nothing very notable. Since we're using so many years of data, the "prior" dominates. The biggest changes are for those intersections that have a very low traffic count, comparatively. Those with high traffic counts have smaller changes.



	normalized yearly accident rate	total collisions	traffic estimate	posterior mean	posterior STD
intersection					
Front St W & York St, Toronto, Ontario, M5J	1.1649e-05	29	276.61	1.19e-05	2.1863e-06
Queens Quay W & Rees St, Toronto, Ontario, M5J	7.1485e-06	9	139.89	7.6438e-06	2.464e-06
Bay St & Queens Quay W, Toronto, Ontario, M5J	7.3792e-06	19	286.09	7.6214e-06	1.7205e-06
Station St & York St, Toronto, Ontario, M5J	7.0922e-06	15	235	7.387e-06	1.8689e-06
Queens Quay W & York St, Toronto, Ontario, M5J	6.9128e-06	18	289.32	7.1523e-06	1.6574e-06
The Esplanade & Yonge St, Toronto, Ontario, M5E	6.734e-06	20	330	6.944e-06	1.5291e-06
Lake Shore Blvd W & York St, Toronto, Ontario, M5J	6.6194e-06	14	235	6.9142e-06	1.8081e-06
Front St W & John St, Toronto, Ontario, M5V	6.469e-06	19	326.34	6.6813e-06	1.5083e-06
Lake Shore Blvd E & Yonge St, Toronto, Ontario, M5E	6.1466e-06	13	235	6.4414e-06	1.7452e-06
Queen St W & Spadina Ave, Toronto, Ontario, M5V	6.2559e-06	74	1314.3	6.3087e-06	7.303e-07

Table 3: Highest accident rate intersections, ordered by posterior estimate.

Thus we have a Bayesian estimate on what the collision rates were. These can be viewed as a more trustworthy estimate of the future probality of a cycling collision.

REGRESSION ANALYSIS

As previously mentioned, using all 25 years of data does not make sense from a predictive analytics standpoint. If the resulting model is to make predictions for the current time period, then surely data closer to this time frame is more applicable. We saw in our exploratory data analysis that there was a total decline in accidents year by year up until 1996. There was a sharp decline in minimal and minor injuries that year, with the slack partly picked up by more "No injuries". Whether this was due to a change in how accidents were classified, or due to the legislation mandating minors to use helmets or both, is hard to say. It seems clear that predictive power would improve if we limited the data set to 1998 onwards. This would roughly half our number of starting collisions, but those remaining may be more indicative of current cycling trends. It also avoids the obvious decreasing trend and sudden jump in 96-97.

We use two dataframes of collisions from 1998 onwards. One includes all of those intersections in our data, while the other is restricted to the same central postal codes we used in our exploratory analysis. Some further data manipulation was required to properly encode features for use with Python's sci-kit learn machine learning library. The first step was to consolidate road class types, so that very rare classes such as 'major arterial ramp', were combined with 'major arterial', and that the intersection types for the entries which had zero collisions (from the intersection data set), could be mapped to appropriate road types. This same logic was extended to the traffic control of the junction. The second step was to select the most common of these categories for each feature, as many intersections had conflicting classifictions in the original Toronto Police dataset. When grouping by intersection, we then have 5 total features: bike traffic estimate, pedestrian traffic, vehicle traffic, road class type, and traffic control type.

We note that for linear regression analysis, the categorical variables of road class type and traffic control were converted into dummy variables as required by the linear regression algorithms in sci-kit learn. Regression was performed with the yearly collision rate (total collisions divided by 13 years) as the target variable. The results of the regression analysis are presented in Table 4. Rather than splitting the data manually, or coding up a cross validation method ourselves (as was required in the linear regression exercise for the course), we used the included cross_val_score and cross_val_predict methods from sci-kit learn to perform 5 fold cross validated predictions for each case.

Regression Algorithm	Intersections Included	Features	R-squared
Ordinary Least Squares	All	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.508928284915
Ordinary Least Squares	Central postal codes	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.60266120383
Lasso	All	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.133455369252
Ridge	All	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.519202385901
Ridge	Central postal codes	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.620701229935
Random Forest	All	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.533043163696
Random Forest	Central postal codes	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.612728419243

Table 4: Cross validated results of regression analysis on full and central only collisions

We first note that the score for ordinary least squares regression isn't terribly good. When plotted against the actual data, very obvious to us that there is a very large tendency to understimate collision numbers, particularily for mid, to high risk intersections. This is seen in Figure 7 below, for the central postal codes data. It plots the predicted collision rate against the actual measured 13 year rates. Keeping to the black central line indicates better predictive power.

We also see that restricting ourselves to intersections which are in central Toronto gives a pretty sizable improvement in R-squared, from around 0.51 to about 0.6. The residuals show an improvement in the underestimation as well, as shown in Figure 8.

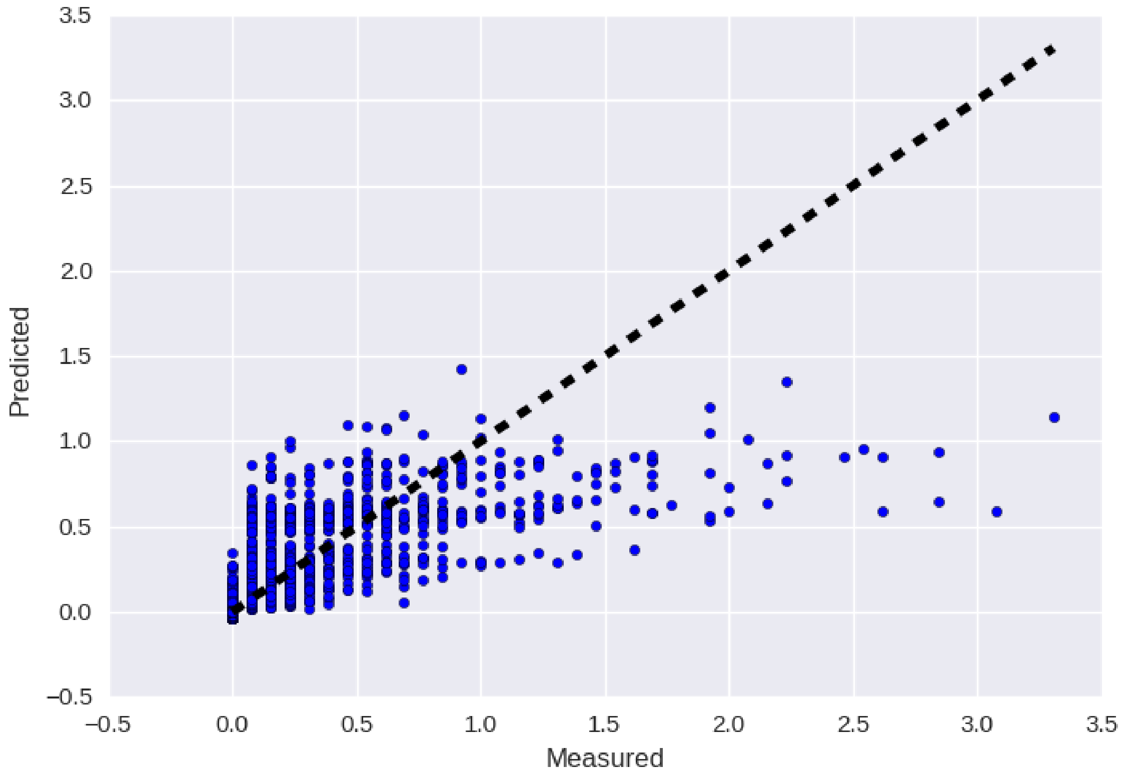


Figure 7: OLS regression of yearly collision rate for central intersections only. R-squared of 0.60266120383

The performance of Lasso regression performed extremely poorly. Lasso regression yields a sparse model. It is useful for reducing the dimensionality of a problem, but in this case, we actually had very few features to work with already. We used the LassoCV method to fit an alpha, and then mad our fit from there. The poor performance was even more prone to under estimation, and was significantly worse than ordginary least squares regression with an R-squared of 0.133455369252. It didn’t make sense to continue with this model by checking the performance with the central-only data, as this isn't the kind of problem that L1 regularization is meant for. Therefore, Table 4 has no entry for LASSO regression performed only on the central data. Moving on to ridge regression (linear least squares with L2 regularization) we see a small improvement in performance, with the central data set scoring above 62 percent. A cross validated random forest of 200 estimators performs similarly to ridge regression as well, scoring better for the entire dataset, but slightly worse for central only. We note that repeated runs of the random forest will give different R-square values, and the values in the table are merely averages over a few runs.

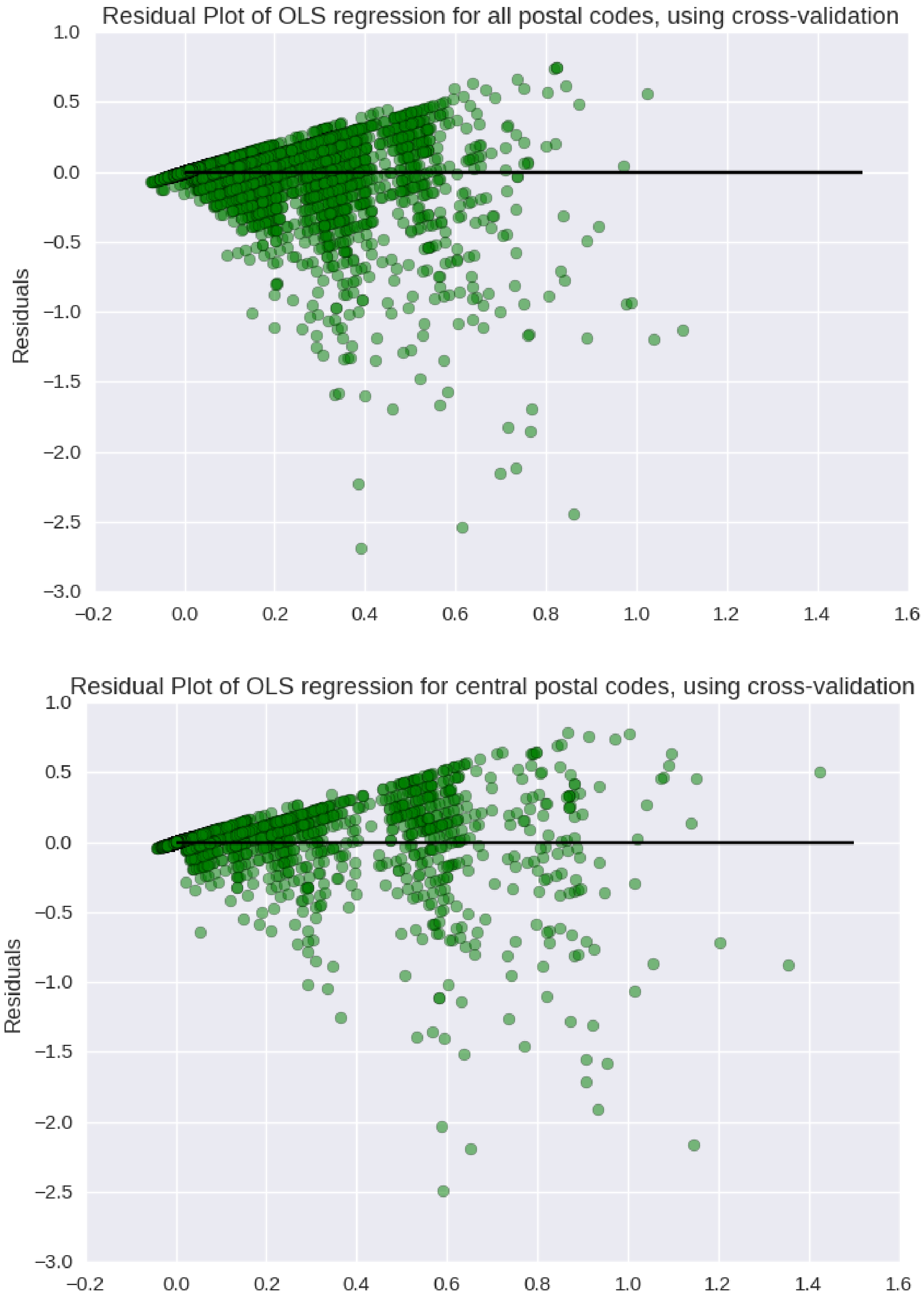


Figure 8: Residuals for orginary least squares regression on full(top) and central (bottom) data

While an R-square of around 0.6 is respectable for a dataset lacking accurate information on bike counts, presence of bike lanes, speed limits and intersections angles, we believed that some better performance could be gained by including locality information, in the form of postal codes. We used a random forest regression with varying features on the same test and training split (one each for the full and central only datasets), using the train_test_split method from sci-kit learn's model_selection module. This allowed a direct comparison of performance. Due to the nature of the random forest algorithm, cross validation isn't neccessary to prevent over fitting, so we are confident in the results, although we acknowledge that different train/test splits may produce different R-squared ranges. The results for this analysis are presented in Table 5.

Regression Algorithm	Intersections Included	Features	R-squared
Random forest	All	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.552963175315
Random forest	Central postal codes	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class	0.685287960362
Random forest	All	vehicle traffic, pedestrian traffic, control type, road class	0.510496830828
Random forest	Central postal codes	vehicle traffic, pedestrian traffic, control type, road class	0.577596318089
Random forest	All	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class, postal codes	0.603725999785
Random forest	Central	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class, postal codes	0.705692469952
Random forest	All	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class, feature important postal codes	0.603824492
Random forest	Central	Bike traffic, vehicle traffic, pedestrian traffic, control type, road class, feature important postal codes	0.706377039372

Table 5: Random forest regression results, performed on the same train/test splits, with varied features

The first set of forests were regressions on the two datasets with the same features as beforehand. We note that the scores are higher than the 5-fold cross-validated forests in Table 4. Thus, we are working with a relatively "easy" train/test split. The next two entries involve dropping the bike traffic estimates as a feature. This column was obtained after the power-fit made from comparing bike and pedestrian ratios at a few intersections in central Toronto, as discussed in our exploratory data analysis. We see a significant loss of predictive power in both data sets, although the effect drops the score by 0.11 for the central data set, as compared to around 0.04 for the full intersection data. This tells us two things. Firstly, the power-fit, while seemingly crude, does possess some helpful information, as it helps the predictive power overall. Secondly, the traffic estimates are relatively more powerful when looking at the central intersections only. This was expected, as our fit was made based only on intersections in this area, with nearest neighbour matching for a couple of small exception areas, also centrally located. Thus, outside of these areas, we had no reason to believe the power-fit held any actual predictive power.

The following entries include the bike traffic estimates, but also use postal codes as a feature. Figure 9 shows the resulting fit for the central dataset. We note the decreased propensity to underestimate collision rates.

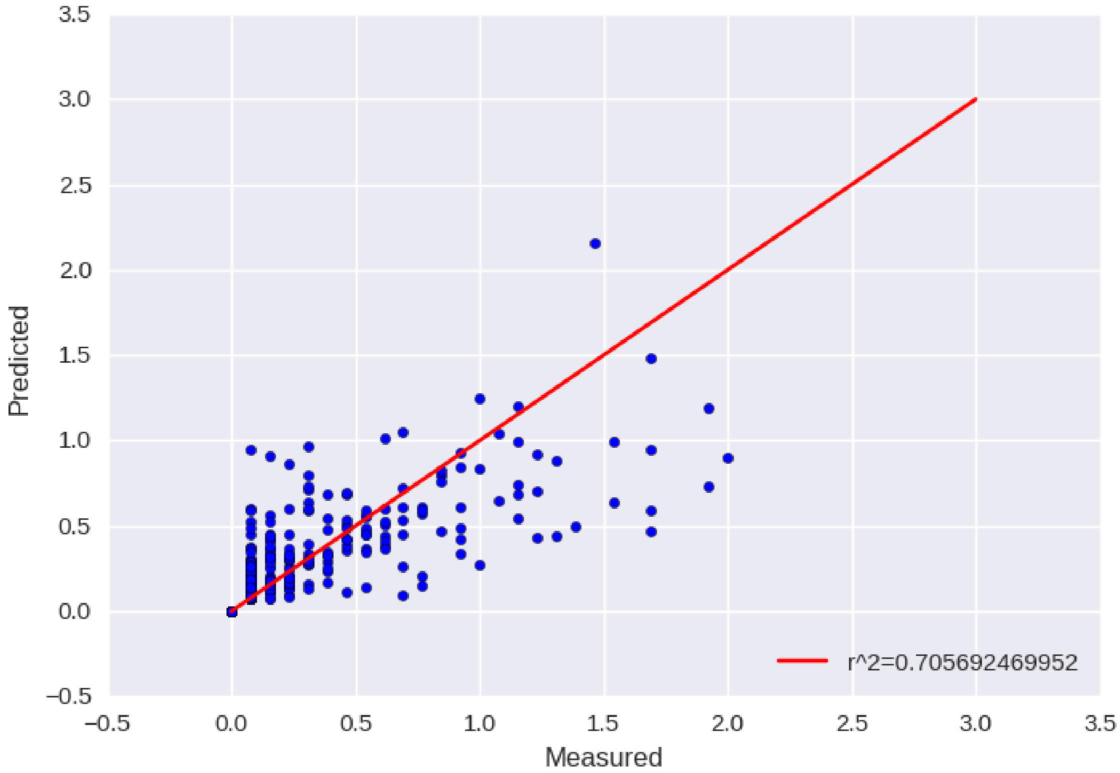


Figure 9: Random forest regression, including postal codes as features, for the central intersections.

The two entries include all postal codes found in the full and central datasets, respectively, as a binary feature. This use of dummy variables was not required for a random forest as for the linear regression algorithms, however it allowed us to use sci-kit learn's feature_importances_ method for random forest regressors. While we saw a significant jump in performance for the full and central datasets, up to ~0.6 and ~0.7 respectively, it would be naive to assume that every postal code carried significant information. We can use the SelectFromModel meta-transformer to specify a threshold (manually or as some multiple of the mean), to remove features which are irrelevant. Sci-kit learn does this natively using mean decrease impurity, which represents how much the variance is decreased by each feature on average. Another feature selection method is to measure how much each feature impacts the accuracy of our model. By permuating the values of each feature, we can check how much it affects our prediction. For important variables, there should be a significant decrease.

Both methods of feature importance were checked, and produced roughly the same order of variables at output. After specifying a cutoff of one quarter of mean impurity decrease, we transform our data to use the first 23 features (18 postal codes) and first 17 features (12 postal codes) for the central only data set. When looking at the individual postal codes, there is much overlap between the two lists. Running the random forests with these postal codes only, we actually stay on average at about the same predictive power as with all the postal codes included. Our results shows a very slight improvement in both cases, but this was only over a few runs and over a specific training/test split. The important take away is that we can in fact selectively remove certain postal codes as a feature, and not harm the algorithm's predictive score. Thus, our hypothesis that there are a few significantly more risky, or safer localities, along with a majority of functionally similar areas, is shown to have merit. Figure 10 plots the predicted collision rates against the measured ones for the central data set, showing that performance has not been hindered after filtering.

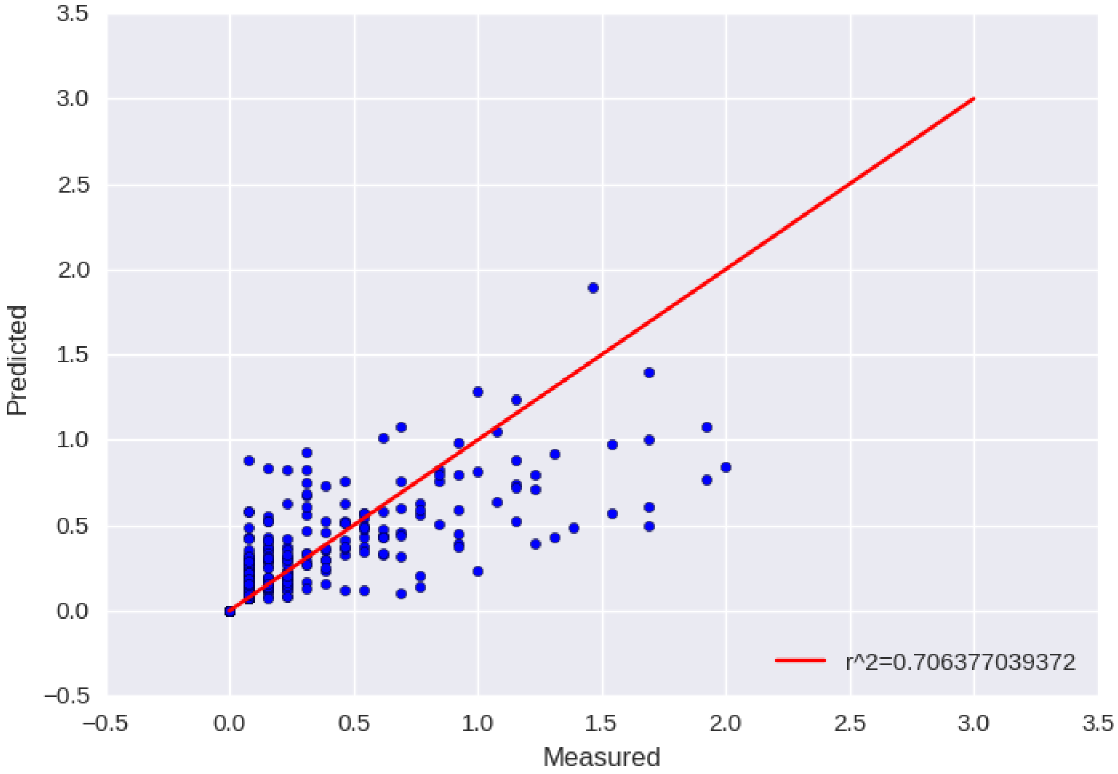


Figure 10: Predicted collision rate versus observed for random forest regression, using filtered postal codes. Central only intersections.

CONCLUSION AND FUTURE WORK

To discuss a final result, we'd like to see how our random forest with filtered postal codes performs on the entire dataset, by cross validating so that every single point is contained in the test set at some point, as we did for table 4. In table 5 we were using the random state of our training splits to compare changes to the same test sets each time. While a random forest should not be prone to overfitting, thus not requiring cross validation in the vast majority of cases, in order to gain a final answer on how well our forest performs, we'd like to generalize it's testing as much as possible. The results of these cross-validated random forest predictions against the measured collision rates are shown in Figure 11 and Figure 12.

While our score is lowered significantly, we still perform quite will, being able to explain over 63 percent of the variance in our data set with our very barebones model. Further improvements could be made by including distances to bike posts/parking lot, speed limits, road intersection angle and whether or not a bike lane is present at the intersection. While this would make the project more involved, combining these extra features with more data from the years since 2010 should result in a model that is able to closely predict real life collision rates. A future application, once bike lane location and/or type was made available, is to train the descision tree ensemble further with new data, and compare predictions for intersections before and after the installation of new infrastructure. With a these extra features, the City of Toronto should be able to better identify areas in need of intervention, and predict how cycling intfrastucture changes, encouragement of cycling traffic, or rerouting existing vehicle/pedestrian traffic affects the safety of cyclists. Similar datasets are available for the city of Montreal, but more complete and more fully featured. These would represent much less trouble in terms of wrangling. A planned future project is to apply other ensemble methods, along with our random forest model, to Toronto and Montreal cycling accidents, and see if the performance is similar across the two cities.

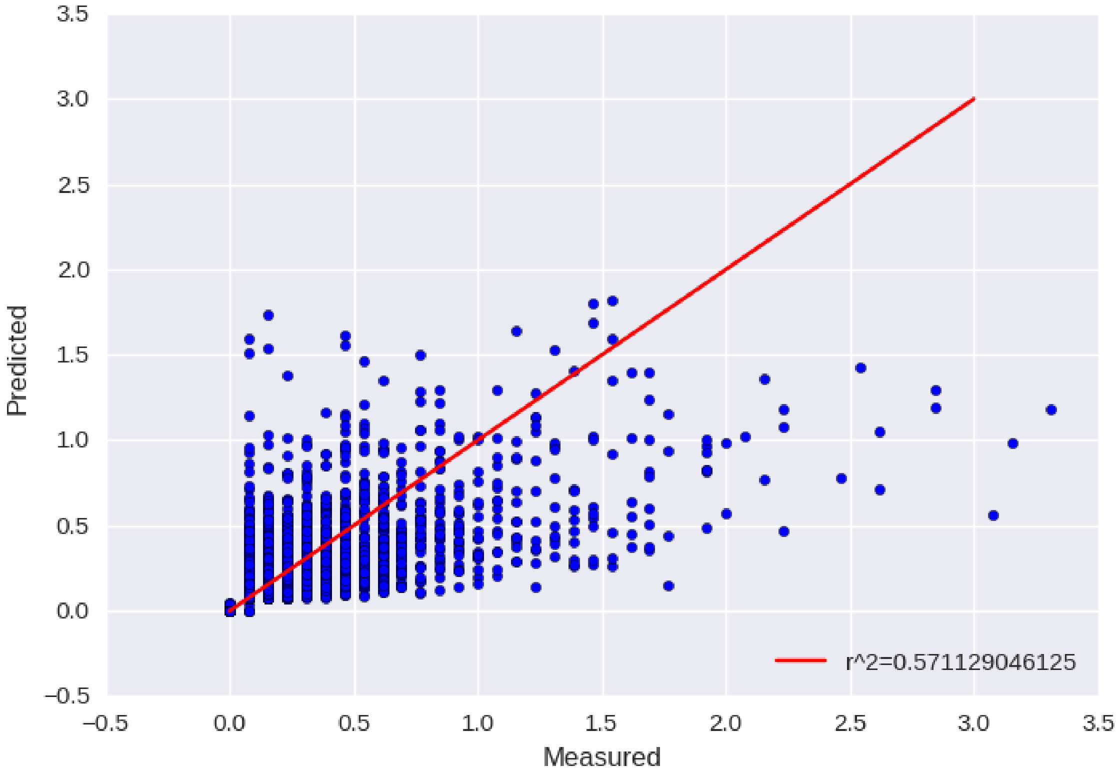


Figure 11: Cross validated predicted collision rates versus measured rates for all itnersections, using a random forest regressor.

In conclusion, we have managed to build a predictive random forest, which using only traffic numbers (and an exponentially fit bike traffic estimate), along with the road type of the busiest road, and the traffic control type, performs a reasonable estimate of the yearly collision rates (accidents per year) at intersections for cyclists. The scores are patricularily good when we restrict ourselves to the central post codes. This is partly attributed to the fact that the bike traffic estimates were fit with a few dozen points, all in the central neighbourhoods of Toronto. Had we better bike counts throughout the whole city, the predictive power would no doubt improve.

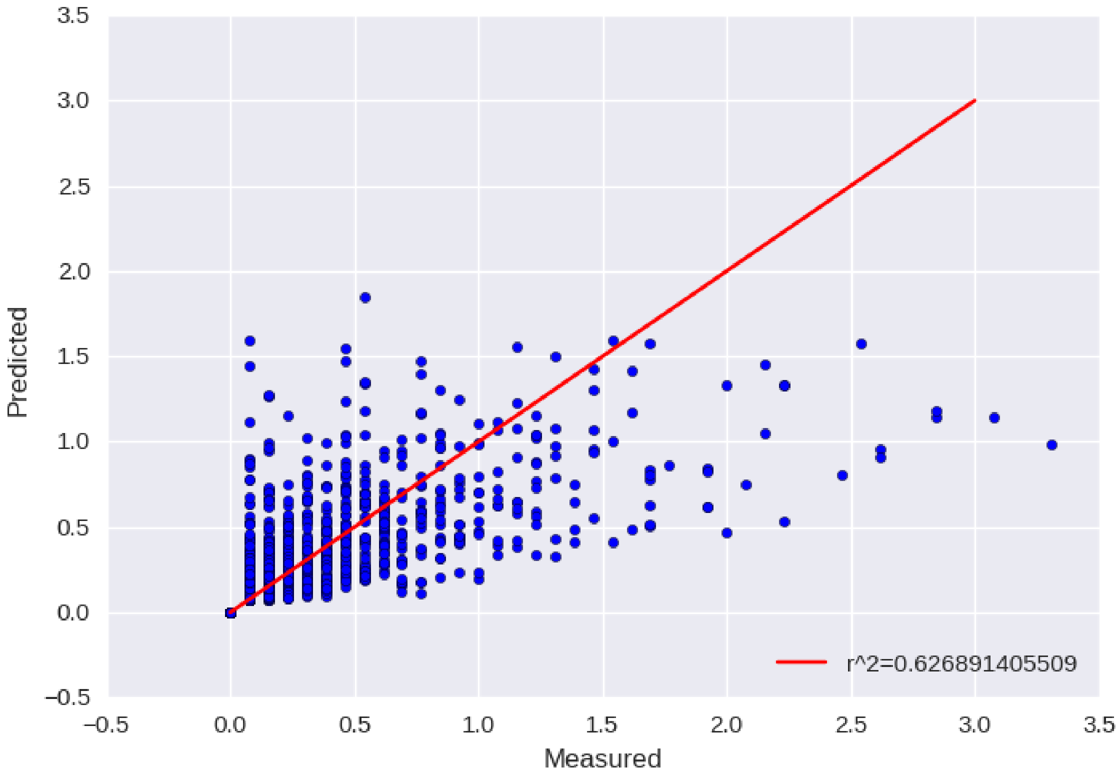


Figure 12: Cross validated predicted collision rates versus measured rates for all itnersections, using a random forest regressor.