

```
26 <br> After cleaning the data and before modeling, I decided to do some
visualizations of the remaining data. After removing null values, I was left
with 36363 entries. The first visualization is a break down of stops that ended
in arrest. 11,278 stops ended in arrest with 25,085 ending with a non-arrest.
Further breakdowns showed that of the age group 26-35 had the highest
percentage of arrests when divided by age of subject. White subjects were the
most stopped, however American Indian/Native Alaskans had the highest
percentage of arrests when stopped. Men were pulled over more likely than
women, but the rate of arrest was only 3% apart. Same with gender of officer.
The race of the officer, however painted a different picture. While white
officers made up nearly 3 times all of the other officers combined, their rate
of arrest was the same or about all races recorded, except Asian, Black, and
Native Hawaiiin/Pacific Islander.
27 Finally, breaking down time by AM and PM resulted in arrest rates withing 0.2%,
and breakdown by hour shows that there are more arrests in the late night hours
of 11pm-3am and the afternoon/evening 4pm-7pm.
28
29
30
31 ##### First Model: Logistic Regression
32 <br> The inital model I went with was a Linear Regression model. After
OneHotEncoding and scaling my data, the initial model returned and accuracy of
57%, or just barely over a coin flip. Looking at a confusion matrix, the model
seemed to predict an arrest when in fact the subject was not arrested. The
other tests of Precision, Recall, and F1 score were average, hoveirng in the
50% range.
33
34 ##### Second Model: K Nearest Neighbors
35 <br> Running a K Nearest Neighbors model increased the accurcacy to 64% but
unfortuneatly knocked the other tests down to 20%-30% range. After running a
GridSearchCV to finding the best parameters for a model and running the model
again, the accuracy jumped up to 68%, whil tanking the recall and F1 score.
36
37 ##### Third Model: XGBoost
38 <br> XGBoost gave us the best accuracy and precision yet, with 69% and 50%
each. However recall and F1 went way downm, sub 1%. After running a GridSearchCV
on XGBoost, which ran for 30 minutes, I was unable to get that accuracy abou
69%.
39
40 ##### Final model: Decision Tree
41 <br> Decision fared worst, not getting us to the heights of 69% and when
running a GridSearchCV, the model actually broke, being unable to return a
score for recall, precision, or F1 score.
42
```