A Model to Estimate Bike-Lane Demand

Aadarsh Vadakattu, Mohammad Amin Kuhail, and Harish Tata

***Abstract*—Unfortunately, bike-related accidents are very common. Some of these accidents can be deadly. On most occasions, these accidents happen due to the lack of a safer means of commute, a bike lane. Further, many citizens could potentially be cyclists if bike lanes were installed. Driven by the idea of improving safety and convenience for cyclists, we contribute to a model that estimates the bike-lane demand in the city. We argue that the demand for bike lanes increases as the number of bike-related accidents increases. Further, the demand increases as the number of popular businesses increases, since some citizens commute to work and get around by bike. Our model estimates the demand for bike lanes using accidents and ratings of businesses. Accidents are defined by features that represent the severity as well as the cause of the accident. Our model uses the Weight of Evidence algorithm to determine the significance of the accident features. Further, the model uses an algorithm that breaks down roads into equally sized sections based on the US addressing standards. The estimation of the bike-lane demand is expressed via scores assigned to road sections.**

***Index Terms*—Smart city, bike-lane demand, bicyclist safety, weight of evidence.**

I. Introduction

The number of cyclists is on rise. The United States alone had 43 million cyclists in 2013. That number increased to 47.5 million in 2017 [1]. Many people use bike share services, which also observed an increased use. Approximately 1 million people used these services in 2010, which increased to 35 million in 2017 [2]. Since many cyclists use roads without bike lanes, people have accidents that can injure or sometimes be deadly. Of the 35,000 road crashes in 2015, the number of cyclist fatalities was 818, which makes up 2.3% of the total fatalities [3]. Among these crashes, nearly a third of them happened because they were hit by a car [4]. In almost all these cases, 39% of bicyclists felt threatened because the motorists drove very close to them [4].

Similar to how cyclists increased exponentially, the usage of motor-based vehicles also increased gradually over time [5], [6]. With the increase in the number of cyclists, it would be inefficient for cyclists to use regular roads that are designed for motorists. Since bike lanes are used strictly by cyclists, the probability of a crash with a car would be less. Thus, more bike lanes would increase the safety for cyclists.

Apart from increasing safety, bike lanes have other benefits. More bike lanes motivate people living or working nearby to use them. This has a positive impact on their health. Increased bike usage also reduces reliance on fossil-fuel vehicles, which lowers pollution levels.

Motivated to improve the safety and convenience for cyclists, we propose a model that analyzes accidents and businesses in a section of a road and estimates the demand for a bike lane. The system computes the estimation by considering the safety and convenience for cyclists. Safety is determined by features from historical bike-related accident data, such as the severity and the type of the accident. Convenience for cyclists is determined by the existence of highly relevant businesses nearby. Historical bike-related accidents are described by several features. Each feature involves several attributes and every attribute has different levels of impact. These factors are thus weighted according to the model. The final estimates depend on the severity factors determined by an expert, along with a defined importance factor. This factor is obtained by using underlying evidence with the help of the Weight of Evidence (WoE) algorithm.

II. Related Work

The related work can be divided into three sections: safety measures for cyclists; scoring models; and estimations of bicyclist frequencies.

A. Safety Measures for Cyclists

As cities add bike lanes, the number of cyclists increases and cycling becomes safer [7], [8]. Many safety measures have been taken in the past to protect and increase the safety of cyclists in bike lanes. For instance, the Context-Sensitive Design is an approach applied while designing bike lanes [9]. This approach considers all possible "contexts of improvement" to improve the safety of cyclists riding in bike lanes. The city of Portland, Oregon, implemented blue thermoplastic "Yield to Cyclist" signs that improved yielding at the most crash-prone spots [10]. Studies also proved that the existence of green colored or highly visible bike lanes encouraged lower levels of conflict [11]. While similar measures are implemented to improve safety for cyclists in bike lanes, few measures are taken for cyclists' safety on regular roads. A law from the state of California states that the motorists who violate a three-foot space distance from a cyclist are subject to a fine [12]. Though these measures help, the safest path for a cyclist to ride is on a bike lane. According to a survey, 30% of cyclists wanted to have more bike lanes installed as they felt safer to ride [4]. Thus, the safety of cyclists and their confidence would be maximized if more bike lanes are installed.

Manuscript received May 17, 2019; revised August 1, 2019. This work was supported in part by the University of Missouri – Kansas City's Department of Computing and Engineering and the School of Graduate Studies.

The authors are with the University of Missouri, Kansas City, Kansas City, USA (e-mail: avhmf@mail.umkc.edu, kuhailm@umkc.edu, avhmf@mail.umkc.edu).

DOI: 10.7763/IJMO.2019.V9.716

B. Scoring Models

The scoring model is a weight-aggregating algorithm that is used primarily in evaluating the worthiness of credit applicants. Bankers look at credit scores of applicants assigned by the scoring model to check if the applicant is worthy of a loan. These credit scores are calculated aggregations of weights that determine the historical information of the applicant. The model follows an evaluation procedure where different attributes are assigned with relevant weights. These weights are aggregated to finally form an interpretable score [13].

The scoring models implement the WoE algorithm to determine the predictive power of each attribute of all variables. The predictive power is determined by dividing the dataset into events and non-events and applying the following formula to all attributes:

$$WoE = \ln\left(\frac{\%\ of\ non - events}{\%\ of\ events}\right) * 100 \qquad (1)$$

Here, usually, a non-event would be a good customer and an event would be a bad customer. Thus, the WoE will be positive if the percentage of non-events is a larger number and vice versa. This model can be fine-tuned and applied to other problems, similar to credit.

The second author designed an evidence-based personalized recommender system to recommend city residents to use bikes for transportation, when the historical trips of the user show evidence of using bikes in similar circumstances. The system used a scoring model that relied on the statistical learning of user transportation habits and preferences [23].

C. Estimation of Bicyclist Frequencies

There have been multiple procedures to estimate bicyclist frequencies. Texas A&M's Transportation Institute had estimated cyclist frequency by using technology such as inductance loop detectors fixed on bike paths, infrared sensors fixed on small poles, video cameras and video processing, and so on [21]. According to Turner's research, bicycle traffic can be roughly estimated using simple calculations that depend on local housing and commercial units [22]. Though Texas A&M's results are highly accurate, using Turner's methodologies can result in an approximate bicycle traffic estimate without the need of expensive equipment. While all these models help in bike traffic estimates, none of these approaches indicate the safety levels of cyclists or try to improve the safety standards. Further, these estimates do not incorporate the accident probabilities of cyclists.

III. Methodology

Our proposed model uses historical data to estimate the need of bike lanes in a city. The model assigns scores to sections of roads. The score is a number from 1 to 10, where 1 means the lowest demand and 10 means the highest demand.

The scores the model generates rely on two factors: severity and importance of the accident. The severity factor is a score assigned by an expert to each feature of an accident. A government roads department official can be the expert. However, we assigned the severity scores based on our personal judgment, and these scores can be adjusted by an expert in future versions of the model. The severity scores are later multiplied with an importance factor. The importance factor is estimated using the WoE algorithm, which determines how predictive each attribute is in determining the final score estimate.

In the following sections, we introduce the architecture of the model and how the individual components manipulate the data and return a score. The main components are the Geocoder, the Address Sectionizer, the Accident Severity module, the Evidence Calculator, and the Score Generator.

A. System Architecture

Figure 1 shows the system architecture of the model. The entire implementation can be found in Vadakattu et al. [20]. The historical accident and business data with latitude and longitude are sent to a Geocoder. The Geocoder generates standardized addresses for each combination of latitude and longitude. The data from the Geocoder are used by the Address Sectionizer, which groups the accident locations into road sections based on the street numbers. We estimate bike lane need to the sections defined by the Address Sectionizer. The Evidence Calculator uses a condition along with a tweaked WoE algorithm. The WoE helps estimate the feasibility of each attribute. Meanwhile, an expert's task is to identify the attributes of all the features of a bike accident and assign severity scores to the Accident Severity module. The Geocoder, the Address Sectionizer, the Evidence Calculator, and the Accident Severity module send their data to the Score Generator, where all the processed data are aggregated to obtain a final determined score. The Score Generator normalizes the scores on a scale of 1 to 10, where 10 would imply that the road highly needs a bike lane and a 0 means a bike lane is not needed.

In the following sections, a more detailed explanation of all components is given.

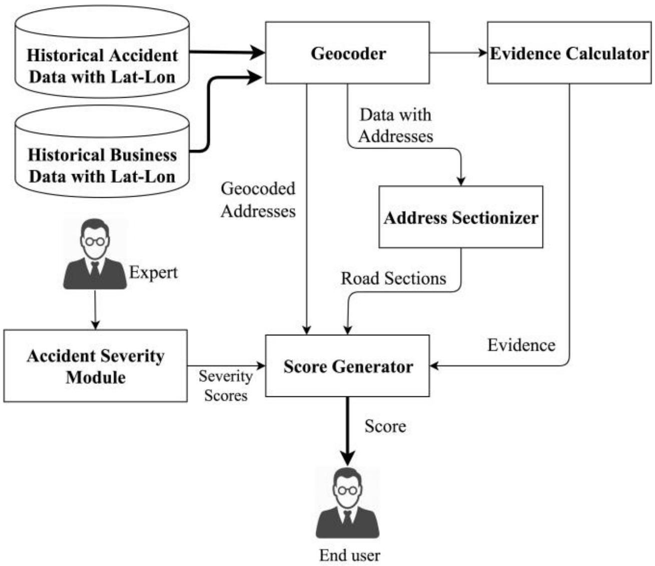


Fig. 1. Data flow architecture.

B. Geocoder

We used two datasets to determine the bike-lane demand estimate. For the accident information, we used historical accident data from the state of North Carolina [15]. For the business information, we used a dataset from Yelp [16]. The address data obtained from both the datasets needed to be cleaned, as there were many occurrences that had the same street names in different types of formats. For example, there were multiple occurrences of Park Place and Park Pl., which were the same road with different names. While correcting these issues, we also had to merge data from both the datasets and make the addresses unique for the same location. We also had to find a way to obtain missing addresses. For all of these purposes, we used the latitude (lat) and longitude (lon) of each accident occurrence and pushed them through an online geocoding service called Geocodio [17]. The Geocodio API returned a standardized address of a building that is the closest to the determined lat–lon pair. This simplified the cleaning process and made it easy to obtain the unknown addresses.

The Geocoder effectively returned addresses with address numbers and their street names in the format shown in Table I, for all the records.

TABLE I: GEOCODER OUTPUT

Latitude	Longitude	Dataset Address	GC Address Num	GC Street
35.250229	-80.790794	800 E SUGAR CREEK RD	800	E SUGAR CREEK RD
35.724439	-77.911416	GOLDSBORO ST	126	Goldsboro St SW
34.699022	-77.060245	TAYLOR NOTION RD	205	Taylor Notion Rd
36.099123	-80.248214	MARSHAL ST	424	N Marshall St
34.705645	-79.130441	SR 1513	3171	Evergreen Church Rd

C. Address Sectionizer

Initially, the dataset had accident occurrences with exact lat–lon point values. As the model aims to assign recommendations to a small stretch of a road, the accidents are needed to be grouped into pieces of road sections. The Address Sectionizer groups the accidents into sections of roads that are equal in length in terms of street numbers. The addresses in the US start with an address number, which is followed by the street name. A simple example is shown in Fig. 2.

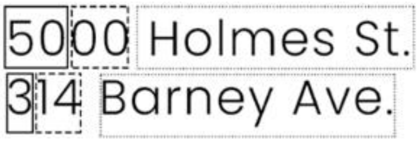


Fig. 2. Address standard example.

The address number is a concatenation of two numbers. The street number is as shown in the solid rectangle and the building number within the street segment is as shown in the dotted rectangle. We utilized this naming scheme and used the street number part of the address to group our accidents into road sections. First, the Address Sectionizer takes the address numbers returned by the geocoder and picks their minimum and maximum for each street name. The minimum address numbers are floored using the floor function [18] to obtain the nearest lower bound. The flooring is done by obtaining the closest numbers that are smaller than the minimum and are multiples of hundreds if the minimum is a value in hundreds, multiples of thousands if the minimum is in thousands, and so on. Doing this gave us imaginary start points of the address numbers with the original street numbers untouched. Similarly, the maximum address numbers are ceiled to obtain the higher bound using the ceiling function [18] to the closest bigger number of the maximum. For example, if a minimum of a road is 3421 and the maximum is a 6434, the lower bound would be 3000 and the upper bound would be 7000. In other words, the starting bound would be from the 30th street and the upper bound would be the 70th street. After this is done, we split the street number bounds into groups of five hundreds, which are our defined sections of a single long road. Doing this gave us sections of a road that would be five street numbers in length.

The Address Sectionizer would create floor and ceiling values as shown in Table II:

TABLE II: FLOOR AND CEILING OF ADDRESS NUMBERS

Street Name	Min (Street Number)	Max (Street Number)	Floor (Street Number)	Ceil (Street Number)
Abbey Pl	1512	1512	1000	2000
Albemarle Rd	5137	12901	5000	13000
Baybrook Ln	9305	9305	9000	10000
Bradford Dr	219	536	100	1000
Camp Stewart Rd	3700	5530	3000	6000

The module would group the roads into sections of five hundreds, as shown in Table III. These are the sections that would be utilized by the score generator to assign the estimate scores. Thus, in the following example, a section of Camp Stewart Rd. would start from the 30th street and would end at the 35th street.

TABLE III: FLOOR AND CEIL OF ADDRESS NUMBERS

Street Name	Start	End
Camp Stewart Rd	3000	3500
Camp Stewart Rd	3500	4000
Camp Stewart Rd	4000	4500
Camp Stewart Rd	4500	5000
Camp Stewart Rd	5000	5500
Camp Stewart Rd	5500	6000

D. Accident Severity Module

From the accident dataset, we selected the nine features listed in Table IV that would determine the cause of a specific accident.

TABLE IV: FEATURES OF THE ACCIDENT DATASET

Feature	Description	
	Type	*Info*
Ambulancer	Boolean	Determines if there was an Ambulance assistance after the crash.
Bikedir	Categorical	Determines if the accident happened because of an ongoing vehicle, a facing vehicle or none.
Bikeinjury	Categorical	Determines the injury levels of the cyclist.
Drvinjury	Categorical	Determines the injury levels of the motorist.
Bikepos	Categorical	Determines the position of the bike during the crash. Examples include Bike Lane, Travel Lane, etc.
Crashgrp	Categorical	Shortly describes the reason of the crash. Examples include: Motorist overtaking Bicyclist, Motorist Right Turn/Merge, etc.
Crashloc	Categorical	Determines the position of the crash with respect to the road intersection. Examples include: At an Intersection, Non-Intersection, Intersection related, etc.
Crashtype	Categorical	Describes the reason of the crash in further detail. Examples include Motorist Overtaking-Undetected Bicyclist, Motorist Left Turn-Opposite Direction and so on.
Development	Categorical	Describes the neighborhood. Examples include Residential, Commercial, etc.

All the features, with the exception of the first, are categorical. For the purpose of our model, we converted the categorical values using a scale from 0 to 5 to indicate the severity of the accident. The scale is interpreted as follows: 0

means the accident is the least severe or irrelevant for our purpose and 5 means deadly. We assigned the scores based on our personal judgment. However, ideally, an expert should assign the severity values to indicate higher scores to attributes that would be causing a severe bike accident on a road without bike lanes. Table V shows an example of severity scores for the Bikedir feature.

TABLE V: SEVERITY SCORES FOR BIKEDIR

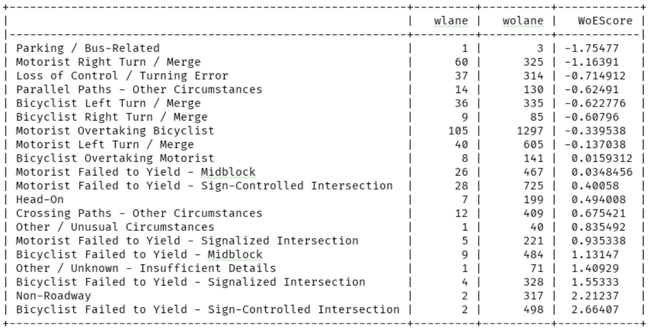
Bikedir	**Score**
Facing Traffic	5
With Traffic	3
Unkonwn	0
Not appicable	0

E. Evidence Calculator

The Evidence calculator evaluates the importance of each accident feature based on the predictive power of each attribute of every feature. To find the attribute importance, we designed the Evidence calculator so that it determines a value from 0 to 1 for all attributes, where 0 implies that the attribute is not important and 1 is highly important. This value is then multiplied by the accident severity scores in the Score Generator module. To add validity to our model, we wanted to weaken the accident features that were present in bike-related accidents that occurred in roads with bike lanes. To identify these importance values, we used the following formula:

$$WoE_{attribute\ ,raw} = \ln\left(\frac{\frac{no.of\ attribute\ specific\ accidents\ w/o\ bike\ lane}{total\ no.of\ accidents\ w/o\ bike\ lane}}{\frac{no.of\ attribute\ specific\ accidents\ with\ bike\ lane}{total\ no.of\ accidents\ with\ bike\ lane}}\right) \qquad (2)$$

We utilized the non-events as occurrences of bike accidents that happened on a road without bike lanes and the events as occurrences that happened in a road with bike lanes. Thus, a positive WoE means that there are accidents that happened on roads without bike lanes, but did not happen on roads with bike lanes. On the other hand, a negative WoE means that the accident features occurred on both roads with and without bike lanes. Hence, we use the negative WoE to weaken the influence of that accident feature. Fig. 3 shows the counts and WoE for the feature "Crashgrp":



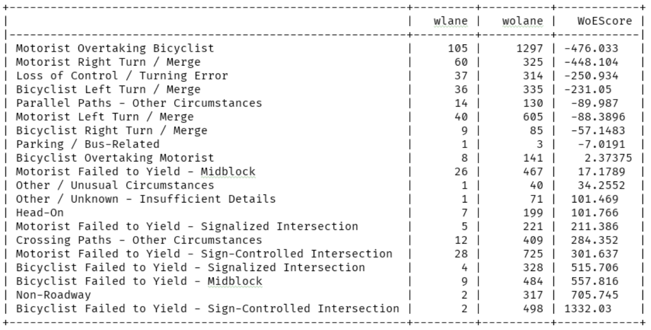
```
+-----------------------------------------------------------+----------+-----------+------------+
|                                                           |  wlane |   wolane |   WoEScore |
|-----------------------------------------------------------+----------+-----------+------------|
| Parking / Bus-Related                                     |        1 |         3 | -1.75477   |
| Motorist Right Turn / Merge                               |       60 |       325 | -1.16391   |
| Loss of Control / Turning Error                           |       37 |       314 | -0.714912  |
| Parallel Paths - Other Circumstances                      |       14 |       130 | -0.62491   |
| Bicyclist Left Turn / Merge                               |       36 |       335 | -0.622776  |
| Bicyclist Right Turn / Merge                              |        9 |        85 | -0.60796   |
| Motorist Overtaking Bicyclist                             |      105 |      1297 | -0.339538  |
| Motorist Left Turn / Merge                                |       40 |       605 | -0.137038  |
| Bicyclist Overtaking Motorist                             |        8 |       141 |  0.0159312 |
| Motorist Failed to Yield - Midblock                       |       26 |       467 |  0.0348456 |
| Motorist Failed to Yield - Sign-Controlled Intersection   |       28 |       725 |  0.40058   |
| Head-On                                                   |        7 |       199 |  0.494008  |
| Crossing Paths - Other Circumstances                      |       12 |       409 |  0.675421  |
| Other / Unusual Circumstances                             |        1 |        40 |  0.835492  |
| Motorist Failed to Yield - Signalized Intersection        |        5 |       221 |  0.935338  |
| Bicyclist Failed to Yield - Midblock                      |        9 |       484 |  1.13147   |
| Other / Unknown - Insufficient Details                    |        1 |        71 |  1.40929   |
| Bicyclist Failed to Yield - Signalized Intersection       |        4 |       328 |  1.55333   |
| Non-Roadway                                               |        2 |       317 |  2.21237   |
| Bicyclist Failed to Yield - Sign-Controlled Intersection  |        2 |       498 |  2.66407   |
+-----------------------------------------------------------+----------+-----------+------------+
```

Fig. 3. Raw WoE values for attributes of "Crashgrp."

The problem with using (2) as our evidence is that it does not consider the total count of occurrences of each accident feature. This might lead to the same WoE for two features if they had the same ratio but differed in their frequencies. For example, the WoE would be the same for two features where one has 1500 occurrences without a bike lane and 30 occurrences with a bike lane, and another has 4500 occurrences without a lane and 90 occurrences with a lane. We thought that the second scenario should be given a higher weight as that feature has more evidence. Thus, we updated the formula in this way:

$$\begin{gathered} WoE_{attribute\ ,tweaked} = WoE_{attribute\ ,raw} * (a+b) \\ a = no.of\ accident\ features\ on\ a\ road\ without\ bike\ lanes \\ b = no.of\ accident\ features\ on\ a\ road\ with\ bike\ lanes \end{gathered} \qquad (3)$$

The multiplication of total occurrences will increase the value of WoE, making them more significant. By implementing the above formula, we converted our values as shown in Fig. 4:



```
+-----------------------------------------------------------+----------+-----------+------------+
|                                                           |  wlane |   wolane |   WoEScore |
|-----------------------------------------------------------+----------+-----------+------------|
| Motorist Overtaking Bicyclist                             |      105 |      1297 | -476.033   |
| Motorist Right Turn / Merge                               |       60 |       325 | -448.104   |
| Loss of Control / Turning Error                           |       37 |       314 | -250.934   |
| Bicyclist Left Turn / Merge                               |       36 |       335 | -231.05    |
| Parallel Paths - Other Circumstances                      |       14 |       130 |  -89.987   |
| Motorist Left Turn / Merge                                |       40 |       605 |  -88.3896  |
| Bicyclist Right Turn / Merge                              |        9 |        85 |  -57.1483  |
| Parking / Bus-Related                                     |        1 |         3 |   -7.0191  |
| Bicyclist Overtaking Motorist                             |        8 |       141 |    2.37375 |
| Motorist Failed to Yield - Midblock                       |       26 |       467 |   17.1789  |
| Other / Unusual Circumstances                             |        1 |        40 |   34.2552  |
| Other / Unknown - Insufficient Details                    |        1 |        71 |  101.469   |
| Head-On                                                   |        7 |       199 |  101.766   |
| Motorist Failed to Yield - Signalized Intersection        |        5 |       221 |  211.386   |
| Crossing Paths - Other Circumstances                      |       12 |       409 |  284.352   |
| Motorist Failed to Yield - Sign-Controlled Intersection   |       28 |       725 |  301.637   |
| Bicyclist Failed to Yield - Signalized Intersection       |        4 |       328 |  515.706   |
| Bicyclist Failed to Yield - Midblock                      |        9 |       484 |  557.816   |
| Non-Roadway                                               |        2 |       317 |  705.745   |
| Bicyclist Failed to Yield - Sign-Controlled Intersection  |        2 |       498 | 1332.03    |
+-----------------------------------------------------------+----------+-----------+------------+
```

Fig. 4. Tweaked WoE values for attributes of "Crashgrp."

The final step is to calculate the importance factor, which is simply the normalized result of the WoE score. The value of the importance factor is between 0 and 1. We used the following formula to calculate the Importance Factor (IF) using the min–max terminology [19].

$$\begin{gathered} Importance\ Factor(IF)_{+ve} = 1 \\ Importance\ Factor(IF)_{-ve} = 1 - \left(\frac{|WoE|}{\max(|WoE|)}\right) \end{gathered}$$

Implementing the above formula to the tweaked WoE of "Crashgrp" yields the importance levels of attributes as shown in Fig. 5:

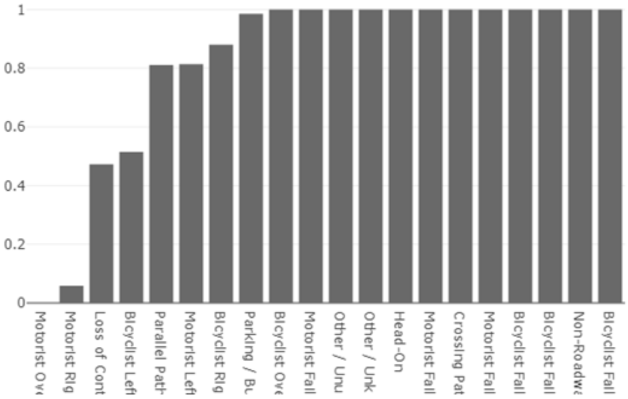


Fig. 5. Importance factor levels for "Crashgrp".

Thus, the IF helps in weakening the weights of accident features that happened even in the presence of bike lanes. We use this factor in the Score Generator module to multiply with the severity scores determined by the expert.

F. Score Generator

The Score Generator calculates a score representing the bike-lane demand for each road section. First, the Score Generator combines the severity scores with evidence from the evidence calculator by multiplying the IF with the severity scores. This yields attribute weights specific for each attribute, which are dependent on their severity and evidence.

$$Attribute\,Weight = Severity\ score * Importance\ Factor$$

Next, the start and end bounds of the address numbers are used from the Address Sectionizer to check with the address numbers of the individual accidents data. If the street numbers fall in the bounds of a specific street name, the attribute weights get added to the feature scores of that road section. Doing this for all accident occurrences gives us the total scores for individual features of the accidents that occurred in a specific section of the road. Furthermore, we similarly matched the address numbers of businesses with the start and end streets. After finding the matches, we added their average ratings to a business rating column. This meant that the score would be high if there are multiple businesses with a high average rating. We also made sure that there were at least a hundred reviewers for each business to be considered. The algorithm for the described process is as shown below:

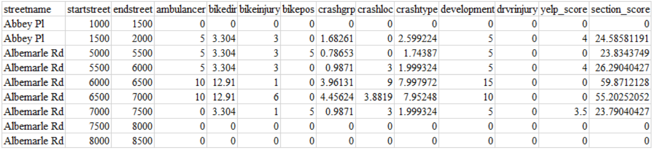
Algorithm 1: compute_final_feature_scores

```
input: a road section © from the Address Sectionizer, records from the crash dataset with attributes replaced with attribute weigh©(C) and list of features (F)
output: Feature Scores (F) being updated
foreach record in C do:
  if C['Street_Name']==R['Street_Name'] do:
    if R['Start_Street_num']<=C['Street_Num']<=R['End_Street_num'] do:
      foreach feature in F do:
        R[feature]+=C[feature]
end
```

As described, a similar algorithm is implemented for the business rating as well. To obtain a final score for the sections, we added the scores of all the features of the section along with the business rating.

$$final\ score = \Sigma(feature\ scores) + business\ rating \tag{6}$$

Using the above formula would yield our scores on undeterminable ranges. Fig. 6 shows a sample of the generated scores.



streetname	startstreet	endstreet	ambulancer	bikedir	bikeinjury	bikepos	crashgrp	crashloc	crashtype	development	drvrinjury	yelp_score	section_score
Abbey Pl	1000	1500	0	0	0	0	0	0	0	0	0	0	0
Abbey Pl	1500	2000	5	3.304	3	0	1.68261	0	2.599224	5	0	4	24.58581191
Albemarle Rd	5000	5500	5	3.304	3	5	0.78653	0	1.74387	5	0	0	23.8343749
Albemarle Rd	5500	6000	5	3.304	3	0	0.9871	3	1.999324	5	0	4	26.29040427
Albemarle Rd	6000	6500	10	12.91	1	0	3.96131	9	7.997972	15	0	0	59.8712128
Albemarle Rd	6500	7000	10	12.91	6	0	4.45624	3.8819	7.95248	10	0	0	55.20252052
Albemarle Rd	7000	7500	0	3.304	1	5	0.9871	3	1.999324	5	0	3.5	23.79040427
Albemarle Rd	7500	8000	0	0	0	0	0	0	0	0	0	0	0
Albemarle Rd	8000	8500	0	0	0	0	0	0	0	0	0	0	0

Fig. 6. Final aggregated scores sample.

To normalize the scores, we used a range of the scores in between 0 and 10, assuming 10 would definitely need a bike lane and vice versa. To fit the obtained scores in this range, we wanted to determine an upper bound that can be the worst possible score for a road section. Any score above this determined score would automatically get 10 as the score. We thought that attaining a score that is higher than the defined upper bound is equally bad as getting the score equal to the upper bound, as both scores would definitely need a bike lane regardless of their bad scores. We can determine the lower bound as a 0 directly, assuming that the road section has no accidents and no businesses.

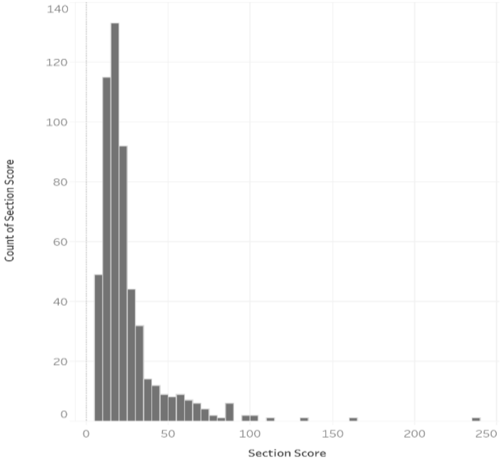


Fig. 7. Section scores histogram.

By looking at the histogram in Fig. 7, we recognized that the scores above 100 are outliers. Thus, we can consider 100 as our upper bound. With these bounds, we implemented the min–max algorithm [19] to obtain a score from 0 to 1. As we needed a score from 1 to 10, we simply multiplied this value with 10. We also determined that any score that crosses the upper bound would be replaced with the upper bound while applying the min–max normalization. Thus, the normalized scoring formula is as follows:

$$\begin{gathered} normalized\_score = \frac{non\_normalized\_score\ - min}{max\ -\ min} \\ max = determined\ by\ ignoring\ outliers\ and \\ finding\ the\ maximum\ possible\ non-outlier \\ min = 0, a\ road\ section\ with\ no\ accidents\ and\ no\ businesses \end{gathered} \tag{7}$$

Thus, on applying the above formula to the total scores, we get the final normalized scores, which determine the need of a bike lane to a specific section of the road.

G. Data-Independent Model

From the scores generated by the score generator, it is possible to generate a statistical model that would be independent of the data generated by the Evidence Calculator, the Accident Severity module, and the Address Sectionizer every time to obtain a score. This would reduce the amount of dependency on the data to obtain the finalized scores. The model is shown in Fig. 8.

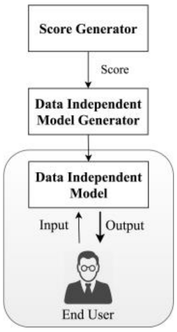
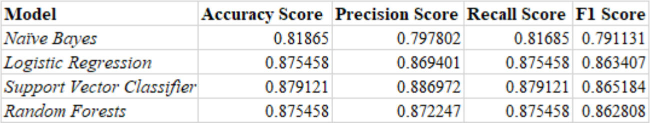


Fig. 8. Data-Independent Model architecture.

The complete score data from the score generator are taken by the Data-Independent Model Generator, which would apply one among the existing statistical models to the data. Before doing this, the module converts the scores into classes by taking a simple assumption. The assumption would split the scores into four equally divided classes. As the scores range from 0 through 10, the classification can be done as follows: scores from 0 to 2.5 do not need a bike lane at all, 2.5 to 5 might not need a bike lane, 5 to 7.5 might need a bike lane, and 7.5 to 10 would definitely need a bike lane. By making our data classifiable, we can fit classification models to our data. By looking at the existing data, we assumed Naïve Bayes, Logistic Regression, Support Vector Classifier, and Random Forests to be good fits. We trained all the above models and obtained accuracy scores, as shown in Fig. 9:



Model	Accuracy Score	Precision Score	Recall Score	F1 Score
Naïve Bayes	0.81865	0.797802	0.81685	0.791131
Logistic Regression	0.875458	0.869401	0.875458	0.863407
Support Vector Classifier	0.879121	0.886972	0.879121	0.865184
Random Forests	0.875458	0.872247	0.875458	0.862808

Fig. 9. Accuracy scores of statistical models.

By looking at the accuracy scores, it can be understood that Support Vector Classifier and Random Forests work best for this scenario. Thus, the Data-Independent Model can be either of two models. Thus, when the user gives different attributes of a road section as inputs, he/she can expect a class output instead of a score that would directly recommend a bike lane.

IV. Results and Evaluation

A. Performance of the Estimates

To estimate the score's performance, we visualized them along with the counts of accidents and businesses. Figure 10 shows the graph with non-normalized scores. The same conclusions can be drawn from Fig. 11, a plot with normalized scores.

On observation, we can determine that the total counts of accidents and businesses are definitely responsible for the generated score. Moreover, it is understandable that more accidents in a section lead to a higher score. The fluctuations can be related to the attribute scores, evidence, and the businesses ratings of every accident.

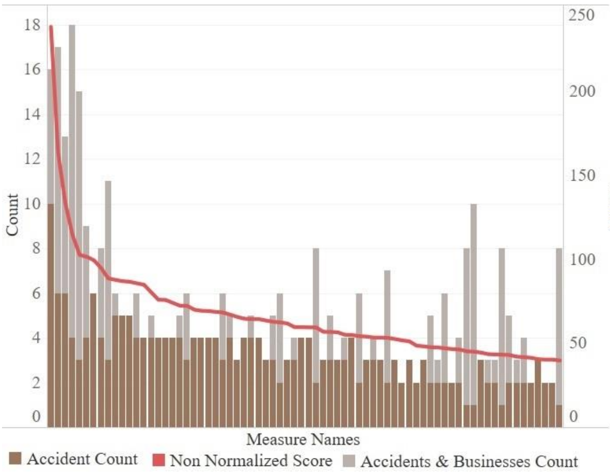


Fig. 10. Non-normalized scores vs. accident and business count.

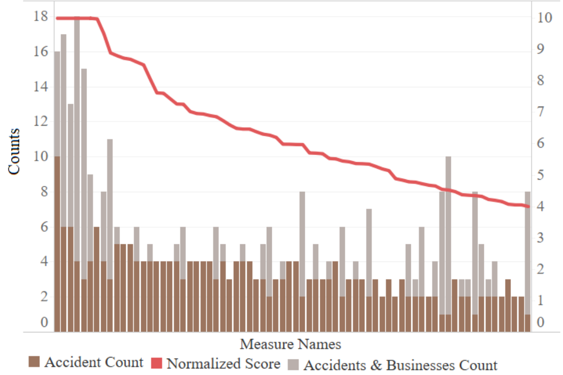


Fig. 11. Normalized scores vs accident and business count.

B. Visualized Estimates

For a better understanding of our results, we visualized the estimates of one road on a street map based on a coloring scheme. Similar to how we split the scores for the Data-Independent Model, we categorized the scores from 0 to 2.5 as green, 2.5 to 5 as yellow, 5 to 7.5 as orange, and 7.5 to 10 as red. The scores visualized for a few road sections of "Beatties Ford Rd" are as shown in Fig. 12:

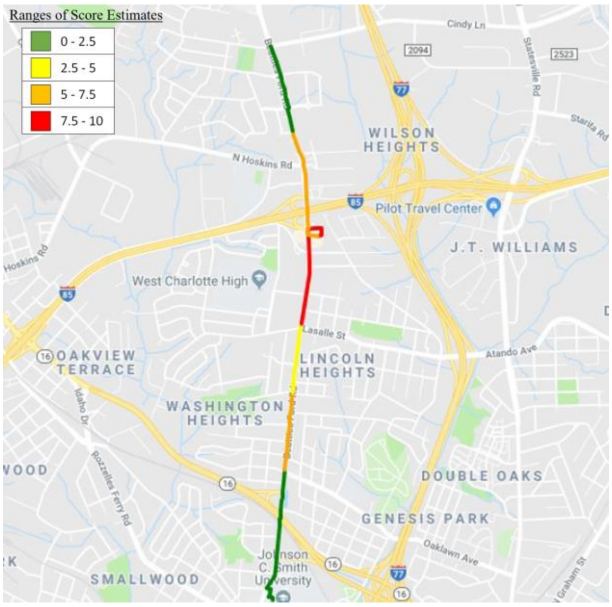


Fig. 12. Visualized estimates of Beatties Ford Rd.

By looking at the visualization, it is evident that the ends of our road section need not have a bike lane, but the sections in between the ending sections are suggested to have bike lanes to improve safety standards for cyclists.

C. Why This Model?

Instead of estimating the bike-lane demand using scores and determined weights, it is also possible to estimate the demand directly, using statistical predicting algorithms. To achieve this, the expert must observe all the accident occurrences in a few road sections and must assign the need for bike lanes to these road sections. These few road sections can be given as a training set to a predicting algorithm of our choice. Though this is possible, there are several problems that might affect the efficiency of the model. Some of them are as listed:

1) The dataset of the model is too small to be reliant on predicting algorithms. If these few accidents are used to train the model, there might be possibilities of overfitting on the less amount of data that we have. Our model uses

the WoE algorithm to determine the strength of the prediction, which has less scope of overfitting when compared to statistical models.

2) Almost all statistical models are not easy to understand. Comparatively, our model proposes simple aggregations along with an evidence calculating algorithm that can be easily understandable.
3) The predicting algorithms identify the patterns and weigh the results, which cannot be easily determinable. Our model determines this importance using the IF, which could easily determine what specific attribute is more responsible in determining the final estimate. The IF is easier to understand than recognizing the patterns in a statistical model.
4) Though both approaches need the help of an expert, it would be difficult for the expert to estimate the bike-lane need on his own completely for a few road sections, which is to be given as a training set to a predicting algorithm. Comparatively, our model requires simple severity scores on a scale of 0 to 5 for all the attributes, where 5 is the most severe.
5) There have been instances of applications that involved complex machine learning strategies to find creditworthiness, a problem that is similar in many ways to estimating the bike-lane demand. The results showed very low accuracy increments compared to simpler scoring models and as these models are not easily understandable, they have a negative impact on interpretability [14]. Alternatively, the scoring model approach used by our algorithm has high interpretability and is simpler to execute.

Thus, our model works better in several ways to estimate the need of a bike lane and is easier to understand.

V. Conclusion and Future Work

In this paper, we introduced a model to estimate the bike-lane demand, a data-driven model that can potentially help city officials in determining the demand of a bike lane based on its various features. Based on historical accident data, a specific section of a road is assigned a score based on accident features that are most responsible for the occurrence of a bike-related accident. Also, cyclist traffic and their convenience are estimated using business data, imagining of more businesses leading to more bike traffic. Along with the scores being based on intuition, the computed algorithms weaken the scores of those accident features that correlated less with bike-related accidents. This evidence is extracted by comparing the occurrences of accidents that happened with the existence of a bike lane and those that happened without their existence. This is done by utilizing a tweaked WoE formula. In the future, we plan to fit neural network models and evaluate the implementation accuracy of the Data-Independent Model. We also plan to increase the number of features, such as surrounding households, demographic information, etc., that can improve the estimates. We also look forward to using datasets from other parts of the world and modify the model to make it location independent. We are planning to develop a simple application that would display the estimates based on the input location, making it easier for the users to use the model.

References

[1] Statista. (2019). Bicycling Statistics : number of participants U.S. 2017. [Online]. Availablet: https://www.statista.com/statistics/191204/participants-in-bicycling-in-the-us-since-2006/

[2] NACTO Report: Bike Share Services in 2017. (2018). [Online]. Availablet:: https://nacto.org/wp-content/uploads/2018/05/NACTO-Bike-Share-2017.pdf

[3] National Center for Statistics and Analysis. (2017). Bicyclists and other cyclists: 2015 data. Traffic Safety Facts. Report No. DOT HS 812 382. [Online]. Availablet: https://crashstats.nhtsa.dot.gov/#/

[4] U. S. Department of Transportation. (2013). National survey of bicyclist and pedestrian attitudes and behavior volume 2: Findings report. [Online]. Availablet: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/811841b.pdf

[5] U.S. Department of Transportation (2019). Transportation Statistics Annual Report 2018. [Online]. Availablet: https://doi.org/10.21949/1502596

[6] Hedges & Company, U.S. Vehicle Registration Statistics. (2019). U.S. Vehicle Registration Statistics From Registration Databases. [Online]. Available: Https://Hedgescompany.Com/Automotive-Market-Research-Statistics/Auto-Mailing-Lists-And-Marketing/

[7] National Association of City Transportation Officials. (2016). Equitable bike share means building better places for people to ride. [Online]. Availablet: https://nacto.org/wp-content/uploads/2016/07/NACTO_Equitable_Bikeshare_Means_Bike_Lanes.pdf

[8] P. L. Jacobsen. (2003). Safety in numbers: More walkers and bicyclists, safer walking and bicycling. [Online]. Available: https://injuryprevention.bmj.com/content/9/3/205

[9] G. Dondi, A. Simone, C. Lantieri, and V. Vignali. (2011). Bike lane design, the context sensitive approach. In: 2011 International Conference on Green Buildings and Sustainable Cities. Bologna, Italy: Elsevier. [Online]. Availablet: https://www.sciencedirect.com/science/article/pii/S1877705811049265

[10] W. Hunter, W., L. Harkey, D., J. Stewart, R. and L. Birk, M. (2019). Evaluation of blue bike-lane treatment in Portland, Oregon. [Online]. Available: http://industrializedcyclist.com/bluebikelane.pdf

[11] A. Sadek, "Effectiveness of green, high-visibility bike lane and crossing treatment," in *Proc. Transportation Research Board 86th Annual Meeting*, Transportation Research Board, 2007.

[12] Stanford University. (2019). Give bicyclists 3-foot clearance when passing. [Online]. Available: https://news.stanford.edu/news/2014/november/threefeet-bike-law-110314.html

[13] N. Siddiqi, (n.d.). Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring. [ebook] SAS Institute Inc. [Online]. Available: https://support.sas.com/content/dam/SAS/support/en/books/cedit-risk-scorecards/59376_excerpt.pdf

[14] E. Dornhelm. (2018). Can machine learning build a better FICO Score?. [Online]. Available: https://www.fico.com/blogs/risk-compliance/can-machine-learning-build-a-better-fico-score/

[15] North Carolina Bicycle Crash Data. (2017). North Carolina Department of Transportation. [Online]. Available: https://catalog.data.gov/dataset/north-carolina-bicycle-crash-data

[16] Yelp Open Dataset. Yelp. [Online]. Available: https://www.yelp.com/dataset/download

[17] Geocodio. (2019). Geocodio. [Online]. Available: https://www.geocod.io/

[18] En.wikipedia.org. (2019). Floor and ceiling functions. [Online]. Available: https://en.wikipedia.org/wiki/Floor_and_ceiling_functions

[19] En.wikipedia.org. (2019). Feature Scaling. [Online]. Available: https://en.wikipedia.org/wiki/Feature_scaling.

[20] A. Vadakattu, M. A. Kuhail, T. Harish, A model for estimating bike lane Demand: implementation. [Online]. Available: https://github.com/thedatabuddy/BikeLaneDemandEstimation

[21] Texas A&M transportation institute, pedestrian and bicyclist counts and demand estimation study, houston-galveston area council. (2013). [Online]. Available: http://tti.tamu.edu/documents/TTI-2013-3.pdf

[22] S. Turner, G. Shunk, and A. Hottenstein, "Development of a methodology to estimate bicycle and pedestrian travel demand," TTI Report No. 1723-S, College Station: Texas Transportation Institute, 1988.

[23] M. Kuhail, B. Ahmad, and C. Rottinghaus, "Smart resident: A personalized transportation guidance system," in *Porc. 2018 IEEE 5th International Congress on Information Science and Technology, Marrakesh*, Morocco, 2018.

Aadarsh Vadakattu belongs to Kakinada, Andhra Pradesh, India. He received his B.S. and M.S. degrees in computer science from the University of Missouri – Kansas City, USA, in 2018 and 2019, respectively.

He worked as a software engineering intern and a data scientist intern for approximately a year in two different startups in Kansas City, USA. His research interests include data analysis, data mining, machine learning, and data science.

Mohammad Amin Kuhail received his M.Sc. in computer engineering from the University of York, York, UK, in 2006 and Ph.D. in computer science from the IT University of Copenhagen, Copenhagen, Denmark, in 2012.

He is an assistant professor at the University of Missouri – Kansas City, USA. He is an experienced computer scientist and software engineer with a diverse skill set that spans web development, object-oriented programming, algorithms, usability, and data science. He is an accomplished researcher and educator, with a proven record of publications, ABET and pedagogical training, and awards. He is a seasoned professional who lived in different countries and is connected with many people from various cultures and backgrounds. He is also a successful project manager, with over four years' experience of leading diverse teams and professionals and advising students.

Harish Tata belongs to Hyderabad, Telangana, India. He received his B.Tech in computer science from Gitam University in 2015. He is now pursuing his M.S degree from the University of Missouri Kansas City, Kansas City, USA, in computer science and is expected to graduate in 2019.

He worked as a senior product developer for more than 3 years in Kony India Pvt. Ltd. in India. His interests include data science, deep learning, data analysis, and data mining.