Predicting the probability of getting into a car accident in Seattle City

Francy Martin

September 21,

2020

1. Introduction

1.1 Background

Seattle city has really troubles to deal with the accidents data. When people are driving to another city for work or to visit some friends. It is rainy and windy, and on the way, they could come across a terrible traffic jam on the other side of the highway. Long lines of cars barely moving. As the people keep driving, police car start appearing from afar shutting down the highway. It is an accident and there's a helicopter transporting the ones involved in the crash to the nearest hospital. They must be in critical condition for all of this to be happening. Now, wouldn't it be great if there is something in place that could warn the people, given the weather and the road conditions about the possibility of they getting into a car accident and how severe it would be, so that the people would drive more carefully or even change their travel if they are able to. Well, this is exactly what you will be working on in this.

1.2 Problem

The objective of the project is describing fatality of accidents. Given that the number of accidents on busy roads is creating problems in the city's transportation. It has the necessity to determine under what circumstances a serious traffic accident is most likely to occur.

Among the factors to be evaluated will be the type of crash, and means of transport, the weather and the road conditions about the possibility of you getting into a car accident and how severe it would be.

2. Data cleaning

2.1 Data cleaning

Data The dataset contain 194673 observations (rows) and 38 attributes (columns). First the columns that have NaN were checked. Columns that had more than half like this were eliminated. Columns that contained duplicate information from other columns were also removed.

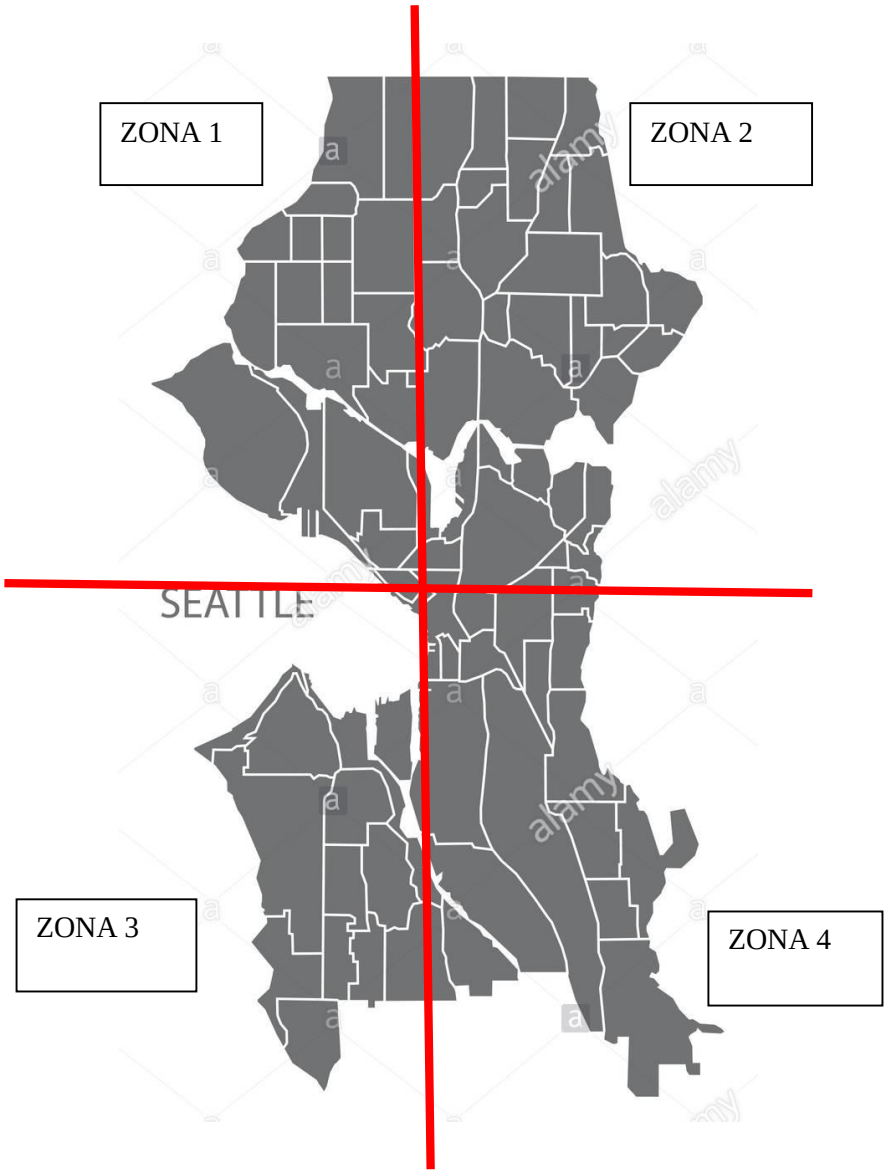
```
df.drop(['INCKEY', #irrelevant
         'OBJECTID', #irrelevant
         'REPORTNO', #irrelevant
         'STATUS', #irrelevant
         'LOCATION', #Duplicate Information,
         'SDOTCOLNUM', #Duplicate information
         'SDOT_COLCODE',
         'SDOT_COLCODE',
         'INTKEY', #irrelevant
         'HITPARKEDCAR',
        'COLDETKEY',
        'EXCEPTRSNCODE',
        'EXCEPTRSNDESC',
        'SEVERITYCODE.1',
        'INATTENTIONIND',
        'SDOTCOLNUM',
        'SPEEDING', #many missing data
        'ST_COLCODE',
        'ST_COLDESC',
        'SEGLANEKEY',
        'ST_COLCODE',
        'CROSSWALKKEY',
        'PEDROWNOTGRNT',
         'UNDERINFL',
         'PERSONCOUNT',
         ], axis = 1, inplace = True)
```

Eliminated Columns

Then the rows that did not have values of the columnhas corresponding to the coordinates, and type of collision were eliminated. All this to eliminate the null and irrelevant values. Finally, it was replaced by the frequency in the columns 'WEATHER' and 'ROADCOND', cause it is most likely to occur.

In addition, the YEAR, MONTH, and TIME columns were created, replacing the previously existing date columns. To see if depending on these the probability of an accident was higher.

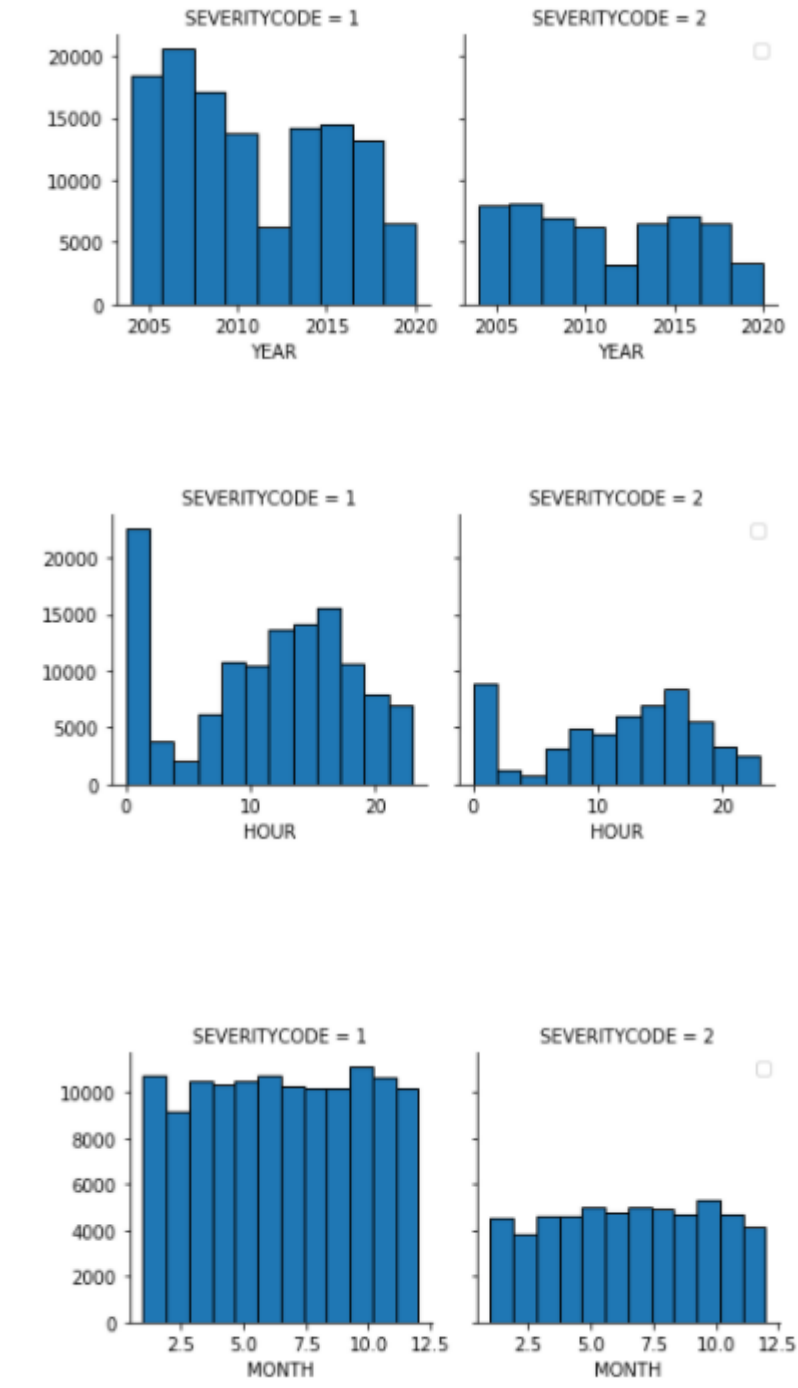
In addition, the coordinates were grouped into four areas as shown in the figure below. In such a way that each data will be grouped in one of these Zones and know which of these is more prone to accidents.



3. Exploratory Data Analysis

3.1 Relationship between 'YEAR', 'MONTH', 'HOUR' and "SEVERITY"

It see that the month of the year does not influence the fatality of the accident. While it is seen that as the years go by, the number of accidents decreases, then only the last 5 years will be taken into account for the study. Finally, a strong relationship is observed between the time of day and the number of accidents.



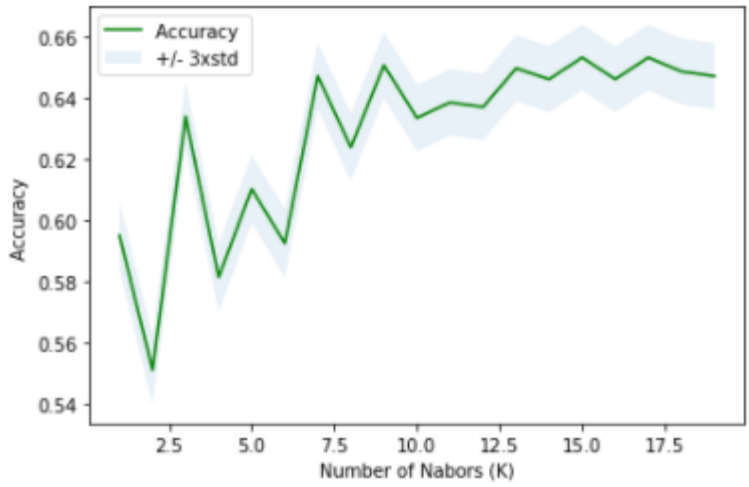
Box plot of improvement of players of different ages.

4. Predictive Modeling

In this case, given that the variables are not continuous, that is, they are discrete. A Classification method had to be used. In this case, K nearest Neighbors was chosen.

4.1 Regression models

4.1.1 K Nearest Neighbor




```
The best accuracy was with 0.6530303030303031 with k= 15
```

jaccard KNN: 0.6530303030303031

F1-score KNN: 0.6259979935509098

Figure 10. Distribution of actual and predicted improvement using linear regression with different weights of samples based on inverse of sample abundance.

4.1.2 Performances of different models

Using the new approach of different sample weights, I built linear regression, SVM, random forest, and gradient boost models using weighted root mean squared error as the evaluation metric. For each model, hyperparameters were tuned using the same metric and cross validation. For comparison, I also built a simple linear regression model with just one independent variable (age) as the benchmark model. SVM had the best performance among all models, which had ~26% less error than the benchmark model (Table 2). The predicted improvements had linear relationship with the actual improvements (Figure 11).

Table 2. Performance of the regression models.

	Benchmark (one feature)	Linear Regression	SVM	Random Forest	Gradient Boost
Weighted RMSE	3.84	2.98	2.86	2.93	2.96

4.2 Classification models

The application of classification models was much more straightforward. I divided the samples into two classes (improvement>=0 or <0). The number of samples in each class were about the same. I chose logarithmic loss as the metric here because the results would probably be presented with probabilities and logarithmic loss puts more emphasis on the probabilities than other metrics. Logistic regression, SVM, random forest, gradient boost models and a voting model were tuned and built. Among the individual models, the SVM model performed the best (~67.5%

accuracy), and voting model performed similarly as the SVM model (Table 3), though the differences between models were small.

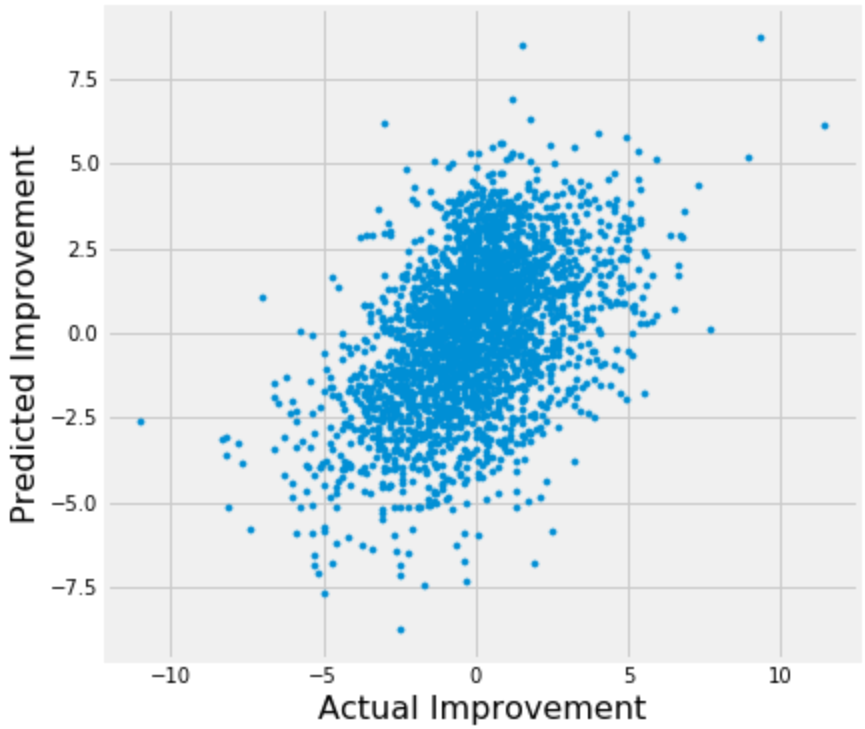


Figure 11. Scatter plot of predicted and actual player improvements of the SVM model.

Table 3. Performance of classification models. Best performance labeled in red.

	Logistic Regression	SVM	Random Forest	Gradient Boost	Voting Model
Log Loss	0.605	0.603	0.612	0.613	0.603
Accuracy	0.675	0.675	0.672	0.672	0.675
No. of True Positives	835	830	810	815	838
No. of False Positives	413	406	396	400	416
No. of False Negatives	438	443	463	458	435
No. of True Negatives	929	936	946	942	926

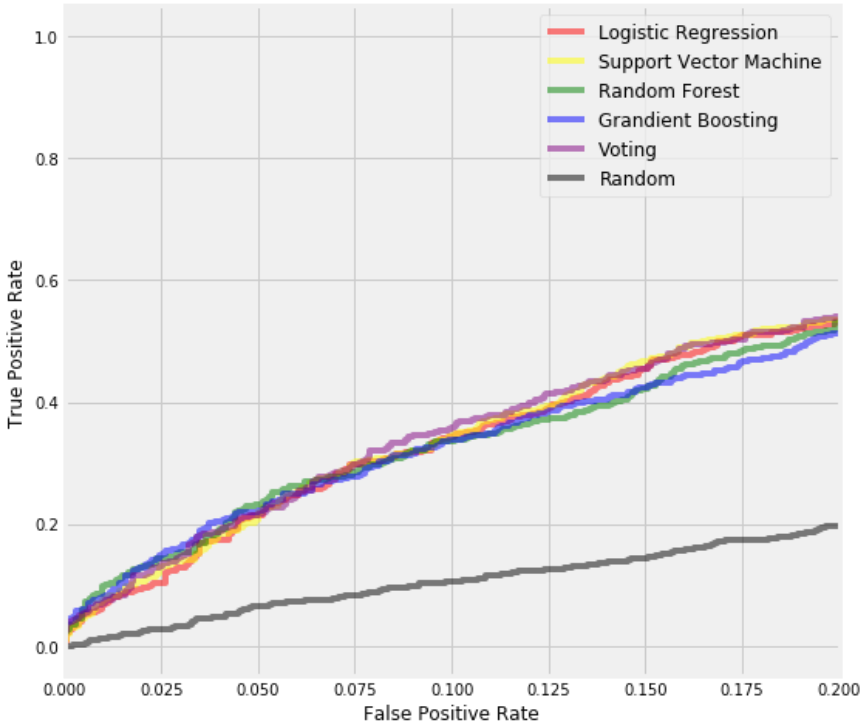


Figure 12. A section of ROC curves of different classification models.

I also evaluated the models using their ROC curves. In this particular problem, lower false positive rate is more important than higher true positive rate. In other words, it is more important to be sure that a player will improve as predicted, rather than predict all players who will improve, simply because a team can only have limited number of players. In the ROC curves with low false-positive rate, the voting model had slightly higher true positive rates than other models (Figure 12).

5. Conclusions

In this study, I analyzed the relationship between NBA players' improvement/decline and their performance and biographic data. I identified age, win share, minutes/games played, improvement last season among the most important features that affect a player's improvement next season. I built both regression models and classification models to predict whether and how much a player would improve/decline. These models can be very useful in helping NBA team management in a number of ways. For example, it could help identify players to acquire, estimate the size of the contract to offer players, plan for performance changes of players already on the team, etc.

6. Future directions

I was able to achieve ~26% improvement from the benchmark model in the regression problem, and ~68% accuracy in the classification problem. However, there was still significant variance that could not be predicted by the models in this study. I think the models could use more improvements on capturing players' individual traits. For example, two players might have similar performance metrics, but one might be more physical and the other might be more finesse. The future performance of these two types of players might be different. Another example is that players whose contracts are expiring might play harder/better than players who just signed hefty contracts. More data, especially data of different types, would help improve model performances significantly.

Models in this study mainly focused on individual features. However, interactions with teammates, coaches, might also contribute to a player's performance. For example, if a player had a new teammate who is a superstar at the same position, his performance is likely to suffer because of competition. These interactions data are obviously more difficult to extract and quantify, but if optimized, could bring significant improvements to the models.