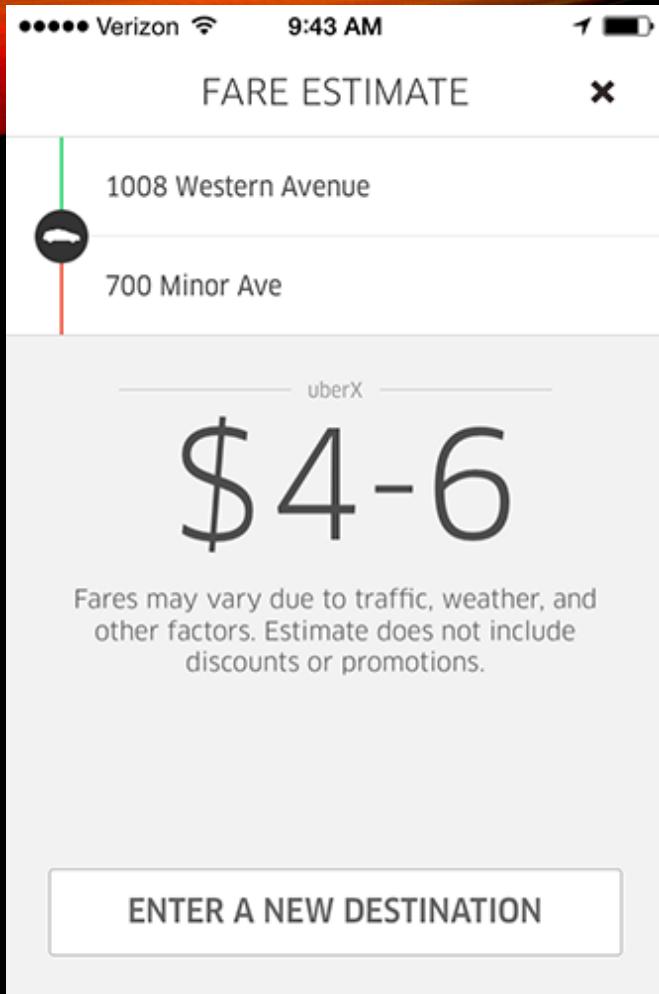


# PREDICTING NYC TAXI FARES

Supervised Learning Capstone

David Antzelevitch  
Thinkful Data Science Bootcamp  
11/1/2018

# THE PROBLEM



## Background:

- Ride hailing is competitive in NYC
- 800,000 rides are hailed per day
- Yellow Cabs are losing to Uber and Lyft
- Yellow Cab share is 35%

## Current Situation:

- Passengers want fare transparency
- Uber/Lyft provide Fare Estimates to potential passengers
- Yellow Cabs do not

## Project Goal:

- Develop a machine learning model to allow Yellow Cabs to quote fares, just like Uber and Lyft

# THE APPROACH

- Data was collected for 55,000 million NY cab rides from 2010—2015
- Data includes:
  - Date/Time of pickup
  - Pickup Location Coord (Lat, Long)
  - Drop off Location Coord (Lat, Long)
  - Fare Amount

The Goal

Create model to predict Fare Amount

Given: Date/Time, pickup location, drop off location

# WHAT MAKES UP A CAB FARE?

## Base Fare

- Min Fare: \$2.50

## Tax

- \$0.50 Taxes

## Distance

- \$0.50 per 1/5 mile

## Peak/Off Peak Fees

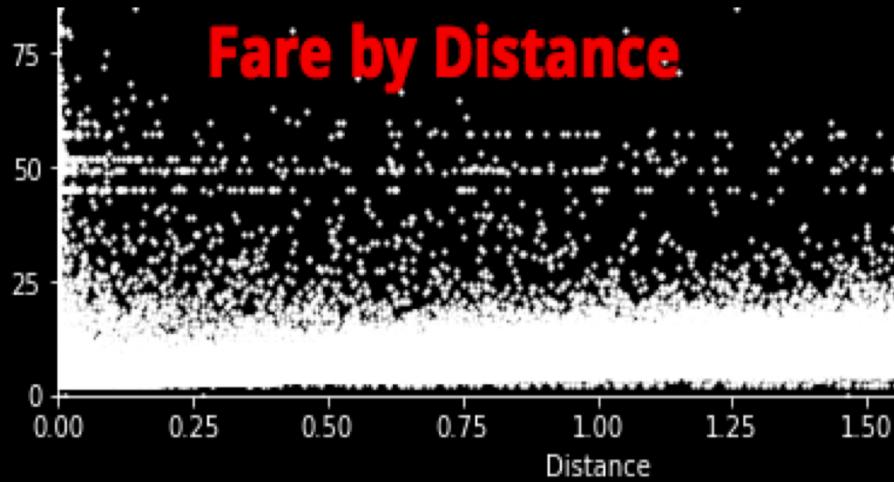
- \$0.50 night fee: 8pm to 6am
- \$1.00 peak fee: 4pm to 8pm

## Tolls

### Waiting Time

- \$0.50 per minute waiting  
(moving < 12.5mph)

# WHY CAN'T WE JUST USE DISTANCE?

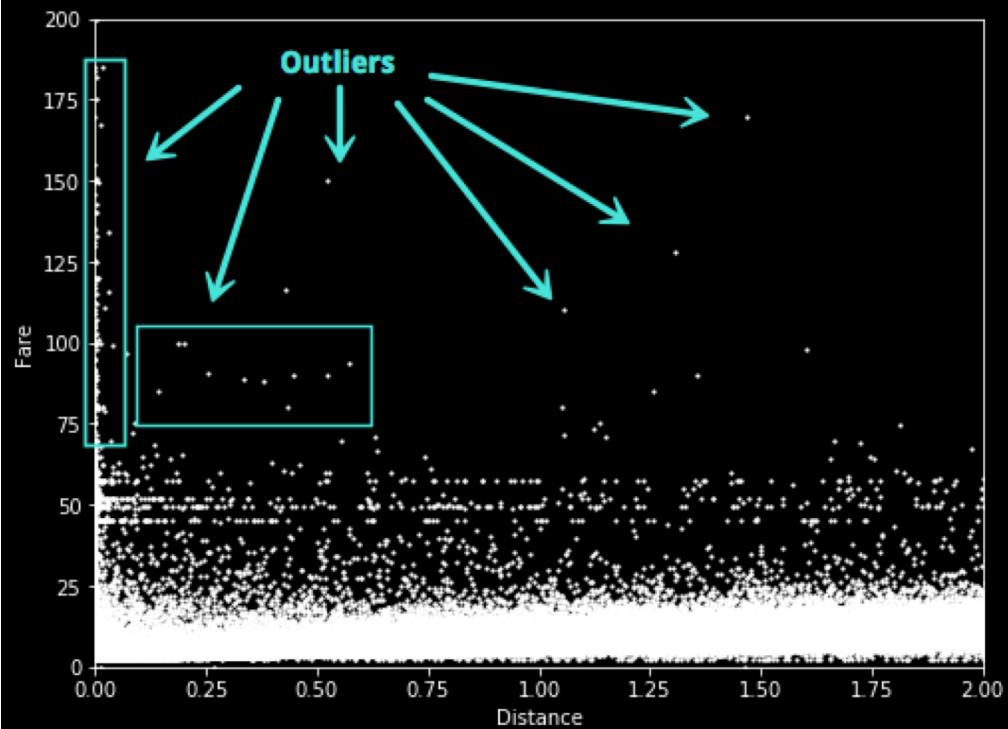


- Waiting time is the main problem
- Hard to predict traffic
- Trips less than 2 miles can have fares ranging all the way up to \$50!

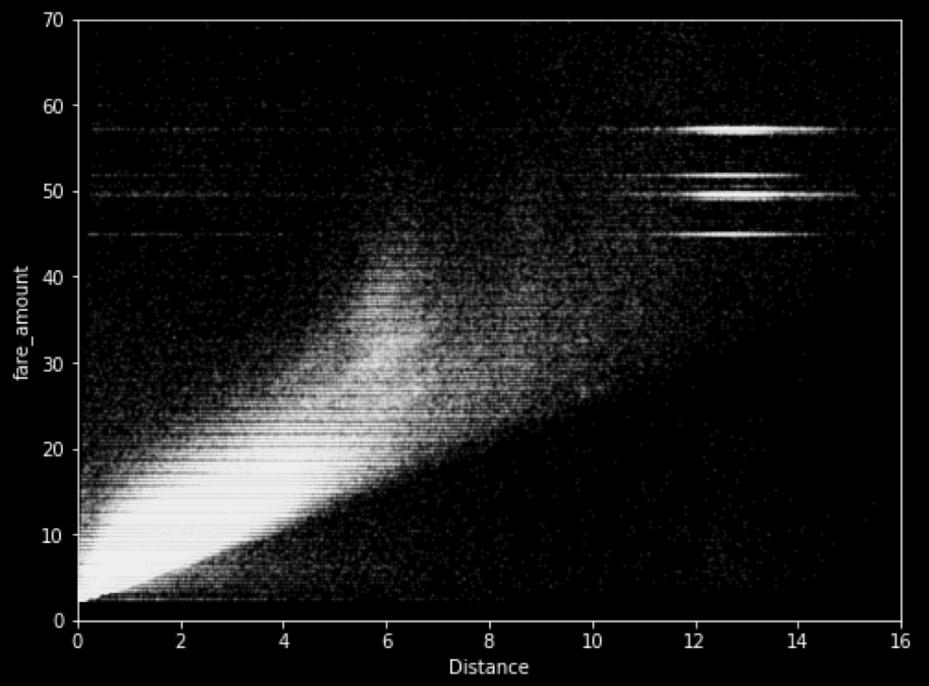
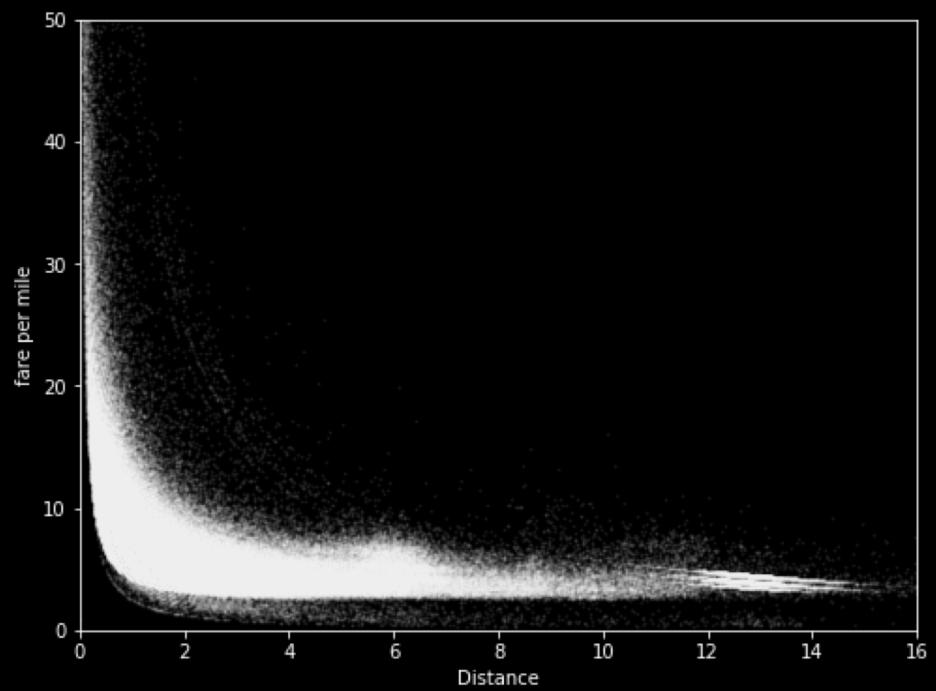
# DATA CLEANSING

Removed:

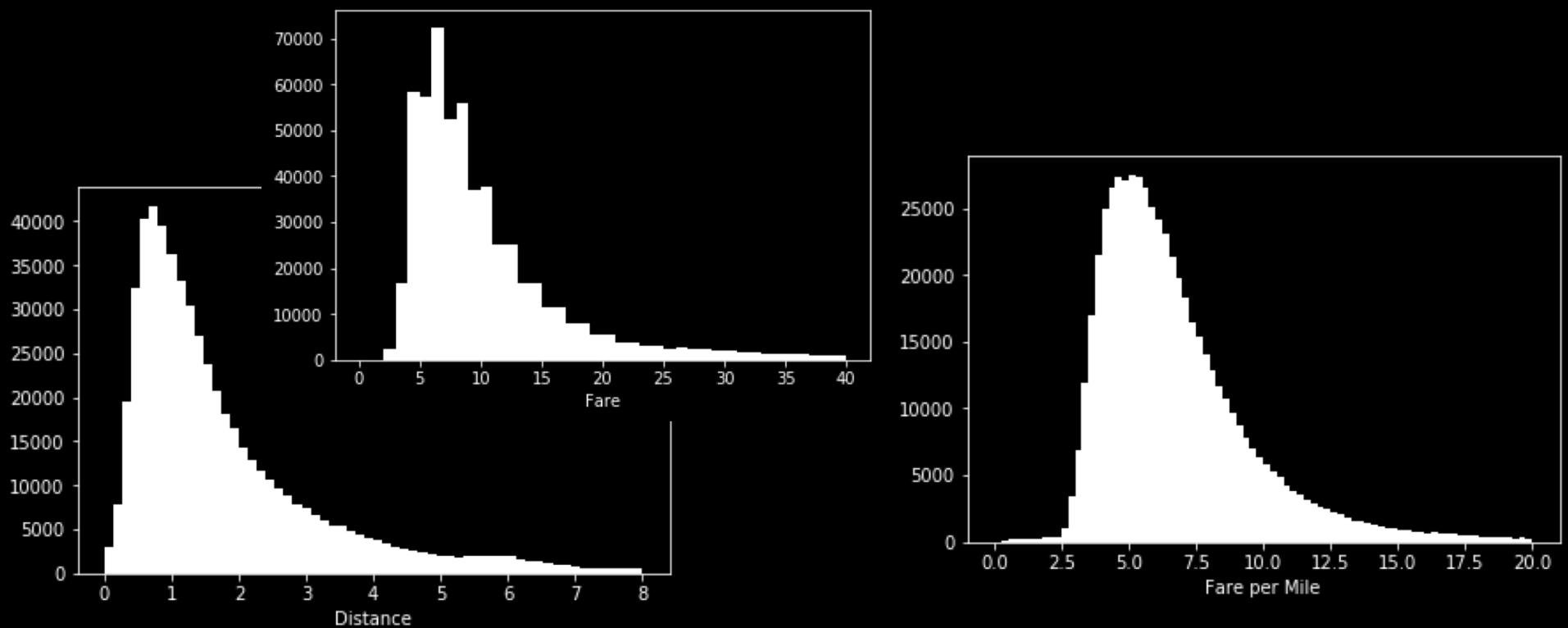
- Trips costing more than \$100 per mile
- Trips with Distance of 0
- Trips with Fare of \$0
- Locations out of Range



# EXPLORING THE DATA



# EXPLORING THE DATA



# PREDICTING FARE WITH MACHINE LEARNING

## Predict Fare via Distance Calculator

An intelligent calculation for fare would be:

$$\text{Fare} = \$5 + \$0.67 \text{ per } 1/5 \text{ mile}$$

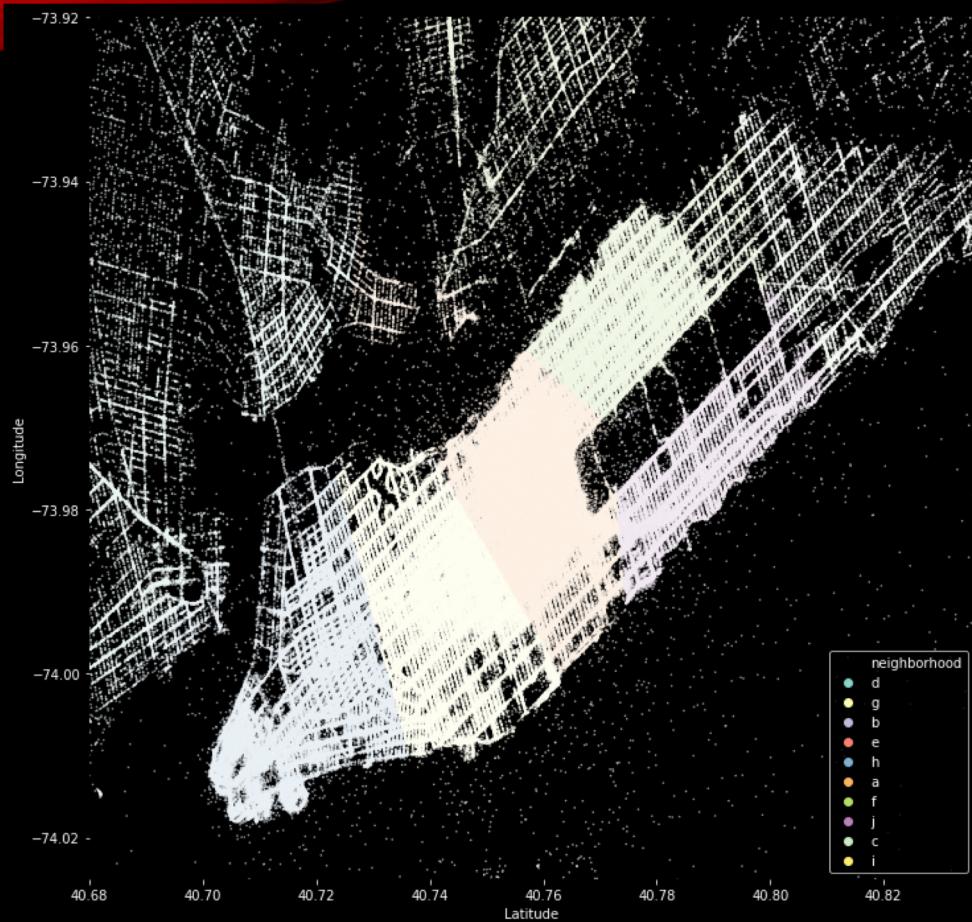
This model predicts the fare with  
**75% accuracy**

## Predict Fare via Machine Learning

By analyzing 100k trips and various variables, machine learning improves the fare prediction.

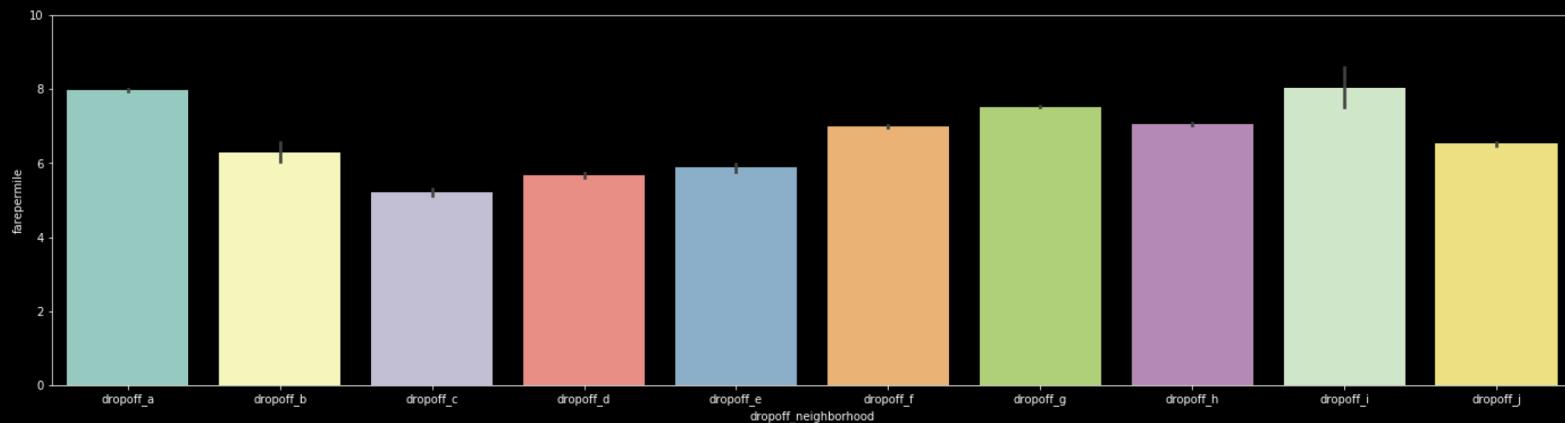
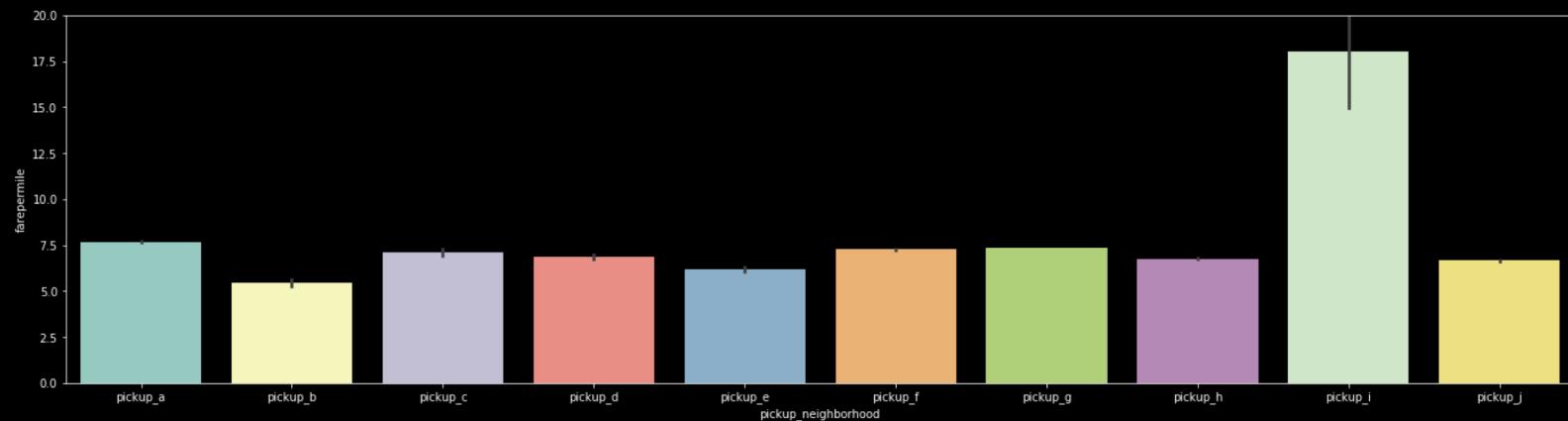
A machine learning model predicts fare with  
**85% accuracy**

# IN WHAT NEIGHBORHOOD IS THE TRIP?

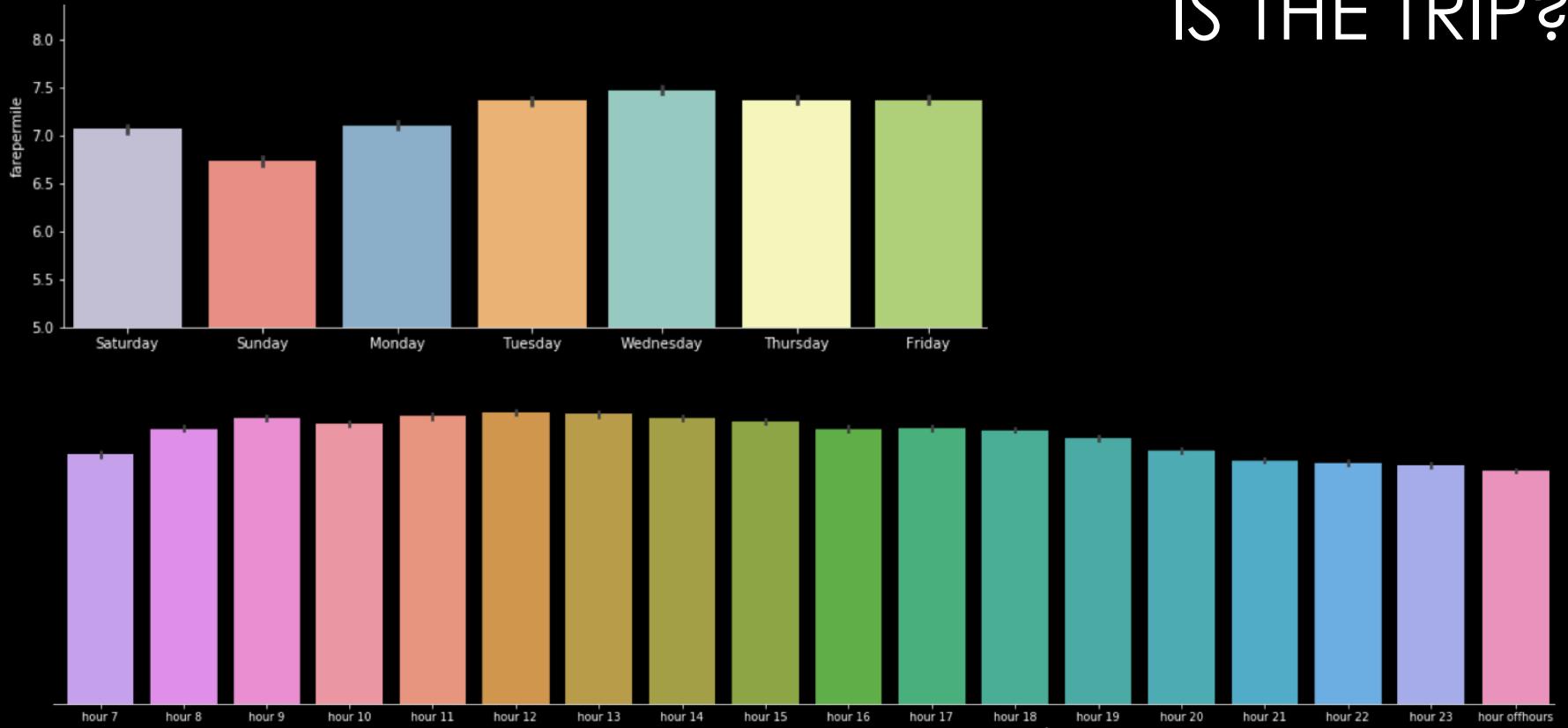


- Raw data did not include neighborhood
- It had latitude/longitude of pickups and drop offs
- Used K-Means clustering to classify each location into a neighborhood

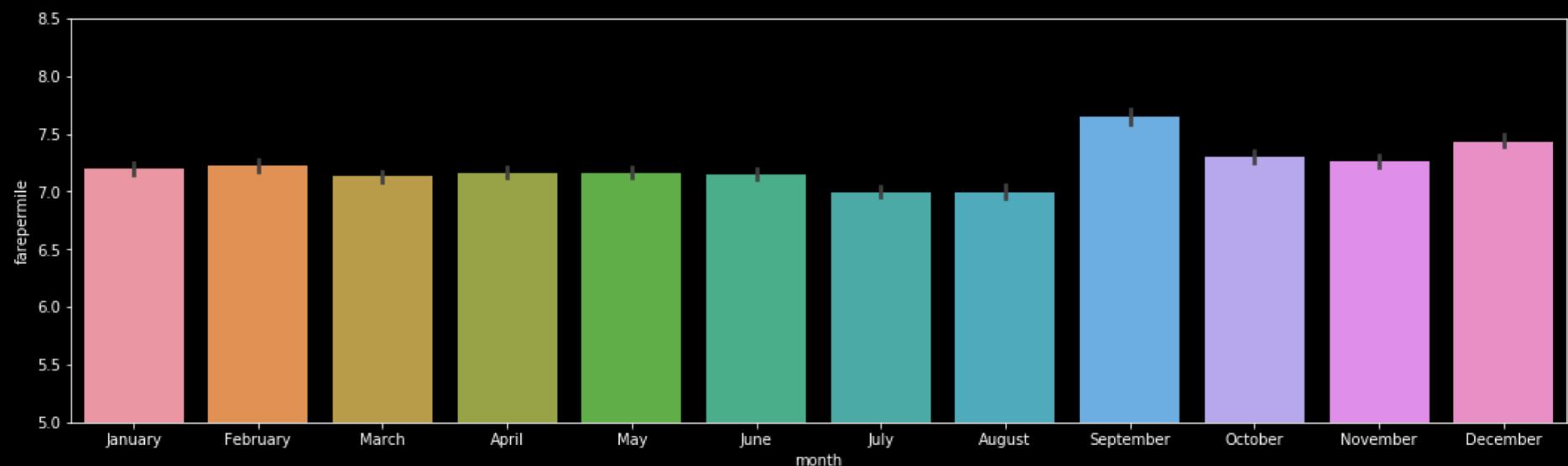
# IN WHAT NEIGHBORHOOD IS THE TRIP?



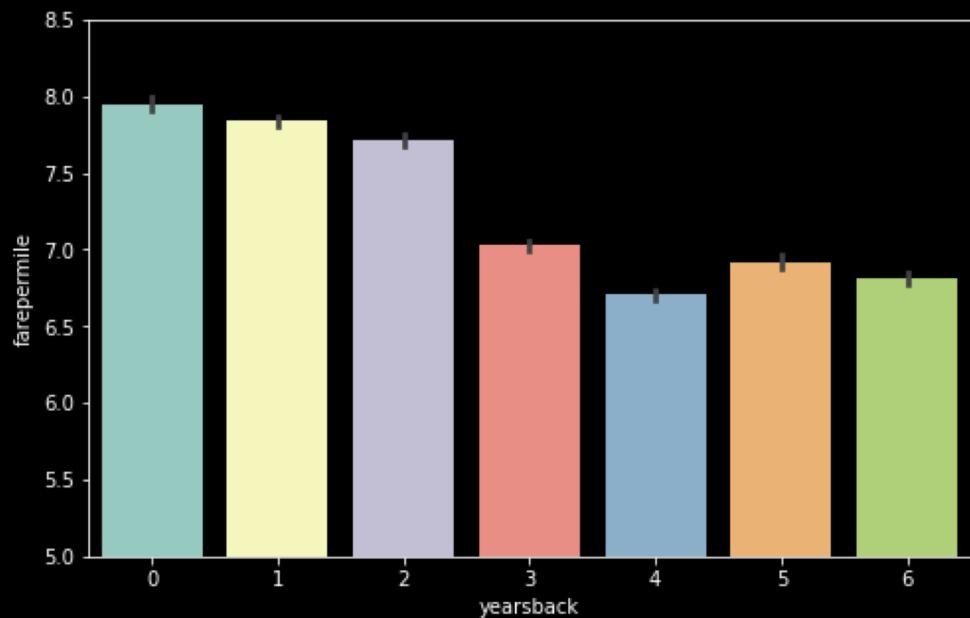
# WHAT DAY/TIME IS THE TRIP?



# IN WHAT MONTH IS THE TRIP?



# ADJUSTING FOR YEAR



- Preferred to use only current year data
- The problem, current year data did not contain all months
- Used data from all years
- Model considers the year

# LINEAR REGRESSION

	Parameter	Coeffecient
hour	Distance	3.18766
	pickup_f	2.72567
	pickup_e	1.80419
	dropoff_f	1.7397
	dropoff_a	-1.24568
	dropoff_e	1.16563
	dropoff_d	-1.16297
	pickup_i	-0.93435
	pickup_h	-0.843274
	pickup_d	-0.802747
offhours	hour offhours	-0.796381
	pickup_g	-0.700399
	dropoff_c	0.669987
	pickup_a	-0.639254
	pickup_c	-0.56551
yearsback	yearsback	-0.539557
	dropoff_g	-0.509957
	hour 22	-0.473082
	dropoff_i	-0.447837
	hour 14	0.428566
	hour 7	-0.42768
	hour 21	-0.423438
	hour 23	-0.405336

hour 20	-0.401007
hour 15	0.399084
hour 11	0.386091
December	0.374973
hour 16	0.36823
September	0.363612
November	0.347777
January	-0.329728
hour 13	0.325502
October	0.320482
hour 12	0.30835
February	-0.287021
March	-0.285356
Sunday	-0.249097
hour 17	0.236832
hour 10	0.233097
April	-0.220042
Thursday	0.19481
hour 9	0.182364
dropoff_h	-0.164548
August	-0.164452

hour 19	-0.163482
July	-0.1503
Wednesday	0.132476
Friday	0.129599
Saturday	-0.122603
hour 8	0.113138
hour 18	0.109152
Monday	-0.0803993
June	0.0476519
dropoff_j	-0.0464079
pickup_j	-0.0464079
May	-0.0175972
Tuesday	-0.0047855
dropoff_b	0.00208805
pickup_b	0.00208805

L1 (Ridge) Regularization helps make coefficients explainable

# COMPARING MODEL ACCURACY

100k Sample	Distance Calculator (No ML)	Linear Regression	K Nearest Neighbors Regression
5 NY Neighborhoods	.751 (+/- .055)	.767 .044	.843 .013
10 NY Neighborhoods	.751 (+/- .055)	.777 .057	.847 .013
15 NY Neighborhoods	.751 (+/- .055)	.793 .053	.843 .012

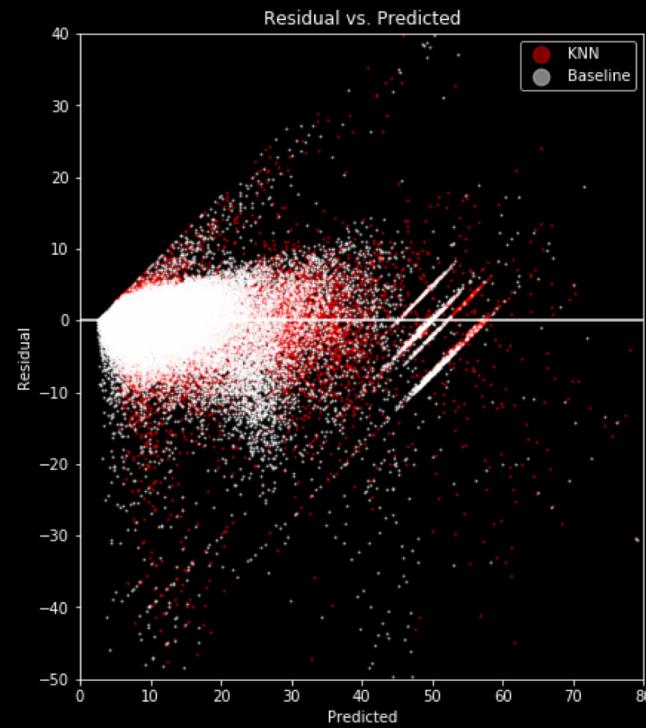
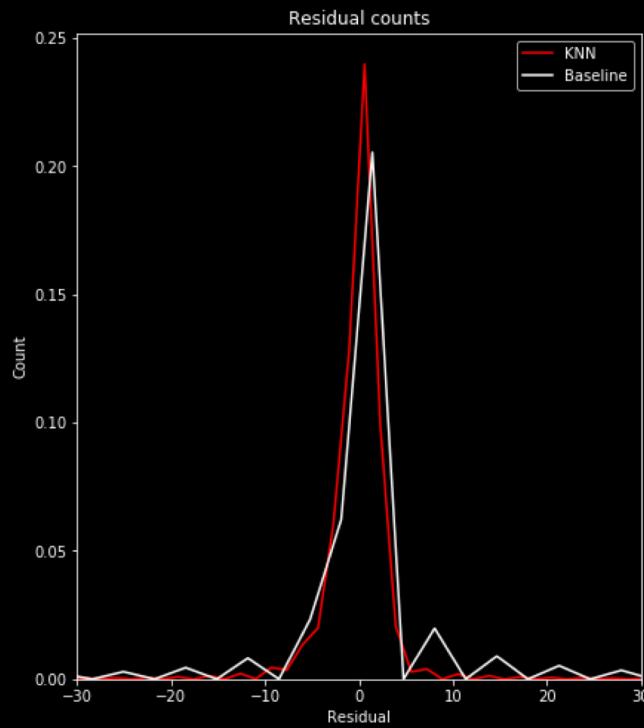
# FINAL MODEL TUNING

Used sklearn GridSearchCV to fine tune the model

Find the optimal number of K neighbors.

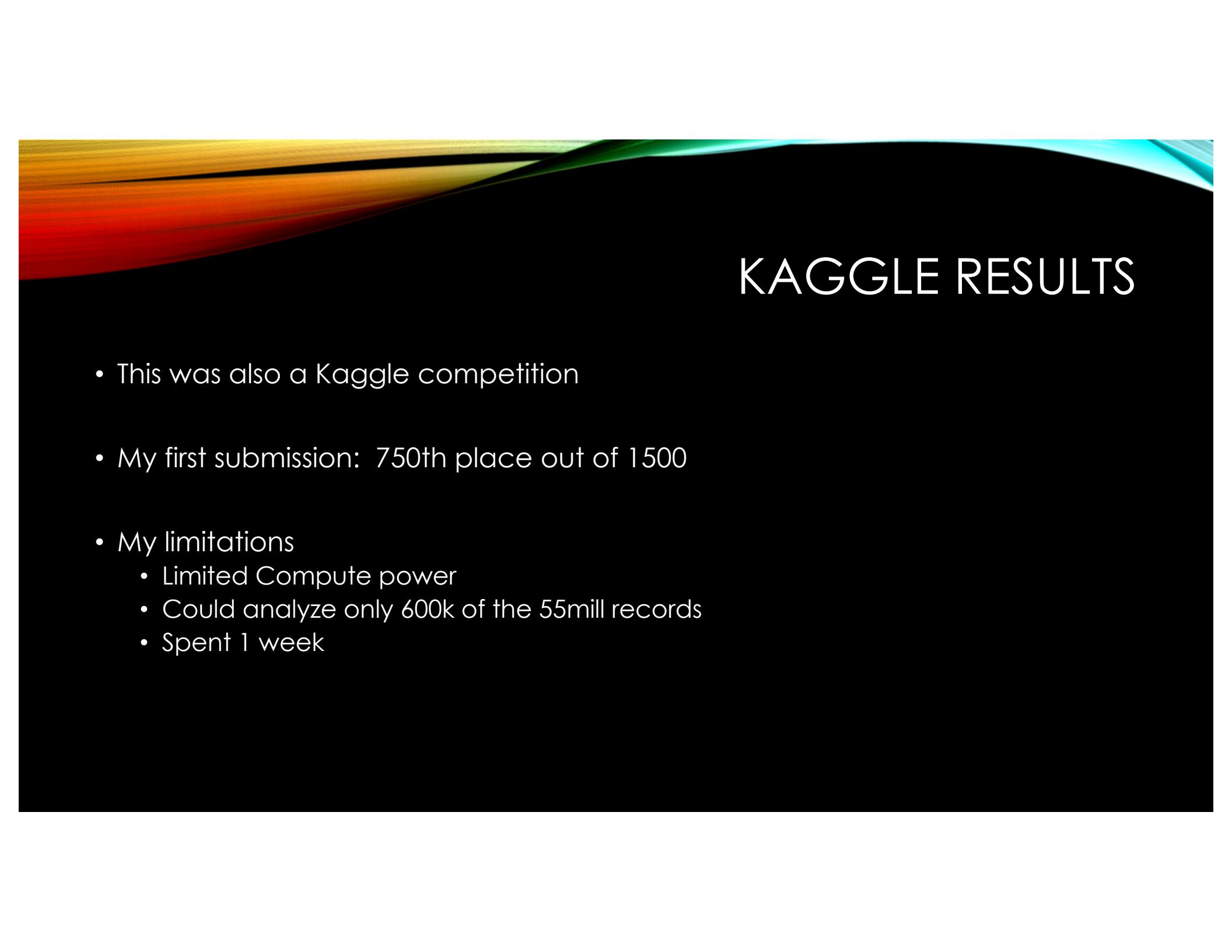
	10 Neighbors	15 Neighbors	20 Neighbors	30 Neighbors	40 Neighbors
Cross Validation Accuracy	.846 (+/- .014)	.847 (+/- .013)	.847 (+/- .013)	.846 (+/- .012)	.845 (+/- .012)

# T-TEST: IS KNN MODEL BETTER THAN USING A SIMPLE DISTANCE CALCULATOR?



	<b>Accuracy</b>
KNN	.847 (+/- .013)
Distance Calculator	.751 (+/- .055)

	<b>T Test</b>
Hypothesis	KNN model improves accuracy over Baseline model
P-Value	1.53e-05
Result	99.9% confident that KNN does improve accuracy



## KAGGLE RESULTS

- This was also a Kaggle competition
- My first submission: 750th place out of 1500
- My limitations
  - Limited Compute power
  - Could analyze only 600k of the 55mill records
  - Spent 1 week