Crash Conjectures

Applied Project

Group 1: Unsupervised Twitter Robots

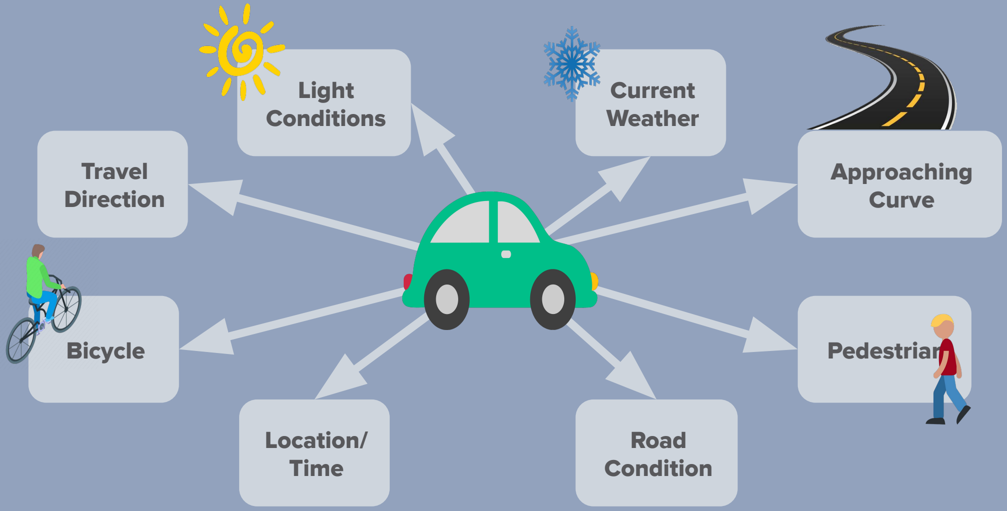
Dagny Stahl ◆ Luke Valentine ◆ Elise Renwick

Purpose

Create a model to predict if an accident, based on the current driving situation, would cause casualties in order to inform self driving cars

Product

API that a self driving car can interact with!



Sends Request

```
{
    "casualties": false,
}
```

Demo at end

The Main Dataset

Denver Car accidents data from CDOT, 189k rows

Information like:

- car type
- road & light conditions
- pedestrian or bike involvement
- Location
- date/time
- car movement

Mix of numerical, text, date, and time data

Feature Selection

Mutual Information Score Feature Selection

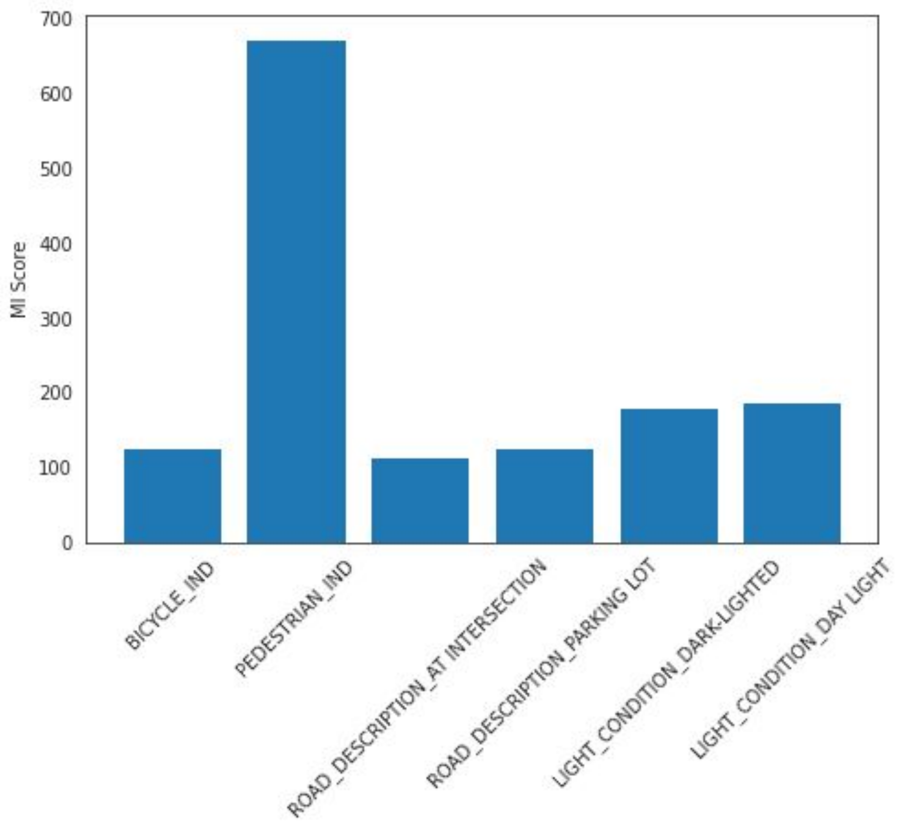
```
BICYCLE_IND is 0.077
PEDESTRIAN_IND is 64.729
ACCIDENT_OFFENSE_ACCIDENT is 12.36
ACCIDENT_OFFENSE_DUI/DUID is 2.85
NEIGHBORHOOD_Congress_Park is 6.53
HIGHWAY_INTERCHANGE is 12.12
INTERSECTION is 0.556
PARKING_LOT is 0.466
ROAD_CONTOUR_HILLCREST is 0.025
ROAD_CONTOUR_STRAIGHT is 1.22
ROAD_CONDITION_DRY is 1.33
ROAD_CONDITION_ICY is 0.32
LIGHT_CONDITION_DARKis 5.1
LIGHT_CONDITION_DAY_LIGHT is 3.50
DIRECTION_NORTH is 1.78
DIRECTION_NORTHEAST is 0.051
```

Data Cleaning

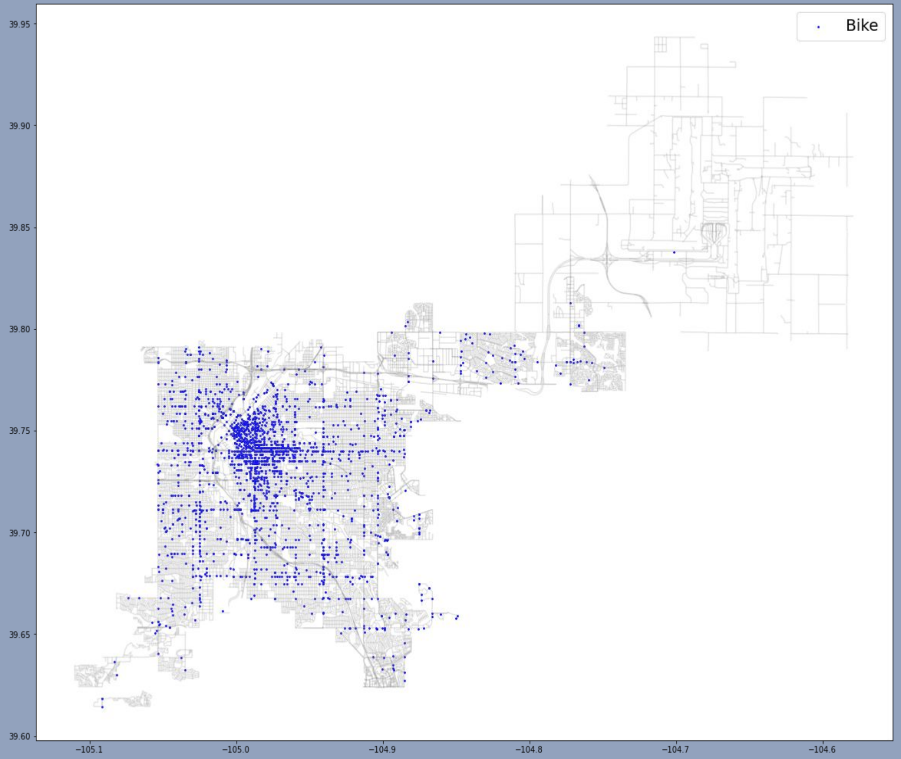
Data Formatting:

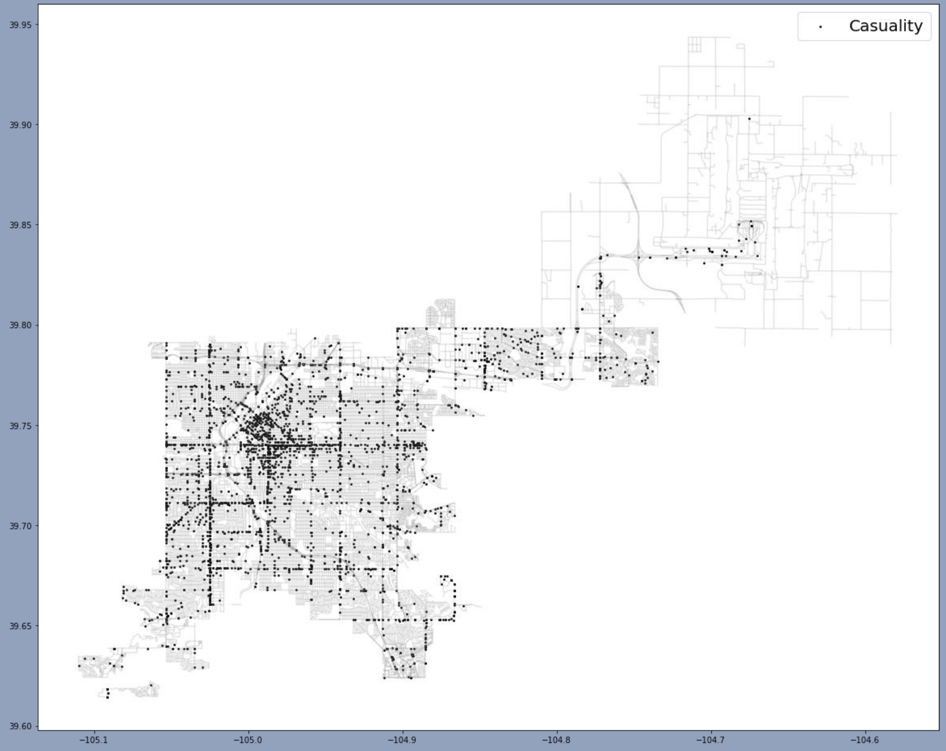
- Got date and time in a consistent format
- Dropped unnecessary columns (ex.“precinct ID”, “latitude”)
- Combined “fatalities” and “injuries” to “casualties”
- Ran dummies on categorical columns (ex. “neighborhood”, “road_condition”)
- Balanced data (1.7% of samples had casualties)
- Normalized the data
- Kept features a self driving car would have, (ex. removed “accident_offense_DUI”

Feature Selection



Feature Exploration





Failed Models: Regression

- Linear
- KNN
- SGD

Scored well on unbalanced data:
R^2 > 0.84 & MSE < 0.0106

...But horribly on balanced data:
R^2 < 0

The Problem: Only predicting no casualties, which was a safe bet, but the model was useless

Failed Models: Neural Net

New NN
Chose 20 best features from MI:

- NEIGHBORHOOD_ID
- BICYCLE_IND
- PEDESTRIAN_IND
- ROAD_CONDITION_ICY
- LIGHT_CONDITION_DUSK
- TIME_HR
- ...

to predict: CASUALTIES

Layers:

```
Dense(10,"relu")
Dropout(.9)
Dense(10,'relu')
Dropout(.9)
Dense(1)
```

Issues with our first NN

- **Trained on unbalanced data**
- **Predicting fatalities from info such as injuries**
- **Used features a SDC wouldn’t have**

Failed Models: Neural Net

Scored well on unbalanced data with more features:

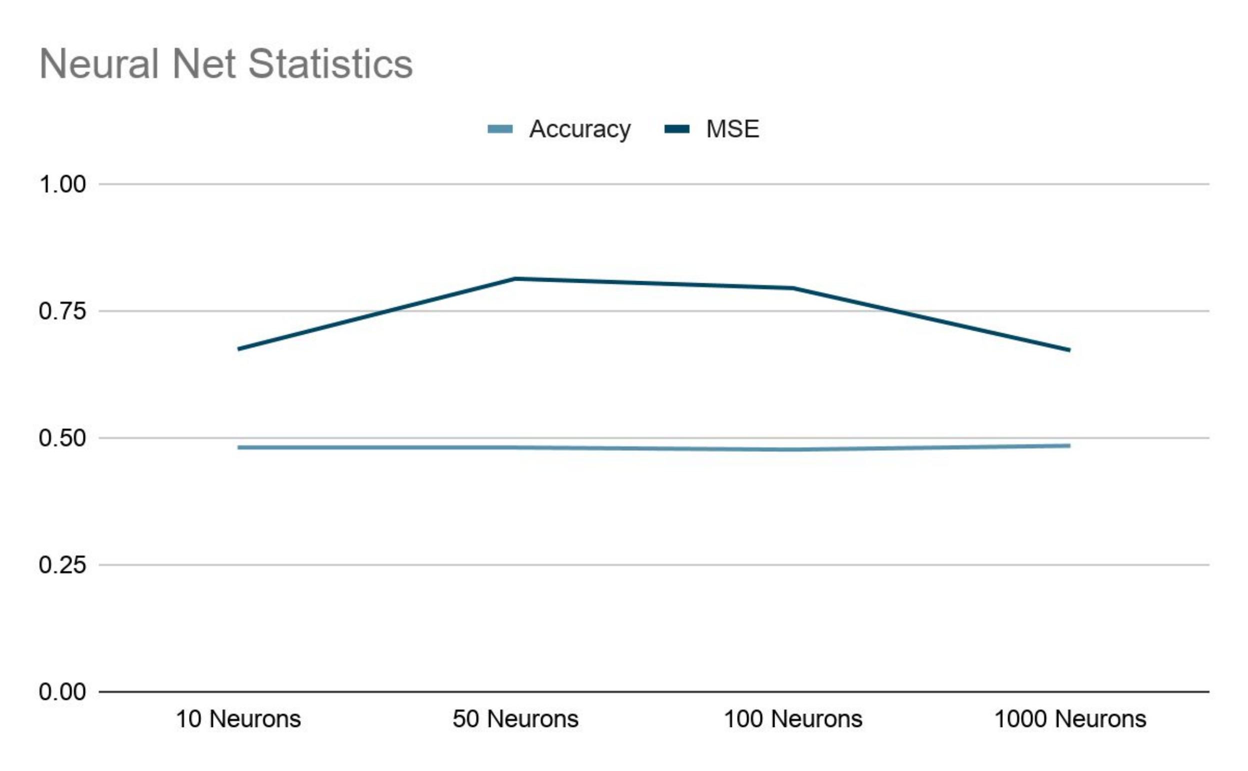
loss: 1.9037e-04
mean_absolute_error: 1.9037e-04
mean_squared_error: 3.6240e-08

...But horribly on balanced data with fewer features:

loss: 0.5981
mean_absolute_error: 0.5981
mean_squared_error: 0.8070
accuracy: 0.4788

The same problem: Only predicting no casualties

Failed Models: Neural Net



Couldn't improve accuracy or MSE

- ***Increasing neurons***
- ***Changing activation functions***
- ***Adding dropout layers***

Best accuracy ~ 0.5
Best MSE ~ 0.65

Final Model: KNN Classifier

Trained a KNN Classifier:

- *Did an 80/20 train_test_split*
- *Transformed CASUALTIES to a binary target*
- *Chose 6 best features from MI scores*

Mean Accuracy Score: 0.51

Attempted to up the score by:

- *Hyperparam search (GridSearchCV, n_neighbors = 24)*
- *Performed Cross Validation*
- *PCA*
- *NCA*
- *Made derived columns (ex. "rush_hour" from "date" & "time")*
- *Tried KNN, SGD*

… After all this, the best Mean Accuracy Score was: 0.65

Final Model: KNN Classifier

	Predicted false	Predicted true
Actual false	646	116
Actual true	408	321

```
              precision    recall  f1-score   support

         0.0       0.61      0.85      0.71       762
         1.0       0.73      0.44      0.55       729

    accuracy                           0.65      1491
   macro avg       0.67      0.64      0.63      1491
weighted avg       0.67      0.65      0.63      1491
```

Product Demonstration

HEROKU

POSTMAN

Lessons Learned

- **The data set may not have the best information for predicting how bad a crash will be. Research shows these things play major roles in fatal accidents:**
  - Speeding
  - Driver behavior (distracted driving, seat belts)
  - Car model
  - Head on collisions (typically caused by extreme driving errors/negligence)
- **We underestimated how important data cleaning, feature selection, and data transforming is to ML. Especially with messy real-life government datasets**
- **Our initial choice of data set was not ideal for our goal!**

Final Thoughts

- We’re especially proud of how well we wrangled the dataset
- Although the classifier isn’t performing amazingly well, it’s getting over 0.5, which shows some predictive ability
- Our preprocessing techniques were successful since we did see improvement in our model after using them.
- We initially had really good results with regression and the NN, but we were able identify why the models weren’t actually useful

Drive Safe!

Group 1: Unsupervised Twitter Robots

Dagny Stahl ◆ Luke Valentine ◆ Elise Renwick