NavigAid: A Route Analysis and Navigation Model for Pedestrian Safety Using Random Forest Algorithm

Rayna Yu, Olivia Wang, and Paris Wu

Abstract Existing navigation tools help users find the quickest route between different locations, but when generating walking routes, these tools often fail to consider key factors affecting pedestrian safety and accessibility. This paper introduces a random forest model for analyzing the safety impact of pedestrian infrastructure features along projected routes and assessing vehicular-pedestrian crash risk. Evaluation of the model yields results that align with past accident analysis findings, showing a strong positive correlation between speed limit and crash risk, and weaker but significant impacts of factors such as sidewalk connectivity and damage, crosswalk coverage, tree canopy coverage, and lighting conditions. We incorporate the model into a pedestrian-oriented mobile navigation app, NavigAid, which shows information about aggregated crash risk, route distance, duration, and relevant street-level features, allowing users to determine the safest walking routes between any locations in Boston. The model provides insight into the effects of pedestrian-accessible infrastructure on pedestrian safety, which may be valuable for the planning of accessible and walkable urban networks.

1 Introduction

The history of urban development in the United States has followed a pattern of urban sprawl facilitated by motor vehicles as a primary mode of transportation since the early 20th century. In the 1920s, the rising automotive industry in the US capitalized on the emerging middle class's concerns by promoting cars as the ideal method of

Rayna Yu
Northeastern University, Boston, MA 02115, United States e-mail: raynayu27@gmail.com

Olivia Wang
Brown University, Providence, RI 02912, United States e-mail: olivia.b.wang@gmail.com

Paris Wu
Cornell University, Ithaca, NY 14850, United States e-mail: cw989@cornell.edu

transportation to cities [1]. As a result, most urban infrastructure in the US was designed around cars, while pedestrian needs were systematically under-invested, especially in redlined communities [50]. Furthermore, popular navigation systems in the US market today continue to ignore pedestrian accessible infrastructure, which may lead to inaccuracies, simplifications, and misrepresentations [19]. This inconsideration may lead to an increased risk of vehicular-pedestrian crashes [49], which is not only a public health concern, but of equitable urban planning as well.

In 2022, about 16% of Americans reported walking as a way of transportation frequently, and adults with a family income below the federal poverty line were the most likely to use walking as a form of transportation [1]. However, pedestrian fatality rates are higher for adults from lower socio-economic communities, especially for Black men, who are twice as likely to be killed due to systematic underinvested pedestrian infrastructure in communities of color [48]. Indeed, wider adoption of design interventions as well as systematic changes of US urban planning are needed [49], as opposed to placing culpability of vehicle-pedestrian crashes on individual behaviors as historically encouraged by the automotive industry [39].

This paper presents a model for analyzing pedestrian routes with a focus on pedestrian-accessible infrastructure and street-level features in order to assess the risk of vehicular-pedestrian crashes. The random forest model not only considers route features such as sidewalks, crosswalks, and ramps, but also considers street furniture such as streetlamps and trees. The presented model analyzes pedestrian-accessible infrastructure and road features on a projected path from a user-chosen point A to point B, aiming to find projected routes for pedestrians that are reliable, safe, and accurate to real-world conditions.

2 Related Work and Background

Several previous studies investigated pedestrian safety by analyzing factors associated with vehicle-pedestrian crashes, and researchers have introduced navigation modeling frameworks that aimed to center pedestrian safety by taking built environments into account.

2.1 Pedestrian Safety Risk Factors

Osama and Sayed (2017) developed a comprehensive set of characteristics of the connectivity, directness, and topography of the sidewalk network in relation to pedestrian safety and traffic accidents. By analyzing historical data on pedestrian-motorist crashes in Vancouver, they found a positive correlation between intersection density and pedestrian-motorist crashes; a negative correlation between sidewalk coverage and crashes; a negative correlation between continuous sidewalks with longer edge lengths and crash risk; and a positive correlation between sidewalk irregularity and crash risk [45].

Hussain et al. (2019) investigated the relationship between impact speed and the probability of pedestrian fatality during a vehicle-pedestrian crash, evaluating reported data on pedestrian fatalities from motorized vehicle crashes with known estimated impact speeds. They found that when the estimated impact speed increases by 1 kilometer per hour, the odds of pedestrian fatality increase on average by 11% [21].

Ferenchak et al. (2022) analyzed motor-pedestrian crashes in California and showed that pedestrian injuries are more frequent and severe for nighttime crashes than during the day. Furthermore, the study found that collisions that occur at night in poor lighting conditions correlate with increased severity of pedestrian injuries [15].

Zhu et al. (2022) examined the relationship between street trees and vehicle-pedestrian casualties in Melbourne City, Australia. Their results indicated that pedestrian injury is negatively associated with tree density and tree canopy cover; related studies proposed that this may be because urban street trees and larger tree canopy can increase driver awareness and pedestrian safety perception, respectively [53].

2.2 Framework for Measuring Pedestrian Accessibility

Liu et al. (2022) provided a low-cost and convenient framework for measuring pedestrian accessibility across a variety of cities using crowd-sourced and open-source data from OpenStreetMap. They measured accessibility by identifying points of interest such as grocery stores, schools, public transit stations, etc., then finding the distance between these nodes to arrive at a measurement of accessibility for daily living needs, using this as a proxy measurement of neighborhood accessibility [25]. However, they mainly relied on road-centerline data, which may be less relevant to pedestrians when roads do not always correspond to pedestrian path networks. Furthermore, they focused on measuring accessibility of static regions, while our work is focused on measuring accessibility of routes.

To address the challenges of using road-centerline networks for pedestrian routing, Hosseini et al. (2023) developed a new methodology that uses satellite imagery to identify pedestrian paths to address the limitations of previous work that relied on road-centerline data, which may not accurately reflect pedestrian path networks. Their Tile2Net model showed promising results in detecting pedestrian infrastructure elements from orthorectified aerial tiles [19].

2.3 Pedestrian Route Choice

Joo and Kim (2011) introduced a pedestrian route guiding framework that uses an analytical hierarchy process (AHP) through the Expert Choice Tool. Although their development of some factors that influence accessibility (such as traffic velocity, crosswalks, sidewalk width, street trees, and lighting conditions) agreed with recent studies on the built environment and pedestrian preferences [23], they failed to justify how their pedestrian route preference factors were selected and weighed in detail. The results of their study are difficult to replicate, as their process of weighting preference factors lacks transparency.

The review of previous work shows that although a few studies have attempted pedestrian-centered routing frameworks, these studies rely on road-centerline data and subjective weights to measure accessibility, and so fail to demonstrate a comprehensive understanding of pedestrian safety in line with recent best practice. This paper addresses the limitations of previous scholarship by developing a framework that not only incorporates a model that identifies accurate pedestrian paths through satellite imagery but also generates safer routes by considering quantifiable factors that influence pedestrian safety according to the latest research.

3 Data and Methodology

To assess the accessibility and safety of a given point in Boston, we used historical crash data with geospatial information from public datasets to produce a binary classification of crash risk.

3.1 Data Collection

Data from Analyze Boston are used to develop the models in this study. Respective ArcGIS Server services Directory REST APIs [12] were used to call data from individual datasets provided by Analyze Boston. Explanatory variables of tree coverage, speed limits, sidewalk damage, crosswalk coverage, streetlight locations, ramp coverage, and pedestrian crashes were extracted. Specifically:

- BPRD Trees [32], created by the Boston Parks and Recreation Department in June 2025 and last updated in July 2025, provided information on parks and street trees.
- Boston Street Segments [28], released by the Street Address Management system in August 2015 and modified nightly until November 2024, provided street segments data. Speed limit data were included in the analysis.
- Sidewalk Inventory [31], completed by the Boston Public Works Department (PWD) in 2014, provided information on Boston sidewalks that was last modified

in December 2023. Geospatial data of sidewalk damage and irregularity were included in the analysis.

- Sidewalk Centerline [30], created in 2011 and last modified in December 2023, provided data on sidewalks, private walks, and crosswalks that were used in the analysis.
- Streetlight Locations [41], maintained by the Street Lighting Division of Boston PWD and published in December 2016 by the Department of Innovation and Technology, provided longitudinal and latitudinal data of streetlamps that were included in the analysis.
- Pedestrian Ramp Inventory [29], completed by the Boston PWD in 2014, provided geospatial data of pedestrian ramp coverage that was used in the analysis.
- Boston Vision Zero [40], compiled by the Department of Innovation and Technology from the City's Computer-Aided Dispatch (911) system, provided records of the date, time, location, and type of crash for incidents requiring public safety response.

3.2 Analysis Variables

The variables included in our analysis are divided into three main categories: direct safety features (estimated impact speed and sidewalk network), indirect safety features (lighting conditions and tree canopy), and pedestrian accessibility (ramp coverage and sidewalk damage). As our literature review revealed, estimated impact speed, traffic exposure, and network topography have strong correlations with vehicular-pedestrian crashes [21] [19], and the relationship between vehicular-pedestrian crashes and lighting conditions and with tree canopy is secondary. Specifically, while vehicular-pedestrian crash fatality rates were 2.4 times higher outside the presence of a streetlight, the lack of street lighting does not seem to be the cause of the disproportionate increase in nighttime pedestrian injuries [15]. As for street trees, while tree canopies are negatively associated with pedestrian casualties, their effects are random as they may vary across individual road segments, especially during different times and weather conditions [53]. Nevertheless, both lighting conditions and tree canopy mitigate the risks of vehicular-pedestrian crashes to a certain extent, and provide perceived safety to pedestrians [37].

3.2.1 Estimated Impact Speed

Speed is the primary contributory factor in vehicular-pedestrian crash risk and severity. Meta-analysis of accident analysis research yields an optimal speed limit of 20-25 mph in high pedestrian activity zones, while higher speed limits are correlated with increased risk of pedestrian fatality [21]. Although Vision Zero's dataset does not indicate the impact speed of crashes, it may be estimated through roadway speed limits as they correlate with average travel speed [11], and thus risk of a pedestrian crash.

Hence, speed limit data from Boston Street Segments [28] are used as explanatory variables for the estimated impact speed in our analysis.

3.2.2 Sidewalk Networks

Sidewalk network indicators of connectivity and topography were found to have a significant impact on pedestrian safety [45]. Connectivity includes crosswalk coverage and sidewalk coverage, data of which were called from Sidewalk Centerline [30]. Marked crosswalks significantly reduce collisions and have been a common approach to protect pedestrians at intersections [27], and sidewalk coverage is negatively associated with pedestrian crashes [45]. Topography includes sidewalk width and sidewalk slope, data of which were called from Sidewalk Inventory [31]. Sidewalk widths that are narrower than 1.5 meters not only induce pedestrians to walk on roadways, which increases the risk of vehicular collision, but also fail to meet the worldwide minimum recommended width to accommodate wheelchair users [51]. Sidewalk slopes are negatively associated with vehicular-pedestrian crashes as they help reduce the speed of climbing cars [45], but their angles must not exceed the 2010 ADA guidelines.

3.2.3 Lighting Conditions

Lighting condition refers to the presence of lights provided by streetlamps during nighttime. As mentioned earlier, Ferenchak et al. (2022)'s work shows the negative associations between the existence of streetlamps and pedestrian crashes [15]. Although Analyze Boston's datasets do not detail the functionality of individual streetlamps, the presence of lighting may be inferred from longitudinal and latitudinal data given by Streetlight Locations [41]. Streetlight locations are used as explanatory variables for lighting conditions in this analysis.

3.2.4 Tree Canopy

Tree canopy coverage is the shade provided by street trees, which enhances the walking environment and overall pedestrian safety [53]. While Analyze Boston does provide tree canopy coverage data in terms of shade area, it does provide individual street tree locations in its BRPD Trees dataset [32]. Street tree location is used as an explanatory variable for tree canopy, as its presence is associated with shade to a certain extent. Location of trees as a variable is also a relatively more stable factor year-round than specific shading effectiveness of each individual tree, which has yet to be found mapped continuously with temporal and seasonal inconsistencies accounted for, as of our literature review.

3.2.5 Ramp Coverage

A pedestrian ramp refers to a curb ramp that connects a street or highway to a sidewalk. To ensure pedestrian accessibility of the route, the location of these ramps, as provided by Pedestrian Ramp Inventory [29], was taken into consideration primarily for analyzing accessibility. While ramps may have negative safety impacts if poorly designed (such as being too narrow or too steep), those that adhere to the Americans with Disabilities Act (ADA) standards may reduce that risk [43]. Ramp location is used as an explanatory variable for the presence of ADA-compliant ramps near pedestrian routes to analyze pedestrian accessibility in this study.

3.2.6 Sidewalk Damage

To ensure pedestrian accessibility of the route, sidewalk deficiencies were analyzed in accordance with the 2010 ADA Standards for Accessible Design [42]. Specifically, the sidewalk damage area over the sidewalk area was analyzed as an explanatory variable for both damaged sidewalk segment width and length.

3.3 Input Data Pre-processing and Feature Engineering

The data pre-processing step involved retrieving a random sample of 300,000 geographic points in the city of Boston from OpenStreetMap (OSM) with the OpenStreetMap NetworkX (OSMnx) walking network package. To bypass API call limit restrictions and minimize call times, the datasets relevant for model training were downloaded from the Analyze Boston website. For the Boston Vision Zero traffic accidents dataset specifically, we filtered for only pedestrian crashes, then filtered the dataset again with dispatch_ts to separate daytime and nighttime crashes that would be treated separately by the model.

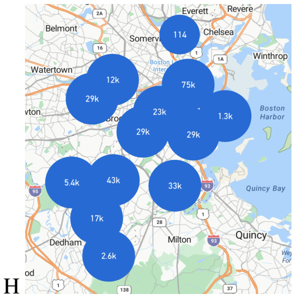


Fig. 1: The 300k sampled points around Boston

For each of the 300,000 sample points, we counted the number of discrete data types, such as ramps, trees, streetlamps, and crosswalks, within a tolerance distance of 10 meters. 10 meters was set as the tolerance distance to ensure reliability for detecting crashes, as it is about half the length of an average roadway width in residential and urbanized U.S. counties (55 feet) [35]. 10 meters was also set as the tolerance distance to account for the average GPS error for smartphones (16 feet) [38], so that analysis variables surrounding a given geospatial point may be accounted for with regard to GPS inaccuracies.

Within the same radius, we computed values for continuous data types such as speed limit, sidewalk width, slope, and damage. Invalid continuous data, such as empty sidewalk values or non-numeric values, were removed from the data.

Day-time and night-time crashes were stored in two different CSV files. Since relevant safety features differ across day and night conditions, we included different consideration factors in each. For the daytime condition, the variables considered included: speed limit, ramp, trees, crosswalk coverage, sidewalk width, sidewalk slope, and sidewalk damage. For the nighttime condition, we considered all of the above variables, as well as streetlights (which are only necessary for lighting at night). Any remaining rows with NaN values were removed from the CSV files.

A crash point is defined as a sample point where a crash took place within the tolerance radius. To ensure the model can distinguish between crash and no-crash scenarios, the data between crash and no-crash classes was balanced by under-sampling the class (crash or no crash) with the greater number of points and randomly deleting crash points until the two classes were balanced. Table 5:

Table 1: Distribution of crash and no-crash cases in the final dataset, stratified by time of day.

Time of Day	Crash	No Crash	Total
Day	20,905	20,905	41,810
Night	9,382	9,382	18,764

3.4 Methods

A binary classification is performed to obtain predictions of whether a vehicular-pedestrian crash is likely to occur on a geospatial location according to the compiled and preprocessed data. A site where at least one crash had occurred is classified as 1, and a site without crashes is classified as 0. A random forest classification model, implemented in Python 3.12.7 with scikit-learn 1.5.2 [47], is used to perform such binary classification using a sample point's associated safety factors. These binary classifications are then utilized to produce a confidence level of whether a given geospatial location is likely to have a crash through scikit-learn's predict_proba method (also implemented in Python 3.12.7 with scikit-learn 1.5.2 [47]). The probabilities are later used for the evaluation of pedestrian routes in the development of the NavigAid application.

3.4.1 Random Forest

Random forests are a supervised machine learning model proposed by statistician Leo Breiman and mainly used for classification and regression tasks [4]. The random forest fits unique subsets of a given dataset onto a collection of decision trees. Random forest classification models have a few advantages: first, they are able to capture non-linear relationships between input factors. Furthermore, random forests produce interpretable feature importance, which helps portray the underlying relationships between various features and results through the mean decrease in impurity (Gini importance) and feature permutation (where the model performance is evaluated after shuffling around each feature). Random forests are also useful for reducing variance, since they combine the results of multiple decision trees, each of which has been fitted on a bootstrap sample (a random sample drawn with replacement from the training dataset). Since each decision tree is only exposed to one random subset of the training data, the random forest trees are decoupled from each other.

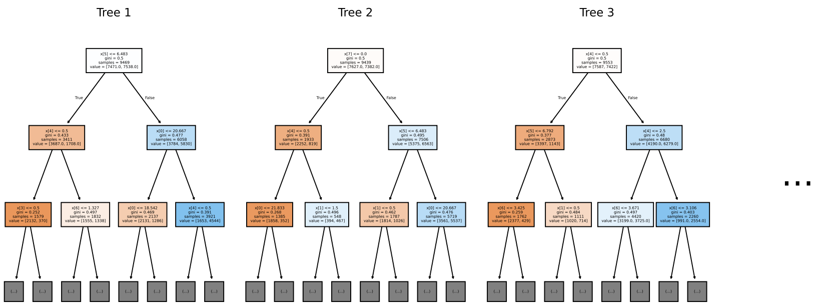

Tree 1
x[5] <= 6.483 gini = 0.5 samples = 9469 value = [7471.0, 7538.0]
True
False
x[4] <= 0.5 gini = 0.433 samples = 3411 value = [3687.0, 1708.0]
x[0] <= 20.667 gini = 0.477 samples = 6058 value = [3784, 5830]
x[3] <= 0.5 gini = 0.252 samples = 1579 value = [2132, 370]
x[6] <= 1.327 gini = 0.497 samples = 1832 value = [1555, 1338]
x[0] <= 18.542 gini = 0.469 samples = 2137 value = [2131, 1286]
x[4] <= 0.5 gini = 0.391 samples = 3921 value = [1653, 4544]
Tree 2
x[7] <= 0.0 gini = 0.5 samples = 9439 value = [7627.0, 7382.0]
True
False
x[4] <= 0.5 gini = 0.391 samples = 1933 value = [2252, 819]
x[5] <= 6.483 gini = 0.495 samples = 7506 value = [5375, 6563]
x[0] <= 21.833 gini = 0.268 samples = 1385 value = [1858, 352]
x[1] <= 1.5 gini = 0.496 samples = 548 value = [394, 467]
x[1] <= 0.5 gini = 0.462 samples = 1787 value = [1814, 1026]
x[0] <= 20.667 gini = 0.476 samples = 5719 value = [3561, 5537]
Tree 3
x[4] <= 0.5 gini = 0.5 samples = 9553 value = [7587, 7422]
True
False
x[5] <= 6.792 gini = 0.377 samples = 2873 value = [3397, 1143]
x[4] <= 2.5 gini = 0.48 samples = 6680 value = [4190.0, 6279.0]
x[6] <= 3.425 gini = 0.259 samples = 1762 value = [2377, 429]
x[1] <= 0.5 gini = 0.484 samples = 1111 value = [1020, 714]
x[6] <= 3.671 gini = 0.497 samples = 4420 value = [3199.0, 3725.0]
x[6] <= 3.106 gini = 0.403 samples = 2260 value = [991.0, 2554.0]
(...)
...

Fig. 2: An example of 3 decision trees, each showing 3 layers that make up the nighttime random forest model

For the classification task, each decision tree within the random forest outputs a single prediction based on the given input data. The resulting predictions from all decision trees are then combined through majority voting (or hard voting), which can be expressed as:

$$\hat{y} = \arg\max_{c\in C} \textstyle\sum_{b=1}^{B} 1\{h_b(x) = c\}$$

where h_b is the prediction for tree i, B is the total number of trees, and C is the set of classes. The resulting majority vote determines the classification of the input data, resulting in a classification that is more stable and reliable than that produced by any single decision tree.

3.4.2 Model Implementation

The processed crash data was split into two exclusive subsets: 80% of the crash data was used for model training, while the remaining 20% was reserved for hold-out testing data.

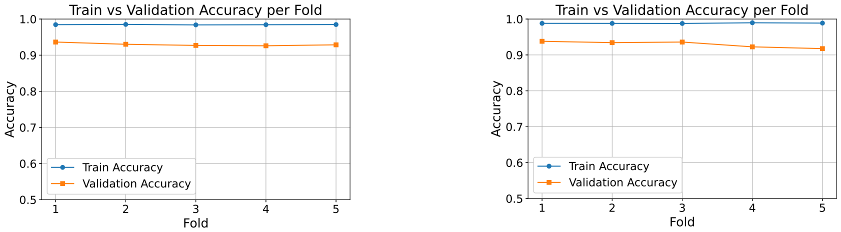
From the subset of training data, an additional 80% was split to train the base model, while the remaining 20% of the training data was reserved for model calibration. The separation of base model training data and calibration data prevents information leakage and pattern overfitting [5].

We used the RandomForestClassifier algorithm in scikit-learn with default parameter values, except for the n_estimators and random_state values. To improve model stability and generalization, we increased n_estimators (the number of decision trees) to 300. Random_state was set to 42 to ensure the reproducibility of results. Although increasing the complexity by increasing the number of decision trees may pose the risk of overfitting in other machine learning models, this does not apply for random forests due to the convergence to the expected value implied by the Strong Law of Large Numbers [4].

When splitting each node during the construction of trees in the random forest, RandomForestClassifier uses

$$\mathrm{m} = \sqrt{p}$$

where p is the total number of features, so that during tree construction, an exhaustive search for the best split is made using a random subset of size the square root of the number of features (speed limit, lighting, trees, etc.).



(a) Training and validation across folds for the day model.

(b) Training and validation across folds for the night model.

Fig. 3: Training and validation across folds for the daytime and night model.

Using the default values given by scikit-learn and the aforementioned modifications, the model achieved a high accuracy for training and validation when performing the 5-fold cross-validation. As shown in Figure 3, the model has a high training and validation accuracy, with an average training accuracy in the daytime model of 0.9845 and average validation of 0.9296, and average training accuracy in the nighttime model of 0.9883 and average validation of 0.9296. The consistently high accuracy values for training and validation for both daytime and nighttime models suggest that further hyper-tuning would not yield substantial improvements in results.

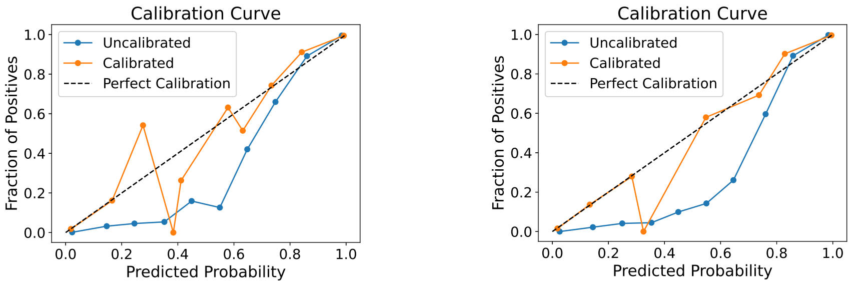
3.4.3 Model Calibration

Although the base model showed high accuracy in classification, it fell short of correctly estimating underlying probabilities for crash and no-crash classifications with scikit-learn's predict_proba method. Further calibration of the model was needed to ensure accurate predictions of underlying probabilities, which are then used as confidence levels for binary classifications.

An isotonic regression method from CalibratedClassifierCV was used to calibrate the model on the reserved calibration data, as the isotonic regression function has fewer restrictions and better performance compared to the sigmoid function (Platt's scaling), given that the sample size is greater than 1000. This method minimizes the Mean Squared Error (MSE) function:

$$\textstyle\sum_{i=0}^{n-1}(y_i - \hat{f}_i)^2$$

where $\hat{f}_i$ is the output of the classifier at sample i, $\hat{f}_i \geq \hat{f}_j$ whenever $f_i \geq f_j$, and y_i is the true label at sample i.[5] Since the model already performs cross-validation, we set the CV to "prefit" to prevent refitting.[1]



(a) Calibration curve for the daytime model

(b) calibration curve for the nighttime model

Fig. 4: Calibration curves for daytime and nighttime models

[1] As of scikit-learn version 1.6, the option CV=prefit has been deprecated. The recommended alternative is to use FrozenEstimator

As seen in Figure 4, the uncalibrated model consistently overestimates the probability of positive classifications. The calibrated model, on the other hand, produces a calibration curve closer to the perfectly calibrated line, indicating more accurate predicted probability estimates.

This result is supported by the Brier Score [16], which measures the mean squared difference between the predicted and actual outcome with the function

$$\frac{1}{N}\sum_{i=0}^{N-1}(y_i - \hat{p}_i)^2$$

where y_i is the target binary classification and $\hat{p}_i$ is the predicted probability. A Brier score closer to 0 demonstrates better calibration, since it indicates a smaller mean residual. The uncalibrated models have Brier scores of 0.0431 for the daytime model and 0.0439 for the nighttime model, while the calibrated model had significantly smaller scores of 0.0242 and 0.0280, respectively, indicating better estimates.

A Brier score can be decomposed into three parts: calibration (reliability), classification power (resolution), and randomness in the data (uncertainty) [36]. Table 2 and Table 3 show the Brier score decomposition before and after calibration for daytime and nighttime models. After model calibration, the reliability term of the Brier score is significantly closer to 0 for both daytime and nighttime models, while there is no significant change in the resolution and uncertainty terms. Hence, we can conclude that the change in Brier score is due to increased reliability in both calibrated models.

Table 2: Brier score decomposition before and after calibration for the day model.

	Reliability	**Resolution**	**Uncertainty**
Uncalibrated	0.0144	0.2216	0.2500
Calibrated	0.0005	0.2222	0.2500

Table 3: Brier score decomposition before and after calibration for the night model.

	Reliability	**Resolution**	**Uncertainty**
Uncalibrated	0.0189	0.2252	0.2500
Calibrated	0.0002	0.2257	0.2500

4 Results

Given a geospatial location within Boston and associated street-level variables, the calibrated model is able to accurately classify the occurrence of vehicular-pedestrian crashes and estimate the probability of such crashes. The model was tested on the 20% hold-out data to evaluate the model's accuracy and precision on unseen data. We use confusion matrices, receiver operating characteristics, precision-recall curves, and predicted probability distributions to provide a comprehensive report on the model's high precision rate and confidence level in its output. We also used impurity-based importance measures, permutation importance measures, and beeswarm summary plots to demonstrate the impact of individual analysis variables on the model results. Our results align with the results of past studies regarding a strong positive correlation between estimated impact speed (approximated by roadway speed limits) and the likelihood of crashes, as well as weaker, positive correlation between lighting conditions (approximated by streetlight locations) and pedestrian safety, and tree canopy coverage (approximated by street tree locations) and pedestrian safety.

For data analysis and visualization, we used the following Python libraries: Pandas version: 2.2.3 [46], NumPy version: 1.26.4 [17], Shap version: 0.48.0 [26], Matplotlib version: 3.9.2 [20], and Seaborn version: 0.13.2 [52].

4.1 Model Evaluation

For the final model and to ensure that there was no leaky data between folds during cross-validation, we tested the model's accuracy on the reserved testing data. Even on the testing data, the model showed high accuracies of 0.9646 and 0.9688 for daytime and nighttime, respectively. This further verifies that the model is not memorizing noise but generalizing patterns to make accurate predictions.

However, the accuracy rate itself does not measure the final model's ability to distinguish false positives and false negatives. Thus, the precision, recall, and F1-scores of the daytime and nighttime models were also used to provide a more comprehensive report of each model's performance.

Table 4: Classification report for the daytime model.

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.9635	0.9658	0.9646	4181
1	0.9657	0.9634	0.9646	4181

Table 5: Classification report for the nighttime model.

Class	**Precision**	**Recall**	**F1-score**	**Support**
0	0.9661	0.9718	0.9689	1877
1	0.9716	0.9659	0.9687	1876

The first performance metric, precision, is a measure of the predicted crashes that are correct, given by the equation:

$$\frac{TP}{TP+FP}$$

where TP is true positive, TN is true negative, FP is false positive, and FN is false negative.

Both the model's high precision indicates that it is not making false positive predictions, which may misinform pedestrians and lower their sense of perceived safety when choosing or using a route.

The second, recall, is a measure of the sensitivity of the model, meaning the proportion of actual crash cases that are correctly identified, given by the equation:

$$\frac{TP}{TP+FN}$$

Where TP is true positive, TN is true negative, FP is false positive, and FN is false negative. Both the model's high recall number indicates its ability to detect a crash when there was one, which minimizes false negatives. This high score is crucial in the context of crash detection, as missing a true crash could result in serious safety implications.

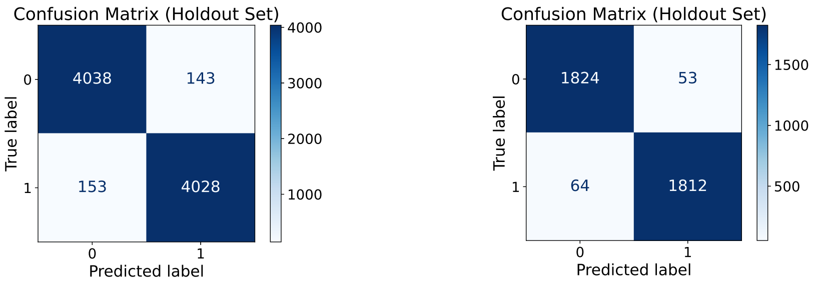
The third, F1-score, is a harmonic mean of the precision and recall given by the equation:

$$s \cdot \frac{Precision \cdot Recall}{Precision+Recall}$$

Both the model's high F1-score indicates that the model is accurately identifying crashes while minimizing false positives and false negatives.

4.1.1 Confusion Matrix

The model's performance can be further illustrated through the confusion matrix.



(a) Confusion matrix daytime model

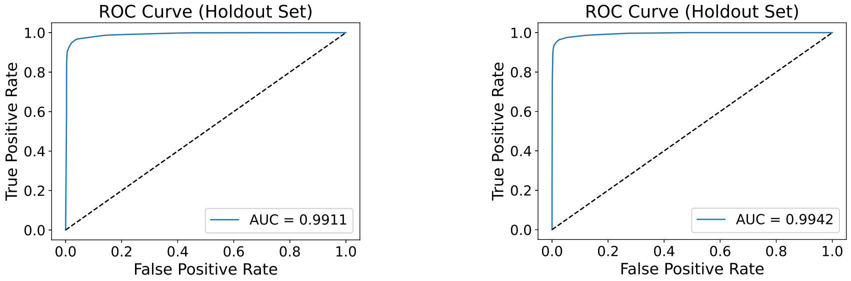
(b) Confusion matrix nighttime model

Fig. 5: Confusion for the nighttime and daytime models.

As seen in Figure 5, both models have high counts for true positives and true negatives, and very low counts for false positives and false negatives. This distribution outlines the reliability of the model in the context of crash prediction: it accurately distinguishes between the occurrence of a crash and no crashes in a given site.

4.1.2 Receiver Operating Characteristics

Another important evaluation concern is the model's ability to differentiate the two classes across different thresholds (independent of where the cutoff point is between the two classes). This can be measured by the Receiver Operating Characteristic (ROC) curve. This metric is an evaluation of the probability of the model ranking a randomly chosen positive example over a randomly chosen negative example (i.e., how well the model can order the predictions rather than looking at a single threshold for a decision)[14]. An AUC ROC score of 0.5 indicates a random model, while a score of 1 indicates a perfect model.



(a) AUC ROC for daytime model

(b) AUC ROC for nighttime model

Fig. 6: AUC ROC for the nighttime and daytime models.

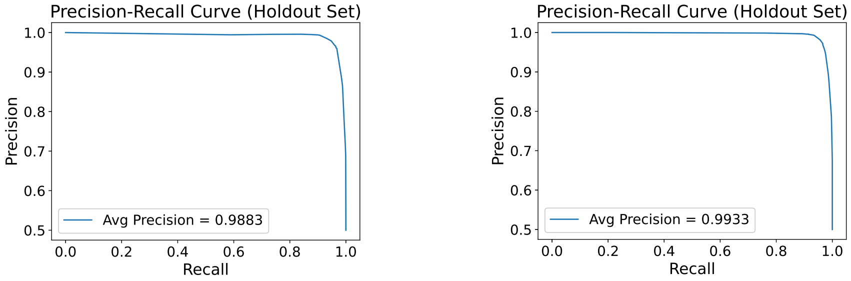
As the figures show, the Area Under the Curve for both nighttime and daytime models is over 0.99, demonstrating each model's near-perfect ability to classify crash and no crash across all classification thresholds. Additionally, this high AUC also suggests that the model consistently ranks positive examples (crashes) higher than negative examples (no crashes), reflecting both good sensitivity and specificity, as well as robustness across decision thresholds.

4.1.3 Precision-Recall Curve

The Precision-Recall curve shows the tradeoff between precision and recall across different thresholds. The plot can be summarized by the average precision given by the equation:

$$\sum_n (R_n - R_{n-1}) P_n$$

where P_n and R_n are the precision and recall at the nth threshold.



(a) Precision-Recall for daytime model

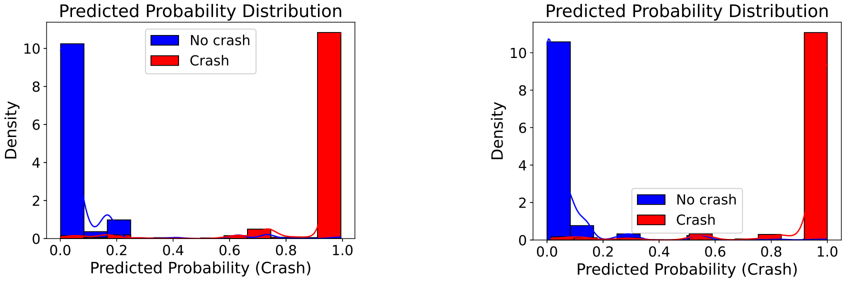
(b) Precision-Recall for nighttime model

Fig. 7: Precision-Recall Curve for daytime and nighttime model

The model's high area average precisions of 0.988 and 0.993 represent both high precision and high recall on the balanced hold-out data, reinforcing both our classification report and our ROC curves.

4.1.4 Predicted Probability Distribution

Furthermore, the predicted probability distribution provides insight into how confident the model is in making predictions, in which borderline predictions, such as 0.5, reveal weak confidence, while predictions at the ends, such as 0 or 1, demonstrate strong confidence.



(a) PPD for daytime model

(b) PPD for nighttime model

Fig. 8: Predicted Probability Distribution (PPD) for daytime and nighttime model

As the figures display, both the daytime and nighttime models have peaks at 0 and 1 with minimal overlap between the red and blue bars, illustrating our models' ability to make good separations between classes as well as their confidence in their predictions.

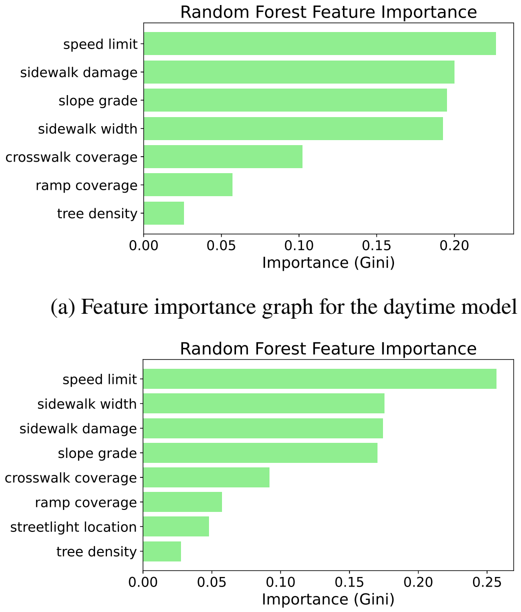
4.2 Feature Importance

A basic overview of each feature's importance can be seen through the impurity-based measures, specifically Gini impurity. An impurity-based measure calculates the random forest model's feature contributions between splits in its decision trees by analyzing how each feature reduces the Gini impurity.

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y_i = k) \text{ for } k = \{0, 1\}$$

$$\text{Gini Impurity} = \sum_k p_{mk}(1 - p_{mk})$$

This measure is built into the random forest classifier implementation in scikit learn as the default criterion [5].



(a) Feature importance graph for the daytime model

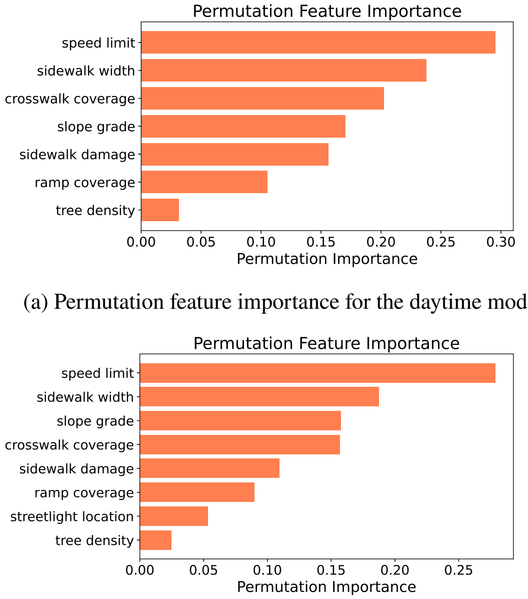
(b) Feature importance graph for the nighttime model

Fig. 9: Feature importance graphs

The impurity-based feature importance graph for both models show that roadway speed limit (explanatory variable for estimated impact speed) has the strongest safety impact, while street light locations (explanatory variable for lighting conditions) and tree location (explanatory variable for tree canopy) has the least safety impact, which agrees with recent studies by Hussain et al. (2019) [21], Ferenchak et al. (2022) [15], and Zhu et al. (2022) [53], as covered in our literature review in 3.2 Analysis Variables.

4.2.1 Permutation Importance

Although impurity-based feature importance may provide an overview of the relative importance of each analysis variable, it does not necessarily reflect predictions and importance on a test set, as it is tightly coupled to training statistics and favors high cardinality features [5]. To address this, permutation importance was used to evaluate the impact on performance when each of the feature values is shuffled.



(a) Permutation feature importance for the daytime model

(b) Permutation feature importance for the nighttime model

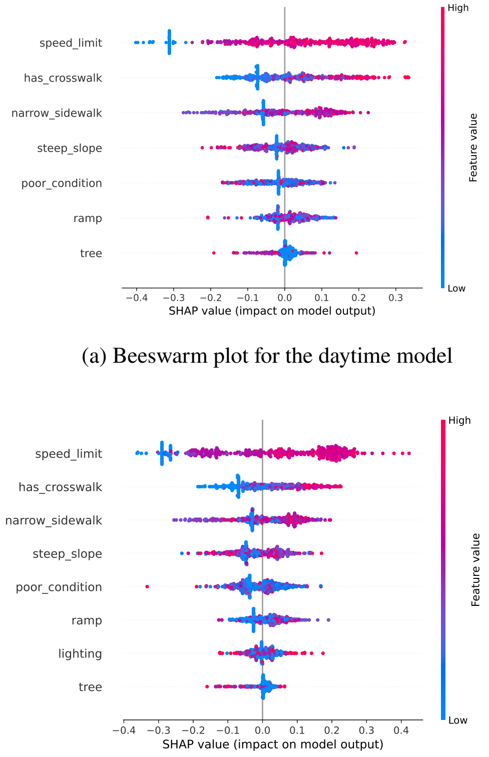
Fig. 10: Permutation feature importance for daytime and nighttime models

Permutation importance ranks the importance by how much a model's performance drops when a specific feature is shuffled around. A greater drop in model performance caused by a shuffle in feature data reveals the feature's greater importance. This metric is independent of the specific model itself, as it runs on the holdout data instead of the training data, providing a better overview of each feature's true contribution to predictions on unseen data.

Compared to the Gini feature importance plots, the permutation plots show consistent rankings for speed limit, ramp locations, and street tree locations, but vary slightly in their rankings for middle values. For example, poor street conditions are ranked second in the Gini feature importance graph for the daytime model but fifth in the permutation importance graph. This suggests an underlying relationship between the different features in the model due to overlapping data.

4.2.2 Beeswarm Summary Plot

Additionally, a beeswarm summary plot[26] was used to demonstrate which features contributed positively or negatively to a crash prediction.



(a) Beeswarm plot for the daytime model

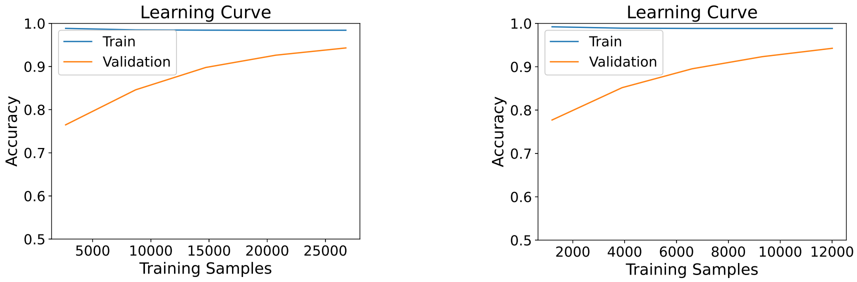
(b) Bee swarm plot for the nighttime model

Fig. 11: Beeswarm plot for daytime and nighttime models

The correspondence of red dots (indicating high feature value) with a high positive SHAP value on the x-axis for speed limit demonstrates how it contributes positively to crash predictions. This agrees with Hussain et al. (2019)'s study on the positive association between speed limits and crash risk [21]. The same correspondence appears for crosswalk coverage as well, suggesting that it contributes positively to crash predictions. This may be because an increase in crosswalks increases traffic exposure to pedestrians, an explanation parallel to Cai et al. (2016)'s study that analyzed sidewalk length as a traffic exposure for pedestrian safety [6]. Other features have a more mixed scattering of colored plots with no clear pattern of direction, which suggests that they are more context-dependent or codependent with each other.

4.3 Validation

Although random forests are known to have low variance and are resistant to overfitting, the model may be at risk of overfitting in cases where training data is insufficient or the model is highly complex. We can check for overfitting by comparing the training and validation learning curves.



(a) Learning curve for the daytime model

(b) Learning curve for the night model

Fig. 12: Learning curves for daytime and nighttime models

The learning curves show that as the training sample size increases, the validation accuracy approaches the training accuracy (Fig. 12). This indicates minimal overfitting and variance, suggesting that the model performs well due to generalization rather than memorization. However, the validation curve has not yet plateaued, suggesting the model may be further improved by training on a larger dataset.

To further verify the model's performance and generalization abilities, a shuffled data check was performed, training the model on shuffled training data to scramble the relationships between training features and outcome. The model was then tested on unshuffled hold-out testing data.

The model trained on the shuffled training data showed an accuracy rate of around 0.5, 0.4636 for the daytime model, and 0.5481 for the nighttime model, indicating that it performed classifications at random. We can conclude that the performance of the original model was the result of learning meaningful patterns, rather than memorization.

5 Conclusion

The framework presented in this study for analyzing pedestrian crash risk accounts for not only pedestrian infrastructure, but also street-level features along a given route, to give reliable and precise information about pedestrian safety. Using public datasets from the city of Boston, the random forest model considers various explanatory variables linked to pedestrian crashes and fatality, such as estimated impact speed, sidewalk network connectivity, lighting conditions, tree canopy coverage, ramp coverage, and sidewalk damage. Evaluation of the random forest model showed that an estimated impact speed of 25 mph or higher has a strong positive correlation with vehicular-pedestrian crashes. Factors such as lighting conditions and tree canopy have a weaker positive correlation with crash risk. These findings align with results from previous crash analysis studies. We then incorporated the model into a pedestrian-oriented mobile navigation app, which allows users to determine the safest walking routes between their desired locations. The proposed framework may be useful for guiding pedestrian-accessible urban planning.

For app development, we used Expo version 53.0.20[13], React version 19.1.1[33], React-native version 0.81.0[34], React-native-maps version 1.25.3[2], Turf.js version 3.0.14[9], Joblib version 1.4.2[10], FastAPI version 0.116.1[7], and pydantic version 2.11.7[8].

5.1 *NavigAid: Navigation Application*

We developed a pedestrian-centered navigation app called NavigAid, which uses the model to help users determine the safest walking routes between two locations. The OpenRouteService API [18] was used to generate possible routes between user-inputted starting points and endpoints. The OpenRouteService API uses OpenStreetMap (OSM), an open-source, crowd-sourced mapping service that utilizes street networks, built environments, and geographic information from aerial imagery, GPS, field maps, commercial, and government data. OSM provides geographic mapping data that is continuously updated and verified by a community of contributors with local knowledge [18]. Previous studies suggest that the geographic, street network, and built environment data from OpenStreetMap have reliable coverage across many cities, including Boston, the city we are focusing on for the model[44].

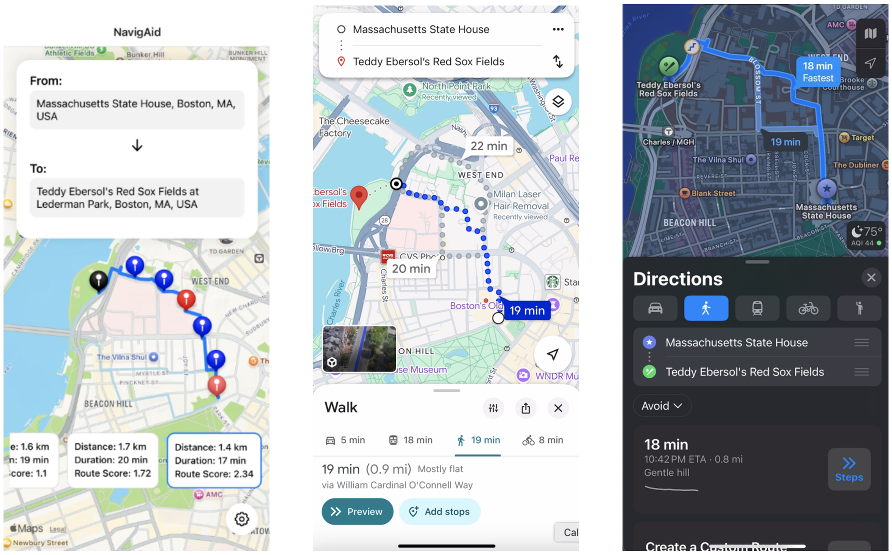
OpenRouteService API's Direction V2 Endpoint was used to generate three alternate paths between a user-chosen starting point and endpoint. The generated paths are displayed by React Maps with three map types (terrain, satellite, and hybrid) to choose from. To improve efficiency, Turf.js' buffer function was used to filter datasets, ensuring that only explanatory variable data within 10 meters of the generated routes was processed. Boston's ArcGIS REST API was queried to return the explanatory variable data within the buffer bounding box already produced, and this data was then processed to yield the relevant feature information:

Feature	**Source**	**Method / Flagging Criteria**
WIDTH	Sidewalk Inventory [31]	Extract `SWK_WIDTH`; flag sidewalks narrower than 5 ft
SLOPE	Sidewalk Inventory [31]	Extract `SWK_SLOPE`; flag slopes steeper than 5%
DAMAGE	Sidewalk Inventory [31]	Compute `DAM_AREA / SWK_AREA`; flag damage > 0.25
LIGHTING	Streetlight Locations [41]	Count street lamps within 10 m; flag points with zero lamps
TREE	BPRD Trees [32]	Count trees within 10 m; flag points with zero trees
RAMP	Pedestrian Ramp Inventory [29]	Count ramps within 10 m; flag points with zero ramps
SPEED	Sidewalk Centerline [30]	Extract `SPEEDLIMIT`; flag limits > 25 mph

Table 6: Overview of sidewalk and roadway features, data sources, and flagging criteria.

This data was passed to the backend, where the calibrated random forest binary classification model was stored. Although the model produces discrete class predictions, we used the predict_proba method to generate the probability of vehicular-pedestrian crashes at each point along the three generated routes. The route safety score is an aggregated measure of the probabilities along a path, which preserves the consideration of the traffic exposure along the path distance. The safety score of each path is then sent to the app's frontend, where the score is displayed in a route summary along with path distance and duration. When each path is selected, color-coded flags with information about street-level conditions are displayed.

Compared to other navigation tools, NavigAid provides an array of detailed, pedestrian-oriented route information (Fig. 13). Relevant street-level features appear on the route as flags, color-coded by feature type; users are able to use a filter function to display features that are relevant to their concerns. All three navigation tools show basic route information such as a map display, route distance, and estimated duration. However, NavigAid is the only tool that provides a route score, a measure indicating pedestrian crash risk as determined by the random forest model, and ranks routes based on the lowest (i.e., safest) route score. These features are important to address pedestrian users' route safety concerns and specific accessibility needs.



NavigAid

From:

Massachusetts State House, Boston, MA, USA

To:

Teddy Ebersol's Red Sox Fields at Lederman Park, Boston, MA, USA

e: 1.6 km
n: 19 min
core: 1.1

Distance: 1.7 km
Duration: 20 min
Route Score: 1.72

Distance: 1.4 km
Duration: 17 min
Route Score: 2.34

Massachusetts State House

Teddy Ebersol's Red Sox Fields

Walk

5 min · 18 min · 19 min · 8 min

19 min (0.9 mi) Mostly flat
via William Cardinal O'Connell Way

Preview · Add stops

Directions

Massachusetts State House

Teddy Ebersol's Red Sox Fields

Avoid

18 min
10:42 PM ETA · 0.8 mi
Gentle hill

Steps

Create a Custom Route

(a) An example route generated by NavigAid

(b) An example route generated by Google Maps

(c) An example route generated by Apple Maps

Fig. 13: A side-by-side comparison of the routes generated and ranked by our application and other popular navigation applications

5.2 *Concerns and Next Steps*

While our model was able to make highly accurate and precise pedestrian safety predictions, a few caveats must be addressed to interpret the results. We used public datasets provided by the city of Boston for efficiency, ease of access, and ease of generalization to other cities. However, the data provided by Analyze Boston is not real-time data, so there may be differences between the actual conditions of a given pedestrian route and the details given by the dataset. For instance, the data used for roadway speed limits was last updated in November 2024, but the actual speed limits may have changed since then. Future research may consider using a wider variety of data sources, including crowd-sourced datasets, independent observations, and satellite imagery for data validation.

Another limitation of our approach is our feature evaluation tolerance distance. We computed values for explanatory variables within a tolerance distance of 10 meters, which does not account for actual spatial effects that may potentially impact the significance of each variable [24]. We chose 10 meters as a conservative tolerance distance that is reasonable for early development and testing. Further research may consider investigating tolerance distances specific to each explanatory variable, such as using a tighter threshold for street tree locations while keeping a wider radius for speed limits. Future research could also explore dynamically adjusting the tolerance radius based on current street conditions, which can be analyzed using roadway design standards.

We also note that although random forests are capable of capturing non-linear relationships, the feature importance measures we present mainly focus on linear relationships. Future research can explore feature contribution over different values with dependence and partial-dependence plots to investigate possible codependent relationships between explanatory variables. Prior studies have reviewed the spatial effects of different ramps on vehicular-pedestrian crashes, but there remains a literature gap on the non-linear relationships between inventoried pedestrian-accessible infrastructure and crash risk.

Furthermore, NavigAid relies on the preexisting OpenRouteService API to obtain possible routes, and the model then ranks the routes according to their generated route scores. Future studies may consider developing a route-generating model that relies on pedestrian-centerline data for more accurate pedestrian route generation. Hosseini et al. (2023)'s Tile2Net model [19] mentioned in section 2.2 Framework for Measuring Pedestrian Accessibility and Dijkstra's algorithm [22] may be incorporated in future work to find shortest paths given the premise of pedestrian safety.

Finally, we bring up some ethical concerns pertaining to NavigAid. One potential risk of labeling certain routes or areas as unsafe is that this will drive continued stigma around the newly invented no-go area. We chose not to include crime data in our pedestrian safety model to mitigate the risk of perpetuating patterns of racialized social sorting and underinvestment in stigmatized neighborhoods. Even so, classifying certain routes as relatively less safe than others and routing users around these paths might pose the risk of resulting in rising insurance premiums, drops in real estate prices, and shop profits in the region [3]. Due to discriminatory policies such as redlining, many low-income communities of color face underinvestment in pedestrian-accessible infrastructure, among other structural needs. Due to this history of structural underinvestment, our app might disproportionately label paths in such regions as less safe, which may affirm racist perceptions of certain neighborhoods.

Furthermore, our app is designed to mitigate pedestrian injury and fatality risks through changing personal behaviors. This may perpetuate a victim-blaming culture that targets pedestrians and shifts the burden of responsibility for traffic accidents away from policymakers, urban designers, and motorists onto individual pedestrians. It should be made clear that our proposed model was developed as an intervention, rather than a solution for the root issue of pedestrian safety and accessibility in car-centric environments. We hope that our model can be used to inform and encourage equitable public investment in improving pedestrian-accessible infrastructure.

Code Availability

The code for the model and the app is publicly available in the following GitHub repositories:

- Model: https://github.com/Rayna-Yu/NavigAid-model/tree/master
- App: https://github.com/Rayna-Yu/NavigAid

Acknowledgements The work was supported by the AltREU program at Portland State University. The authors would like to thank Dr. Christof Teuscher and other Portland State University faculty mentors for their guidance and support throughout the project.

References

1. Dzifa Adjaye-Gbewonyo and Elizabeth M. Briones. Walking for leisure and transportation among adults: United states, 2022. Technical report, National Center for Health Statistics, 2024.
2. Inc. Airbnb. React native maps. https://github.com/react-native-maps/react-native-maps, 2025. Version 1.25.3.
3. Frederik Zuiderveen Borgesius, Jonathan Gray, and Mireille van Eechoud. Open data, privacy, and fair information principles: Towards a balancing framework. *Berkeley Technology Law Journal*, 30(3):2073–2131, 2015.
4. Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
5. Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.
6. Qing Cai, Jaeyoung Lee, Naveen Eluru, and Mohamed Abdel-Aty. Macro-level pedestrian and bicycle crash analysis: Incorporating spatial spillover effects in dual state count models. *Accident Analysis Prevention*, 93:14–22, 2016.
7. FastAPI Contributors. Fastapi. https://fastapi.tiangolo.com/, 2025. Version 0.116.1.
8. Pydantic Contributors. Pydantic. https://docs.pydantic.dev/, 2025. Version 2.11.7.
9. Turf Contributors. Turf.js. https://turfjs.org/, 2025. Version 3.0.14.
10. Joblib Developers. Joblib. https://joblib.readthedocs.io/, 2025. Version 1.4.2.
11. Rune Elvik, Peter Christensen, and Astrid Helene Amundsen. *Speed and road accidents: an evaluation of the Power Model*. PhD thesis, Transportøkonomisk Institutt, 2004.
12. ArcGIS Enterprise. urlhttps://gisportal.boston.gov/arcgis/help/en/rest/services-reference/enterprise/get-started-with-the-services-directory/. 2025-08-17.
13. Expo. Expo. https://expo.dev/, 2025. Version 53.0.20.
14. Tom Fawcett. Introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 06 2006.
15. Nicholas N. Ferenchak, Risa E. Guitierrez, and Patrick A. Singleton. Shedding light on the pedestrian safety crisis: An analysis across the injury severity spectrum by lighting condition. *Traffic Injury Prevention*, 23:434–439, 2022.
16. W Brier Glenn et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.
17. Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
18. HeiGIT / Heidelberg Institute for Geoinformation Technology. Openrouteservice api. https://openrouteservice.org, 2025. Map data © OpenStreetMap contributors.

19. Maryam Hosseini, Andres Sevtsuk, Fabio Miranda, Roberto M. Cesar, and Claudio T. Silva. Mapping the walk: A scalable computer vision approach for generating sidewalk network datasets from aerial imagery. *Computers, Environment and Urban Systems*, 101:101950, 2023.
20. J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
21. Qinaat Hussain, Hanqin Feng, Raphael Grzebieta, Tom Brijs, and Jake Olivier. The relationship between impact speed and the probability of pedestrian fatality during a vehicle-pedestrian crash: A systematic review and meta-analysis. *Accident Analysis & Prevention*, 129:241–249, 2019.
22. Adeel Javaid. Understanding dijkstra's algorithm. *Available at SSRN 2340905*, 2013.
23. Yong jin Joo and Soo-Ho Kim. A new route guidance method considering pedestrian level of service using multi-criteria decision making technique. *Journal of Korea Spatial Information Safety*, 19, 2011.
24. M. Ahsanul Karim, Mohamed M. Wahba, and Tarek Sayed. Spatial effects on zone-level collision prediction models. *Transportation Research Record*, 2398(1):50–59, 2013.
25. Shiqin Liu, Carl Higgs, Jonathan Arundel, Geoff Boeing, and Nicholas Cerdera. A generalized framework for measuring pedestrian accessibility around the world using open data. *Geographical Analysis*, 54:559–582, 2022.
26. Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, 2(1):2522–5839, 2020.
27. Moran M. E. Where the crosswalk ends: Mapping crosswalk coverage via satellite imagery in san francisco. *Environment and Planning B: Urban Analytics*, 49:2250–2266, 2022.
28. Boston Maps. Boston street segments (sam system) -dataset- analyze boston, 2015.
29. Boston Maps. Pedestrian ramp inventory -dataset- analyze boston, 2015.
30. Boston Maps. Sidewalk centerline -dataset- analyze boston, 2015.
31. Boston Maps. Sidewalk inventory -dataset- analyze boston, 2015.
32. Boston Maps. Bprd trees -dataset- analyze boston, 2025.
33. Inc. Meta Platforms. React. https://react.dev/, 2025. Version 19.1.1.
34. Inc. Meta Platforms. React native. https://reactnative.dev/, 2025. Version 0.81.0.
35. Adam Millard-Ball. The width and value of residential streets. *Journal of the American Planning Association*, 88:1–14, 05 2021.
36. Allan H Murphy. A new vector partition of the probability score. *Journal of Applied Meteorology and Climatology*, 12(4):595–600, 1973.
37. Basu N., Oviedo-Trespalacios O., King M., Kamruzzaman Md., and Haque Md. M. What do pedestrians consider when choosing a route? the role of safety, security, and attractiveness perceptions and the built environment during day and night walking. *Cities*, 143:104551, 2023.
38. Navigation National Executive Committee for Space-Basaed Positioning and Timing. Gps accuracy, 2022.
39. Peter D. Norton. Street rivals: Jaywalking and the innovation of the motor age street. *Technology and Culture*, 48:331–359, 2007.
40. Department of Innovation and Technology. Vision zero crash recrods -dataset- analyze boston, 2015.
41. Department of Innovation and Technology. Streetlight locations -dataset- analyze boston, 2016.
42. U.S. Department of Justice: Civil Rights Division. 2010 ada standards for accessible design, 2010.
43. U.S. Department of Transportation: Federal Highway Administration. Federal highway administration university course on bicycle and pedestrian transportation, 2006.
44. Kazi Omar, Gustavo Moreira, Daniel Hodczak, Maryam Hosseini, and Fabio Miranda. Crowdsourcing and sidewalk data: A preliminary study on the trustworthiness of openstreetmap data in the us, 10 2022.

45. Ahmed Osama and Tarek Sayed. Evaluating the impact of connectivity, continuity, and topography of sidewalk network on pedestrian safety. *Accident Analysis & Prevention*, 107:117–125, 2017.
46. The pandas development team. pandas-dev/pandas: Pandas, February 2020.
47. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
48. Mathew A. Raifman and Choma F. Ernani. Disparities in activity and traffic fatalities by race/ethnicity. *American Journal of Preventive Medicine*, 63:160–167, 2022.
49. Jonathan Stiles and Harvey Miller. The built environment and the determination of fault in urban pedestrian crashes: Towards a systems-oriented crash investigation. *Journal of Transport and Land Use*, 17:97–113, 2024.
50. Nandi L. Taylor, Jamila M. Porter, Shenee Bryan, Katherine J. Harmon, and Laura S. Sandt. Structural racism and pedestrian safety: Measuring the association between historical redlining and contemporary pedestrian fatalities across the united states, 2010-2019. *American Journal of Public Health*, 113:420–428, 2023.
51. Maria Vittoria, Paola Di Mascio, and Laura Moretti. Managing sidewalk pavement maintenance: A case study to increase pedestrian safety. *Journal of Traffic and Transportation Engineering (English Edition)*, 3:203–214, 2016.
52. Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.
53. Manman Zhu, N. N. Sze, and Sharon Newman. Effect of urban street trees on pedestrian safety: A micro-level pedestrian casualty model using multivariate bayesian spatial approach. *Accident Analysis & Prevention*, 176:106818, 2022.