



Predicting vehicle collisions for the area of Seattle

Introduction: Business Problem

Background

The Seattle Department of Transportation's annual traffic report poses a significant challenge to governments to review their road safety policies and take new initiatives that make drivers' lives safer.

Problem Statement

Given this scenario, the objective of this project is to provide the best classification model to predict the type of collision (an auto accident involving injury or property damage) based on the weather and road conditions during the collision, the geographical area where the collision took place, the type of the collision and the day of the week that the collision occurred.

Interest

The information produced through this model may serve as an instrument for decision support in the prioritization of actions, such as changing road infrastructure, the procurement and installation of the road signs to reflect the weather conditions. Moreover, these informations could alert drivers, given the weather and road conditions about the possibility of a car accident and the severity of it, which would allow the driver to drive more carefully or, even change his itinerary.

Data sources

- The data used corresponds to the Seattle Department of Transportation's annual Traffic Report Dataset.
- This data was collected by the during the period of 2004-2020 and contains such information as severity, location, collision type, weather conditions, road conditions, and light conditions, among others.

For further details about the data set used, see

https://www.seattle.gov/Documents/Departments/SDOT/GIS/Collisions_OD.pdf

Data cleaning

- As part of the cleaning process the data was downloaded and extracted, the whole dataset was loaded and converted into a pandas dataframe, allowing for a better use of the data set to get a sense of the data structure, missing and bad data, and what kind of data processing cleaning should take place.
- So, the data set consists of 194,673 observations and 38 features.
- The method “seaborn.heatmap” was used to find the missing values. The columns with more than 90% of missing values were removed from dataset.
- After data cleaning, there were 194,673 samples and 35 features in the data.

Feature selection

- After this verification, the variables given below were used for machine learning purposes, where the target variable was the column SEVERITYCODE:

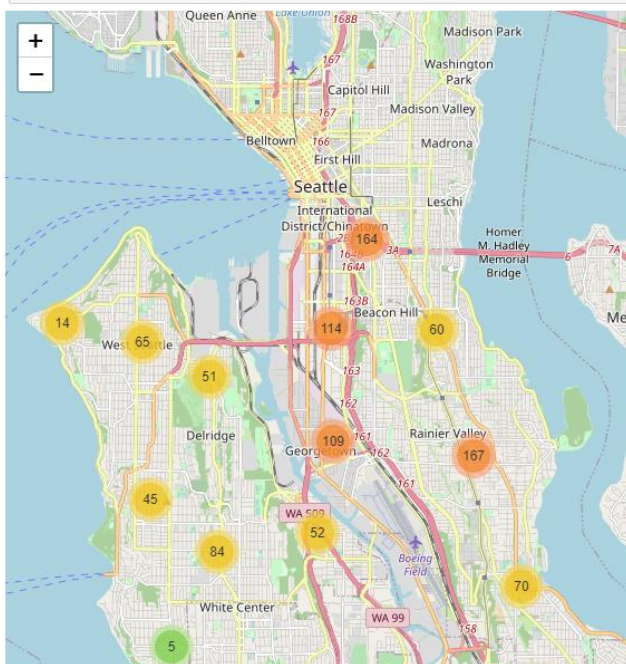
Data Set Basics	
SEVERITYCODE	A code that corresponds collision: • 2—injury, • 1—prop damage
SEVERITYDESC	Description collision: Injury Collision, Property Damage Only Collision.
WEATHER	A description of the weather conditions during the time of the collision: Clear, Severe Crosswind, Sleet/Hail/Freezing Rain, Snowing, Overcast, Raining.
ROADCOND	The condition of the road during the collision: Wet, Dry, Ice, Standing Water, Sand/Mud/Dirt, Oil, Snow/Slush.
LOCATION	The geographical area where the collision took place.
INCDATE	The date of the incident.
ADDRTYPE	Collision address type: • Block • Intersection

Feature selection

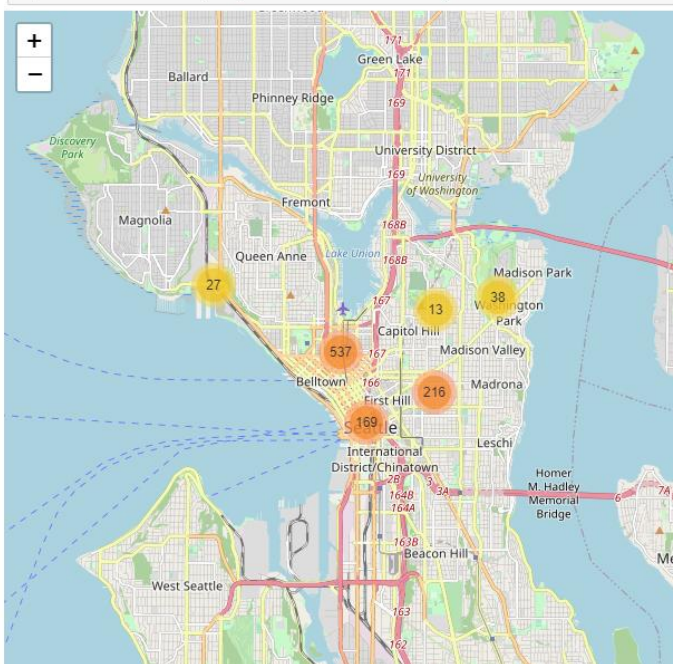
- In addition to these, a new column was created “DISTRICT”, grouping car accidents by area. For this, columns "longitude" and "latitude" were used, dividing Seattle into four districts.

- District 1: from latitude 47.4 to latitude 47.5.
- District 2: the latitudes 47.60, 47.61, 47.62 and 47.63.

map1 # DISTRICT 1: sampling the data to select only 1000 vehicle collisions



map2 # DISTRICT 2: sampling the data to select only 1000 vehicle collisions

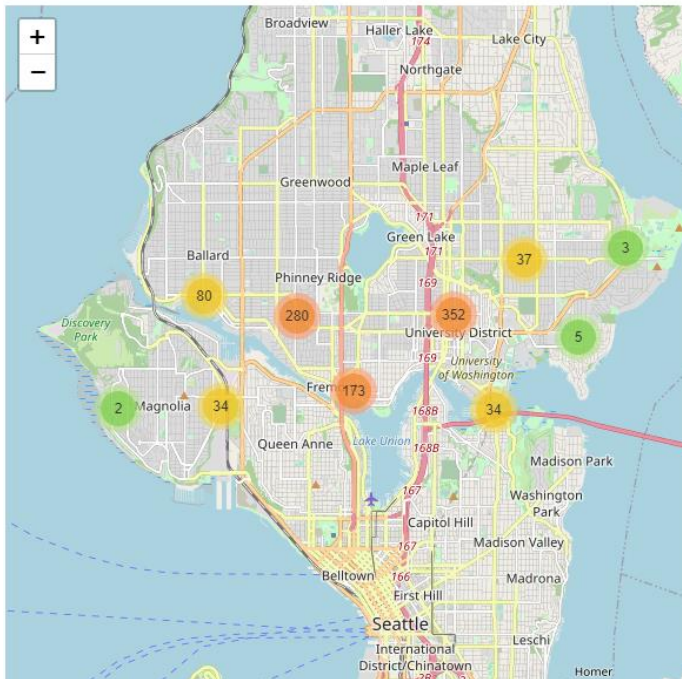


Feature selection

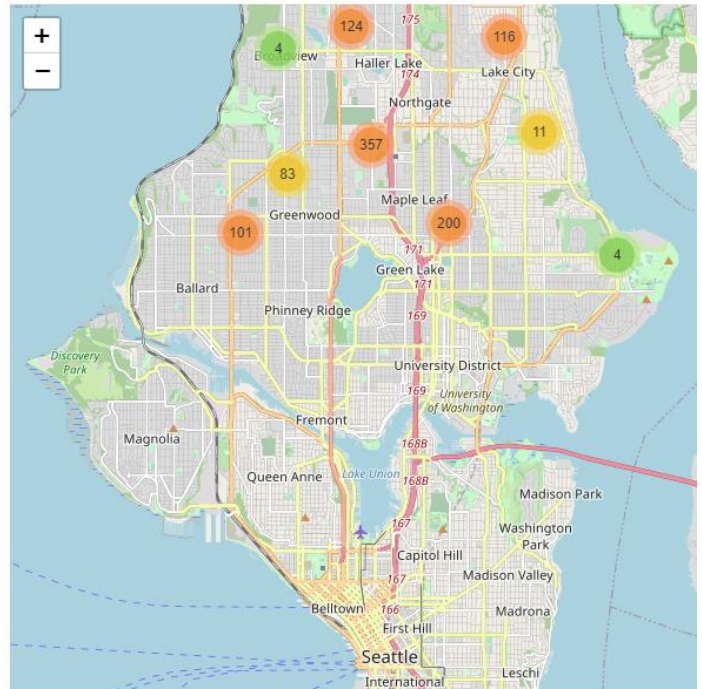
- In addition to these, a new column was created “DISTRICT”, grouping car accidents by area. For this, columns "longitude" and "latitude" were used, dividing Seattle into four districts.

- District 3: the latitudes 47.64, 47.65, 47.66 and 47.67.
- District 4: the latitudes 47.68, 47.69 and 47.7.

map3 # DISTRICT 3: sampling the data to select only 1000 vehicle collisions



map4 # DISTRICT 4: sampling the data to select only 1000 vehicle collisions

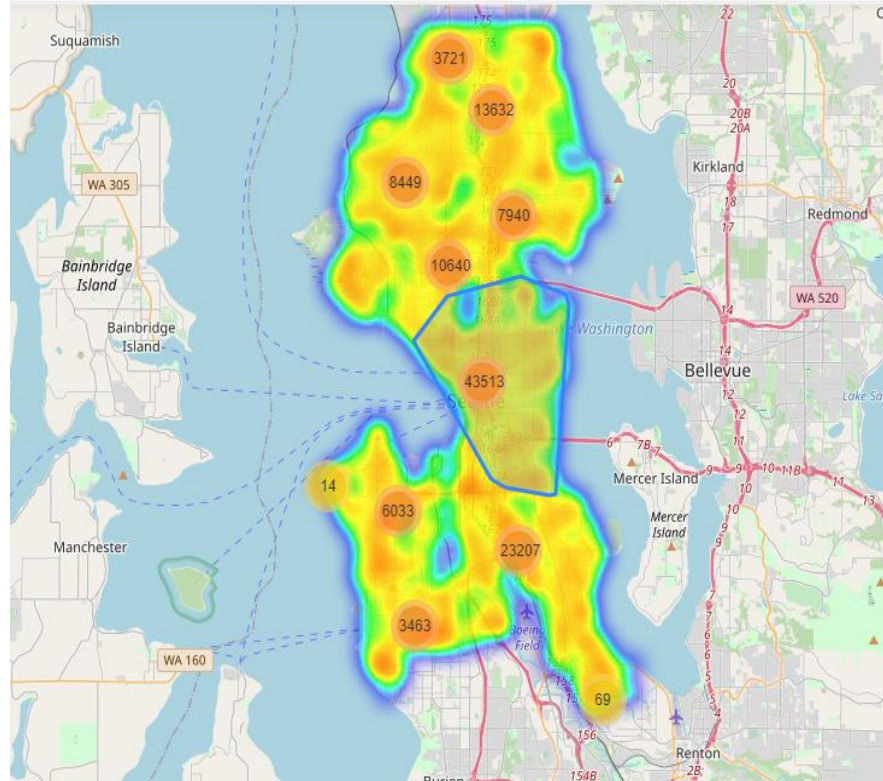


Methodology

- The methodology used was based on the CRISP-DM (Cross Industry Standard Process for Data Mining) model.
- In first step, the analytical approach was identified, based on the question: "In a given condition of road, climate, day of the week and an area of the city of Seattle, if an accident occurs, what would be the severity: an auto accident involving injury or property damage?"
- The second step was to identify the analytical approach, that is, the type of model needed to address the question (defined in the first step) more effectively, from which a classification model was used to identify the combination of conditions that lead to the result of each accident.
- In third step, the phases "data requirements", "data collection", "data understanding" and "data preparation" were performed in a way that addresses missing or invalid values and removes duplicates, toward ensuring that everything was properly formatted.
- In fourth and final step, the modeling and evaluation phases were applied to the model building as such, but it was also to see if the model meets the initial question.

Analysis

- Based on this heat map, we can clearly see the areas with the highest risk of car accidents, as well as the areas with the lowest accident rate. Thus, the blue areas indicate a low volume of collisions and the red areas a high volume.



Predictive Modeling

In order to find the best model, the following algorithms were used ...

- k-Nearest Neighbour
- Decision Tree
- Support Vector Machine
- Logistic Regression

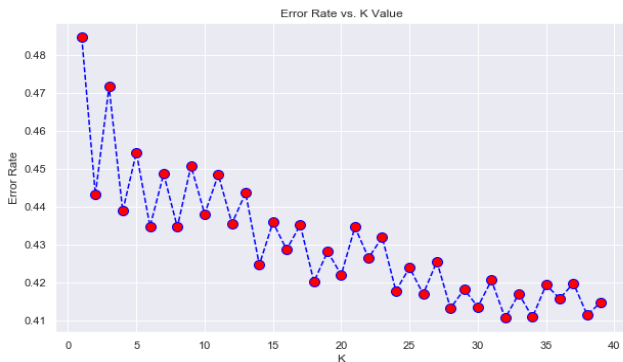
As the evaluation metrics ...

- Jaccard index
- F1-score
- LogLoass.

K Nearest Neighbor (KNN)

```
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

Text(0, 0.5, 'Error Rate')



CONFUSION_MATRIX :

```
[[9918 6317]
 [4563 3339]]
```

REPORT :

	precision	recall	f1-score	support
0	0.68	0.61	0.65	16235
1	0.35	0.42	0.38	7902
accuracy			0.55	24137
macro avg	0.52	0.52	0.51	24137
weighted avg	0.57	0.55	0.56	24137

```
print('ACCURACY :')
metrics.accuracy_score(pred_i,y_test)
```

ACCURACY :

0.5492397563906036

Decision Tree

CONFUSION_MATRIX :

```
[[14013  9283]
 [   468   373]]
```

```
: print('REPORT :\n')
print(classification_report(prediction,y_test))
```

REPORT :

	precision	recall	f1-score	support
0	0.97	0.60	0.74	23296
1	0.04	0.44	0.07	841
accuracy			0.60	24137
macro avg	0.50	0.52	0.41	24137
weighted avg	0.94	0.60	0.72	24137

```
: print('ACCURACY :\n')
metrics.accuracy_score(prediction,y_test)
```

ACCURACY :

0.5960144176989683

Support Vector Machine

```
print('CLASSIFICATION_REPORT :\n')
print(metrics.classification_report(y_pred,y_test))
```

CLASSIFICATION_REPORT :

	precision	recall	f1-score	support
0	1.00	0.60	0.75	24134
1	0.00	0.67	0.00	3
accuracy			0.60	24137
macro avg	0.50	0.63	0.38	24137
weighted avg	1.00	0.60	0.75	24137

```
print('CONFUSION MATRIX :\n')
print(metrics.confusion_matrix(y_pred,y_test))
```

CONFUSION MATRIX :

```
[[14480 9654]
 [   1    2]]
```

```
print('ACCURACY :\n')
print(metrics.accuracy_score(y_pred,y_test))
```

ACCURACY :

0.5999917139661102

Logistic Regression

```
linear.fit(X_train,y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,  
                    intercept_scaling=1, l1_ratio=None, max_iter=100,  
                    multi_class='auto', n_jobs=None, penalty='l2',  
                    random_state=None, solver='lbfgs', tol=0.0001, verbose=0,  
                    warm_start=False)
```

```
y_pred = linear.predict(X_test)  
y_pred
```

```
array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

Model Evaluation using Test set

```
print('CLASSIFICATION_REPORT :\n');  
print(metrics.classification_report(y_pred,y_test))
```

CLASSIFICATION_REPORT :

	precision	recall	f1-score	support
0	0.99	0.60	0.75	23968
1	0.01	0.48	0.02	169
accuracy			0.60	24137
macro avg	0.50	0.54	0.38	24137
weighted avg	0.99	0.60	0.74	24137

```
print('CONFUSION_MATRIX :\n');  
print(metrics.confusion_matrix(y_pred,y_test))
```

CONFUSION_MATRIX :

```
[[14393  9575]  
 [   88    81]]
```

```
print('ACCURACY_SCORE :\n');  
print(metrics.accuracy_score(y_pred,y_test))
```

ACCURACY_SCORE :

0.599660272610515

Report

- Accuracy of the built model using different evaluation metrics:

Algorithm	Jaccard	F1-score	LogLoss
KNN	0.549240	0.539595	NA
Decision Tree	0.596014	0.473522	NA
SVM	0.599992	0.450109	NA
Logistic Regression	0.599660	0.455767	0.666213

Results and Discussion

- Based on the report, as seen before, the results show that the best model is logistic regression with log loss metric.
- In addition, considering the size of the data set, + 100k, a logistic regression is more efficient in building a Machine Learning model.

Conclusion

- Purpose of this project was to answer the question "In a given condition of road, climate, day of the week and an area of the city of Seattle, if an accident occurs, what would be the severity: an auto accident involving injury or property damage?", in order to aid authorities to reach decisions as regards the preventive safety measures to prevent accidents, such as changing road infrastructure, the procurement and installation of the road signs to reflect the weather conditions. Moreover, these informations could alert drivers, given the weather and road conditions about the possibility of a car accident and the severity of it, which would allow the driver to drive more carefully or, even change his itinerary.

By analyzing accidents distribution from data, the following conclusions can be made:

- (1) Friday is the day of the week with the highest risk of car crashes in Seattle.
- (2) The clear weather and the dry road increase the risk of car crashes in Seattle.
- (3) Areas / districts 1 (from latitude 47.4 to latitude 47.5) and 2 (latitudes 47.60, 47.61, 47.62 and 47.63.), are the places in Seattle with the highest risk of car crashes.
- (4) The injury intersections and blocks have practically the same risk.

Conclusion

- By looking at confusion matrix, in the case of Logistic Regression with Jccard index, the first line refers to accidents whose value in the test set is 1, that is, "Injury". Thus, out of 24 137 accidents, the value (or type of collision) of 9 656 of them is 1. And of these 9656, the classifier correctly predicted 81 of them as 1 and 9575 of them as 0.
- This means that, for 81 accidents, the real value was 1 in the test set, and the classifier also predicted those as 1. However, while the actual label of 9 656 customers was 1, the classifier predicted those as 0, which it's not very good. We can consider it an error of the model of the first line.
- Accidents with a value of 0, ie "Property", correspond to the second line. It appears that there were 14 481 accidents with a value of 0. Where, the classifier correctly predicted 14 393 of them as 0 and 88 of them wrongly as 1. Therefore, he did a good job of predicting customers with a value of 0.
- A good thing about the confusion matrix is that it shows the model's ability to correctly predict or separate classes. In the specific case of the binary classifier, as in this example, we can interpret these numbers as the count of true positives, false positives, true negatives and false negatives

