Traffic and Weather Conditions in D.C.

Traffic and inclement weather tend not to mix. My goal is to observe interactions between the two, by attempting to determine which factors in the weather have the most significant impacts on the number of crashes.

Weather Data

- 2466 entries (2013-04 to 2019)
- Some of the data isn't particularly useful (but may be nice to have)
- AWND - Average wind speed
- PRCP - Precipitation
- SNWD - Snow Depth
- SNOW - Snowfall
- TAVG - Average temperature
- I originally gathered data from 2012-2019, but the 2012 data and some of the 2013 data was missing the predictors I was looking for, so I cut it out, along with repeat dates (since there are multiple observation areas, I just took the first unique one per day).

	STATION	NAME	LATITUDE	LONGITUDE	ELEVATION	DATE	AWND	PRCP	SNOW	SNWD	TAVG
0	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	39.1733	-76.684	47.5	2013-04-01	10.74	0.00	0.0	0.0	50.0
1	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	39.1733	-76.684	47.5	2013-04-02	8.28	0.00	0.0	0.0	40.0
2	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	39.1733	-76.684	47.5	2013-04-03	10.51	0.00	0.0	0.0	40.0
3	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	39.1733	-76.684	47.5	2013-04-04	4.25	0.04	0.0	0.0	37.0
4	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	39.1733	-76.684	47.5	2013-04-05	8.05	0.11	0.0	0.0	48.0
...	...	...	...	...	...	...	...	...	...	...	...
2461	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	NaN	NaN	NaN	2019-12-27	2.46	0.00	0.0	0.0	47.0
2462	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	NaN	NaN	NaN	2019-12-28	3.13	0.00	0.0	0.0	50.0
2463	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	NaN	NaN	NaN	2019-12-29	5.59	0.59	0.0	0.0	46.0
2464	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	NaN	NaN	NaN	2019-12-30	6.93	0.24	0.0	0.0	52.0
2465	USW00093721	BALTIMORE WASHINGTON INTERNATIONAL AIRPORT, MD US	NaN	NaN	NaN	2019-12-31	7.61	0.00	0.0	0.0	46.0

2466 rows × 11 columns

Weather Data (cont.)

- Dropped Latitude, Longitude and Elevation due to there being missing values in over half of the data
- Isolated one station to have a consistent reading per day, since crashes were “in D.C.” rather than in any specific area
- Dropped Name for a similar reason
- Dropped date, as it’s represented in the crash data

Crash Data

- Fully cleaned data
- Removed data prior to 2013/04/01 and after 2019/12/31
- Derives year/month/day from ReportDate
- Derives CrashCount through a groupby on ReportDate
- Removes 59 columns
- Dropped ReportDate, as it’s represented in individual columns

	ReportDate	CrashCount	Year	Month	Day
0	2013/04/01	49	2013	04	01
1	2013/04/02	35	2013	04	02
2	2013/04/03	49	2013	04	03
3	2013/04/04	38	2013	04	04
4	2013/04/05	60	2013	04	05
5	2013/04/06	49	2013	04	06
6	2013/04/07	52	2013	04	07
7	2013/04/08	64	2013	04	08
8	2013/04/09	59	2013	04	09
9	2013/04/10	70	2013	04	10
10	2013/04/11	55	2013	04	11
11	2013/04/12	49	2013	04	12
12	2013/04/13	53	2013	04	13
13	2013/04/14	58	2013	04	14
14	2013/04/15	52	2013	04	15
15	2013/04/16	46	2013	04	16
16	2013/04/17	62	2013	04	17
17	2013/04/18	64	2013	04	18
18	2013/04/19	73	2013	04	19
19	2013/04/20	47	2013	04	20
20	2013/04/21	40	2013	04	21
21	2013/04/22	52	2013	04	22
22	2013/04/23	46	2013	04	23
23	2013/04/24	47	2013	04	24
24	2013/04/25	65	2013	04	25

Merging Data

- Merged data
- Pulled data into spark using the following schema, resulting in the following dataframe

```
combined_schema = StructType([
    StructField('STATION', StringType()),
    StructField('NAME', StringType()),
    StructField('LATITUDE', FloatType()),
    StructField('LONGITUDE', FloatType()),
    StructField('ELEVATION', FloatType()),
    StructField('AWND', FloatType()),
    StructField('PRCP', FloatType()),
    StructField('SNOW', FloatType()),
    StructField('SNWD', FloatType()),
    StructField('TAVG', FloatType()),
    StructField('CRASHCOUNT', IntegerType()),
    StructField('YEAR', IntegerType()),
    StructField('MONTH', IntegerType()),
    StructField('DAY', IntegerType())
])
```

AWND	PRCP	SNOW	SNWD	TAVG	CRASHCOUNT	YEAR	MONTH	DAY
10.74	0.0	0.0	0.0	50.0	49	2013	4	1
8.28	0.0	0.0	0.0	40.0	35	2013	4	2
10.51	0.0	0.0	0.0	40.0	49	2013	4	3
4.25	0.04	0.0	0.0	37.0	38	2013	4	4
8.05	0.11	0.0	0.0	48.0	60	2013	4	5
5.59	0.0	0.0	0.0	47.0	49	2013	4	6
5.82	0.0	0.0	0.0	50.0	52	2013	4	7
4.92	0.0	0.0	0.0	64.0	64	2013	4	8
6.71	0.0	0.0	0.0	73.0	59	2013	4	9
7.61	0.0	0.0	0.0	76.0	70	2013	4	10
9.62	0.0	0.0	0.0	72.0	55	2013	4	11
10.29	0.67	0.0	0.0	55.0	49	2013	4	12
5.14	0.0	0.0	0.0	54.0	53	2013	4	13
5.59	0.0	0.0	0.0	54.0	58	2013	4	14
8.28	0.0	0.0	0.0	56.0	52	2013	4	15
6.04	0.0	0.0	0.0	60.0	46	2013	4	16
5.82	0.0	0.0	0.0	65.0	62	2013	4	17
9.17	0.04	0.0	0.0	63.0	64	2013	4	18
7.83	0.79	0.0	0.0	66.0	73	2013	4	19
10.29	0.07	0.0	0.0	54.0	47	2013	4	20
4.92	0.0	0.0	0.0	47.0	40	2013	4	21
11.18	0.0	0.0	0.0	46.0	52	2013	4	22
6.26	0.0	0.0	0.0	51.0	46	2013	4	23
6.93	0.0	0.0	0.0	56.0	47	2013	4	24
7.16	0.0	0.0	0.0	58.0	65	2013	4	25
4.25	0.0	0.0	0.0	55.0	55	2013	4	26
4.47	0.0	0.0	0.0	55.0	44	2013	4	27
5.37	0.0	0.0	0.0	57.0	54	2013	4	28
7.16	0.26	0.0	0.0	56.0	66	2013	4	29
7.83	0.22	0.0	0.0	56.0	62	2013	4	30

Linear Regression

- To see how each of the features (average wind speed, precipitation, snow depth, snowfall, average temperature, year, month, day) affected the crash count, I created a linear regression model
- Created a vector of the data, and separated my target column from the the features
- Split the data 80/20 training/testing
- Created the model using the vector

Results

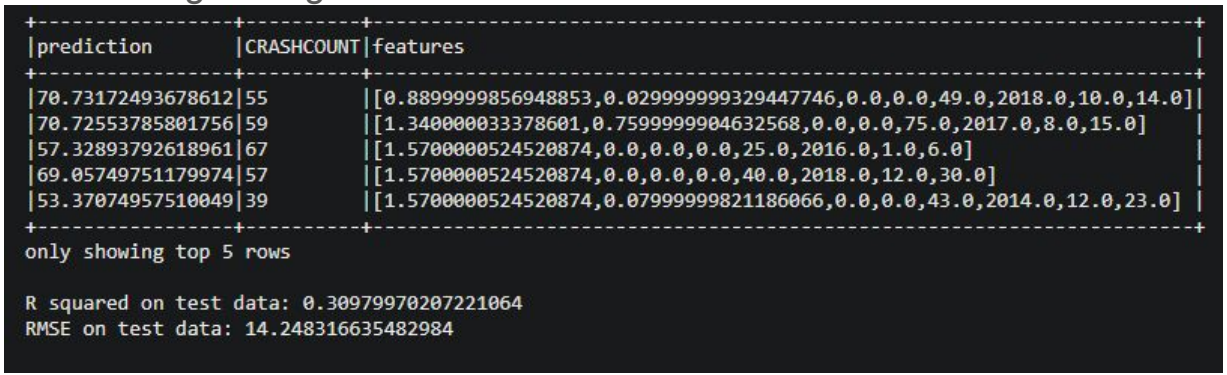
- Coefficients per feature
  - Each feature has either a negative or positive relationship with the target variable, causing the model to predict higher or lower, depending on the coefficient
- RMSE of 13.38
- R^2 (or, the amount of variation in crashes that can be explained by the model, was .32, which would say the correlation is low, given my features

```
Coefficients per feature:
AWND: 0.36195599060329114
PRCP: 0.0
SNOW: -0.7022002992185691
SNWD: -0.5092373215720103
TAVG: 0.16902245624939513
YEAR: 4.133793415842702
MONTH: 0.1905461293696973
DAY: -0.04876547970346394

RMSE: 13.381737643659294
R squared: 0.32237870593192286
```

Results (cont.)

- Pictured below is the first 5 results of predictions on the testData, with corresponding statistics on the predictions
  - R^2 is slightly lower, but similar to the trainingData, which means the model is consistent (consistently poor, but…)
  - Similar RMSE as well
  - Overall, slightly worse when using testing data

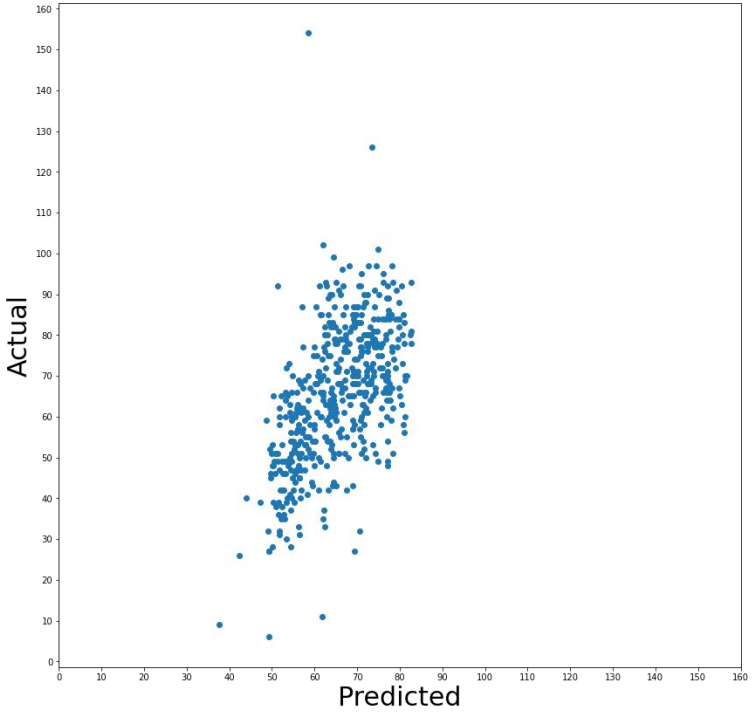


```
+-----------------+----------+--------------------------------------------------------------------------+
|prediction       |CRASHCOUNT|features                                                                  |
+-----------------+----------+--------------------------------------------------------------------------+
|70.73172493678612|55        |[0.8899999856948853,0.029999999329447746,0.0,0.0,49.0,2018.0,10.0,14.0]|
|70.72553785801756|59        |[1.340000033378601,0.7599999904632568,0.0,0.0,75.0,2017.0,8.0,15.0]    |
|57.32893792618961|67        |[1.5700000524520874,0.0,0.0,0.0,25.0,2016.0,1.0,6.0]                      |
|69.05749751179974|57        |[1.5700000524520874,0.0,0.0,0.0,40.0,2018.0,12.0,30.0]                    |
|53.37074957510049|39        |[1.5700000524520874,0.07999999821186066,0.0,0.0,43.0,2014.0,12.0,23.0]    |
+-----------------+----------+--------------------------------------------------------------------------+
only showing top 5 rows

R squared on test data: 0.30979970207221064
RMSE on test data: 14.248316635482984
```

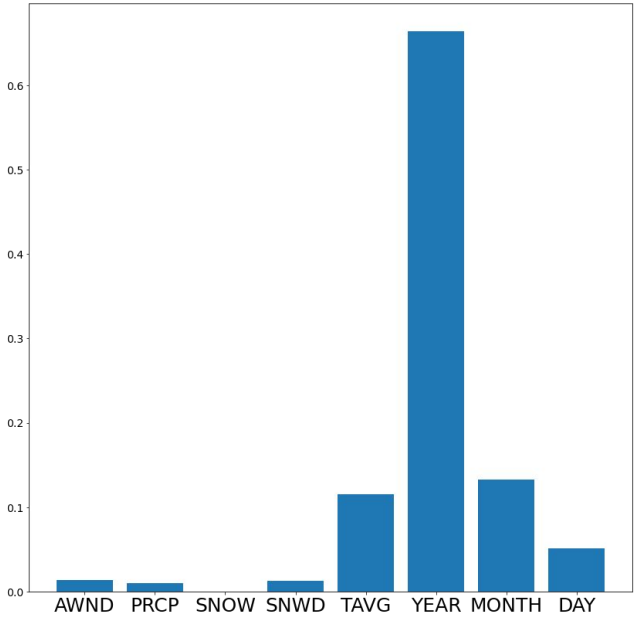
Results (cont.)

- Graphed predicted crashes vs actual crashes using the same scaling on axes
- If the model was good, we would expect to see something relatively close to a straight line from the bottom left to top right. As it is, it seems like the model is underfitting, since the predictions are relatively localized compared to the actual



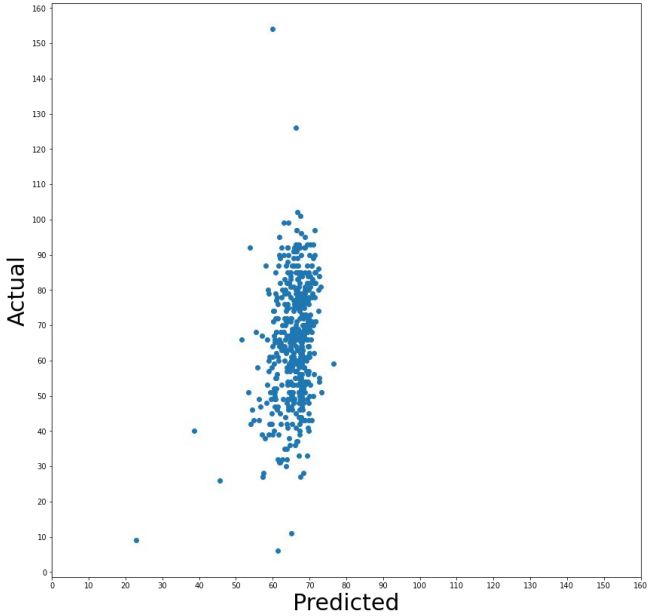
Results(cont.)

- Then, I decided to use a decision tree regressor to look at feature importances
- This result is strange - year is by far the best predictor in this dataset. This could possibly be explained by specific years having extreme incidents.



Results(cont.)

- Given that year was such a strong predictor for the model, I removed it to see what happened
- As may be expected, it results in far more extreme underfitting
  - increased RMSE by 2 points on both testing and training data
  - Reduced R^2 to .067 on training and .075 on testing, so an incredibly poor model



Conclusions

- It seems that this data cannot be used to effectively predict crashes
  - Possibly because driving habits change with worse weather?
- We would either need more, or better, data to determine if these features are truly poor predictors for crash count, given that my model was underfitting
- No effective conclusions can be drawn other than the need to look more deeply into what data there is, if there is something to be learned

Solution

- Determine which of the factors most influence crashes by creating a model to predict crashes
- Plot scores of each feature side by side