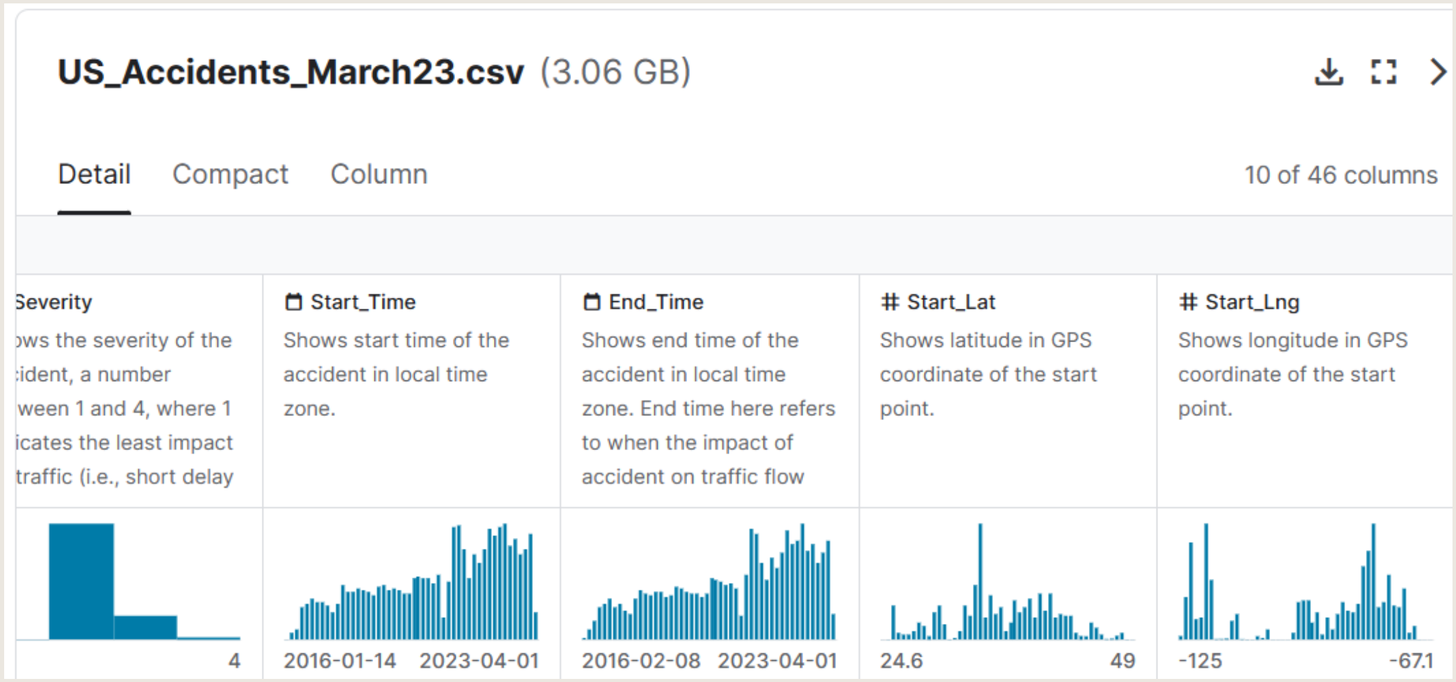
Matias OTTENSEN
Tiphaine KACHKACHI
Manon GARDIN

Explainability AI TD4 Presentation

Le15 avril 2020

Our Dataset

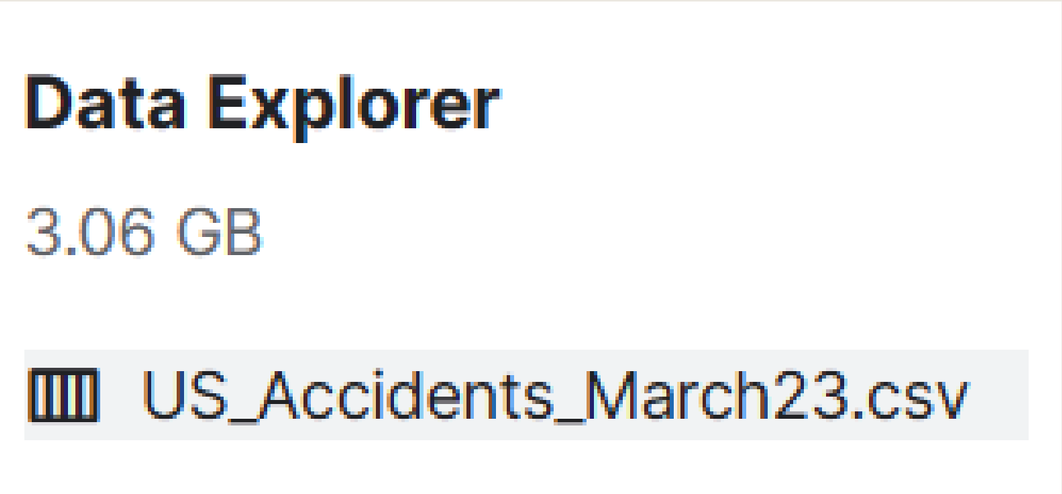


- Accidents US
- 46 Columns
- very large width

Specificity

Size: 3 GB

7 Million lines




```
df.shape
```

```
(7728394, 46)
```

Good exercice : errors / unoptimized code =
very long execution time

DATA CLEANING

first: drop columns

```
# Delete the columns we don't need
df = df.drop(['Street', 'City', 'Description','County', 'Zipcode', 'Country', 'Bump', '
```

DATA CLEANING

Missing values

```
Entrée [11]: print(df.isnull().sum())
```

```
ID                          0
Source                      0
Severity                    0
Start_Time                  0
End_Time                    0
Start_Lat                   0
Start_Lng                   0
End_Lat               3402762
End_Lng               3402762
Distance(mi)                0
State                       0
Timezone                 7808
Weather_Timestamp      120228
Temperature(F)         163853
Wind_Chill(F)         1999019
```

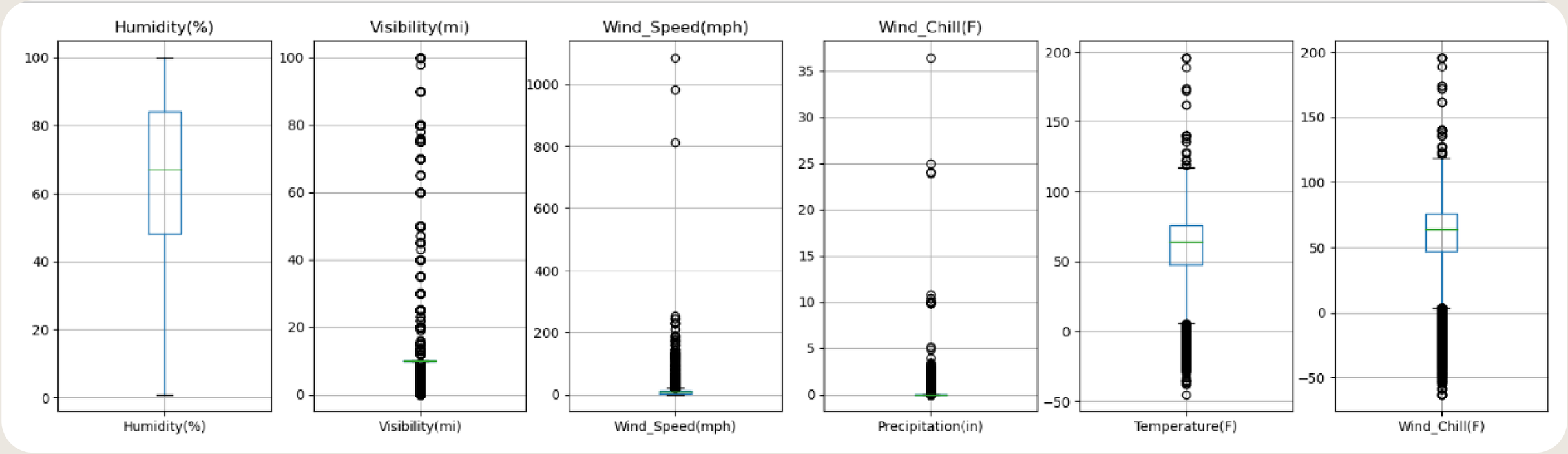
DATA CLEANING

Missing values

```
.fillna(newdf['Start_Lat'])

fillna('unknown')

fillna(newdf['Temperature(F)'].mean())

fillna('###')
```

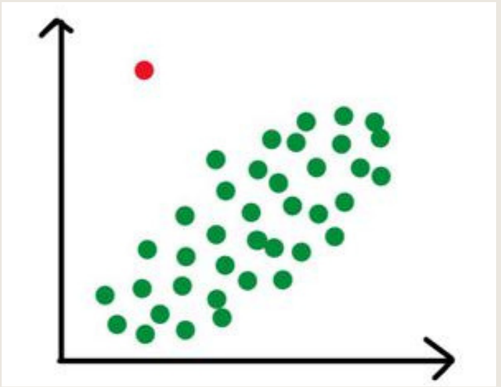
DATA CLEANING

OUTLIERS



DATA CLEANING

OUTLIERS

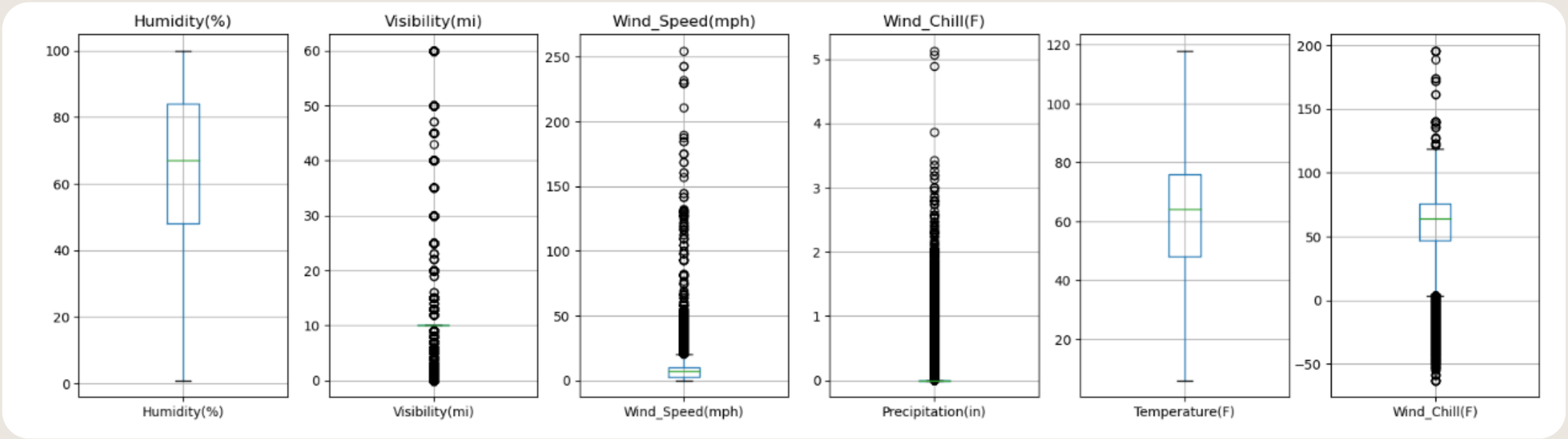


Outlier cleaning sample:

```
Q1= newdf['Wind_Speed(mph)'].quantile(0.25)
Q3= newdf['Wind_Speed(mph)'].quantile(0.75)
IQR = Q3 - Q1
moyenne = newdf['Wind_Speed(mph)'].mean()

threshold = 1.5

newdf.loc[(df['Wind_Speed(mph)'] < Q1 - threshold * IQR), 'Wind_Speed(mph)'] = moyenne
newdf.loc[(df['Wind_Speed(mph)'] > Q3 + threshold * IQR), 'Wind_Speed(mph)'] = moyenne
```

DATA CLEANING

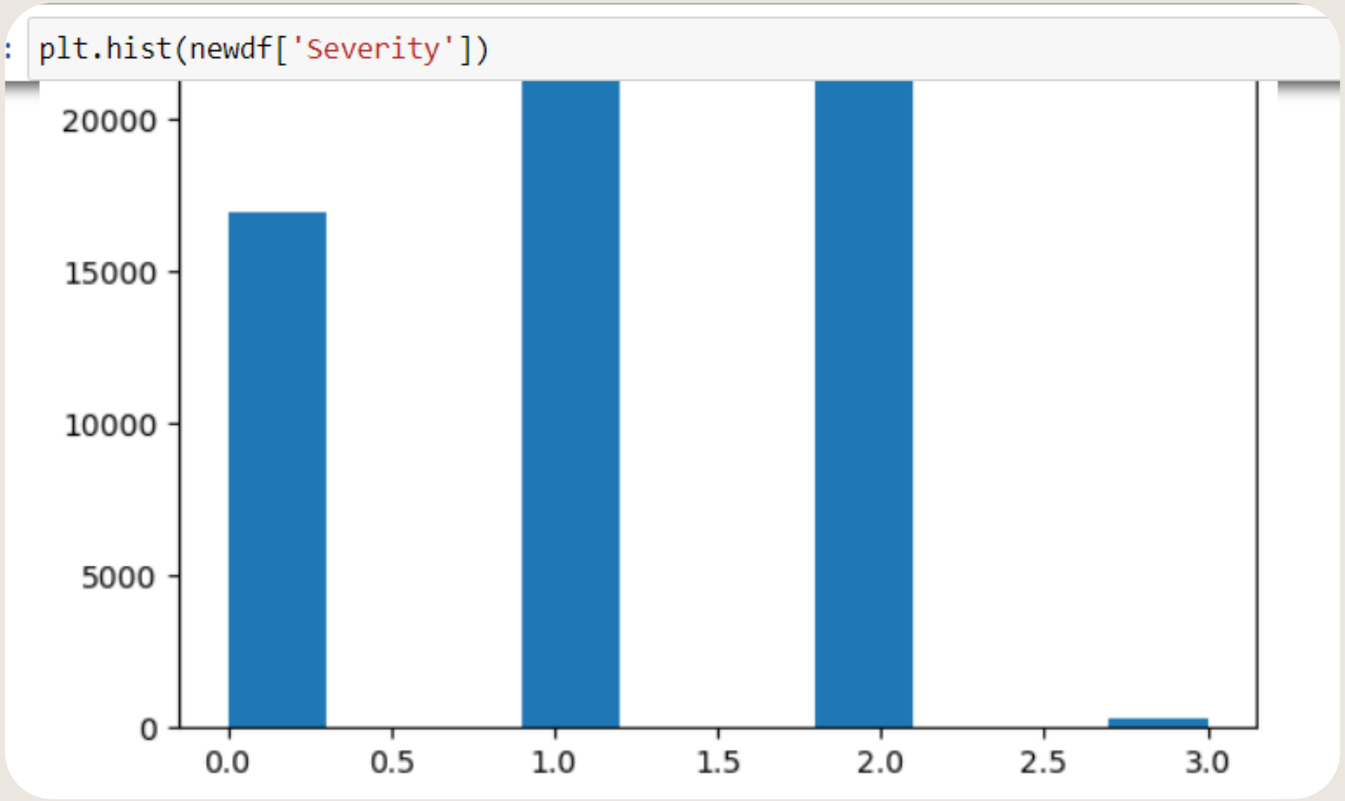
OUTLIERS - After data cleaning



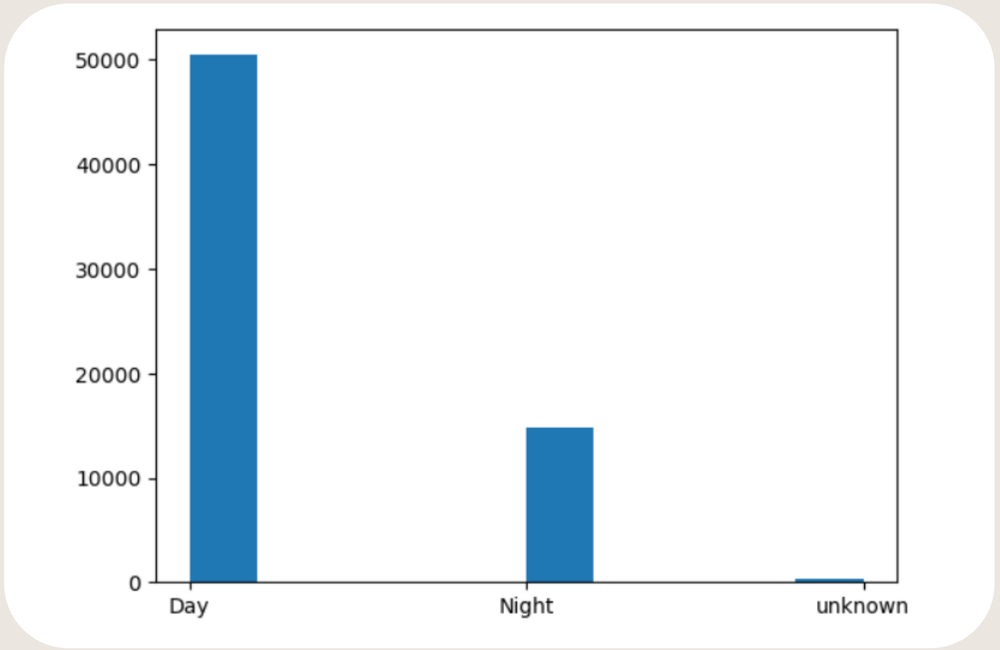
Variable choice

Severity

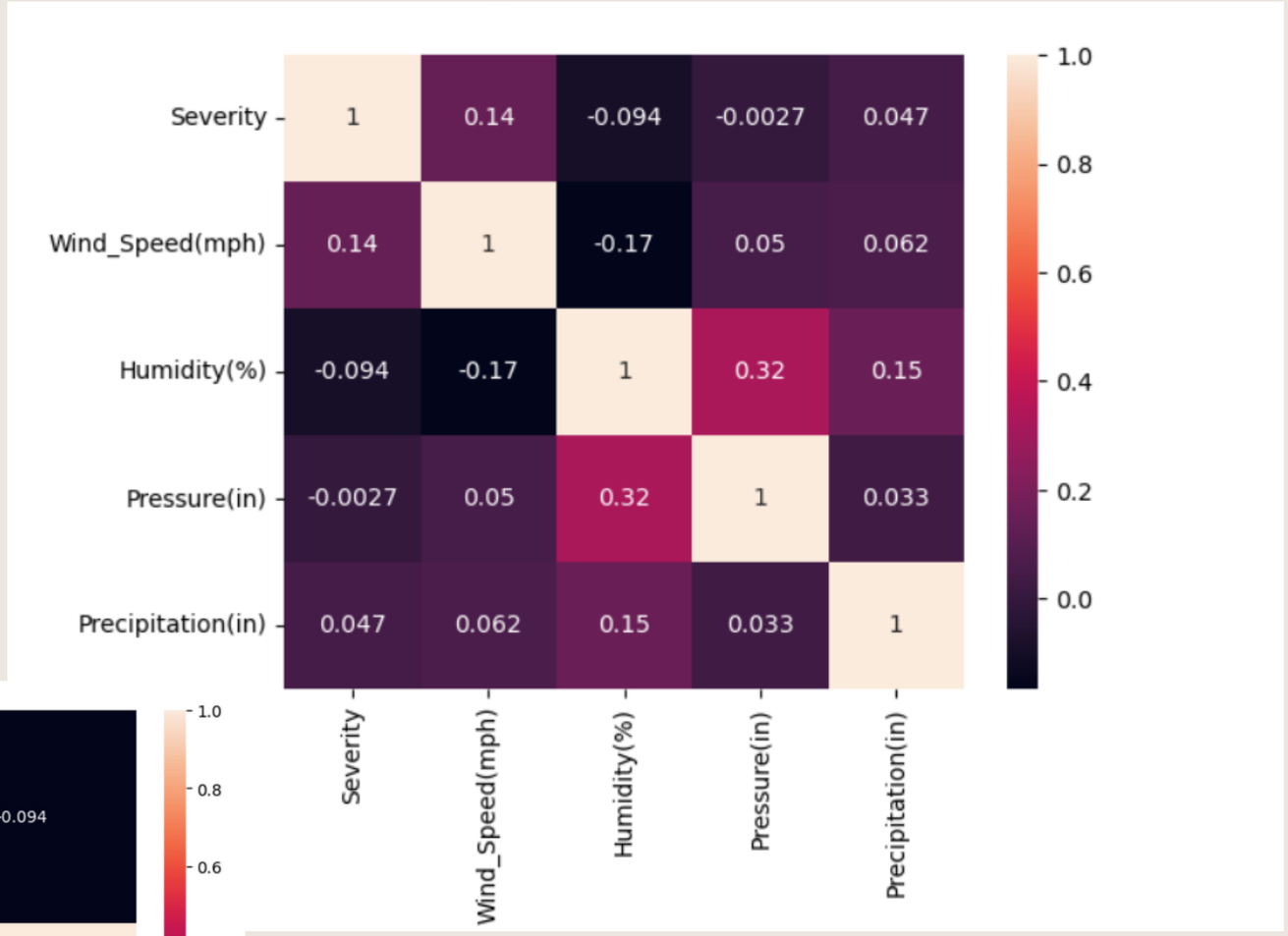
```
plt.hist(newdf['Severity'])
```

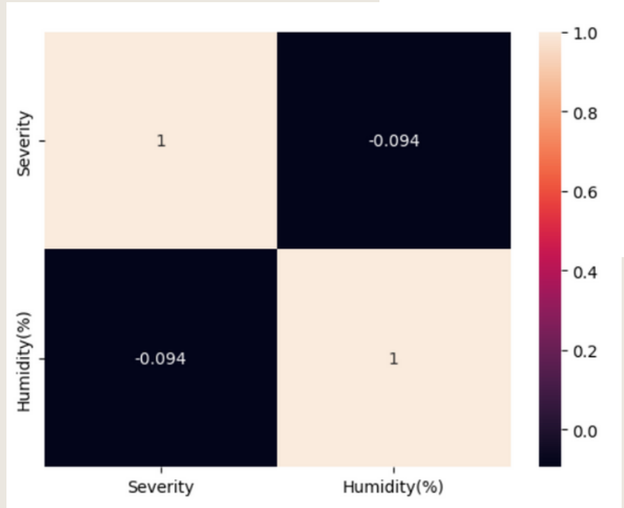


Time of the day



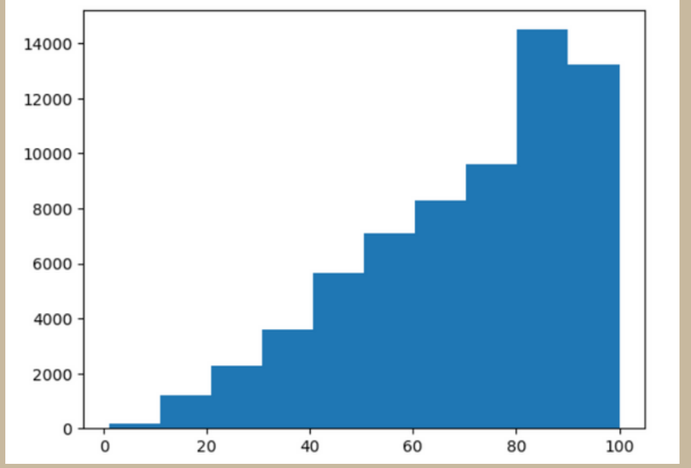
Other Graphics





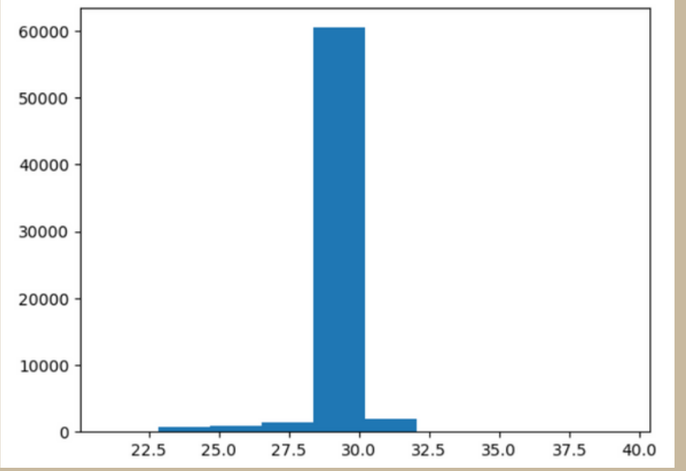

```
plt.hist(newdf['Humidity(%)'])
```

```
(array([  192.,  1209.,  2256.,  3587.,  5660.,  7079.,  8292.,  9607.,
        14489., 13213.]),
 array([  1. ,  10.9,  20.8,  30.7,  40.6,  50.5,  60.4,  70.3,  80.2,
         90.1, 100. ]),
 <BarContainer object of 10 artists>)
```




```
plt.hist(newdf['Pressure(in)'])
```

```
(array([2.7000e+01, 7.2500e+02, 9.4100e+02, 1.4690e+03, 6.0432e+04,
        1.9890e+03, 0.0000e+00, 0.0000e+00, 0.0000e+00, 1.0000e+00]),
 array([20.98 , 22.827, 24.674, 26.521, 28.368, 30.215, 32.062, 33.909,
        35.756, 37.603, 39.45 ]),
 <BarContainer object of 10 artists>)
```



LINEAR REGRESSION

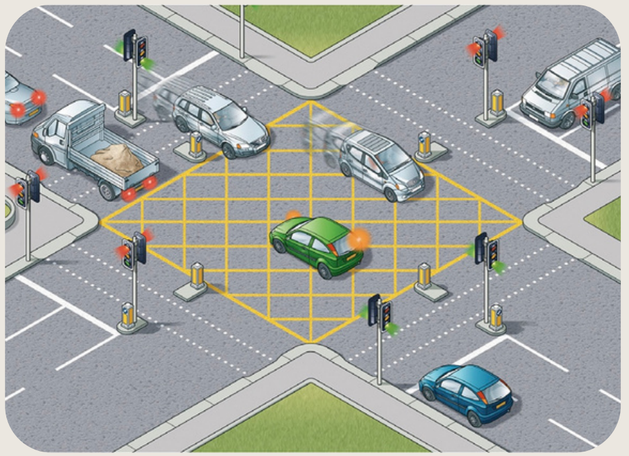
Variable choice

Weather-related

- Visibility (mi)
- Weather condition*
- Precipitation (in)

Environment-related

- Crossing (bool)
- Junction (bool)
- State*
- Stop (bool)
- Traffic Signal (bool)
- Station (bool)



Target : Severity

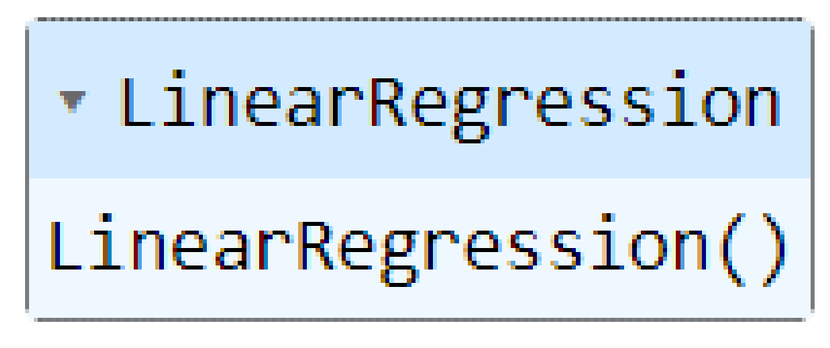
*Needs hot-encoding

LINEAR REGRESSION

Our results

```
Training R^2 score: 0.24430378517137308
Testing R^2 score: 0.23833267491677146
```

Training score : Measures how well the model fits the training data.
Testing score : Prediction of new, unseen data



Low score for both Training and Testing --> maybe there is a non-linear relationship on Severity

LINEAR REGRESSION

Coefficients

```
Intercept: 0.7858510659522875
```

Value of Severity if all independant variables as set to 0

Our coefficients with the highest values :

```
Coefficient for State_WV: 0.9754979967003913
Coefficient for State_WY: 1.859076752819203 1
Coefficient for Weather_Condition_Light Thunderstorms and Rain: 0.7000497
Coefficient for Weather Condition Light Thunderstorms and Snow: 1.2477980
```

Complex Model - XgBoost

Why XgBoost ?

--> Series of decision trees, non-linear

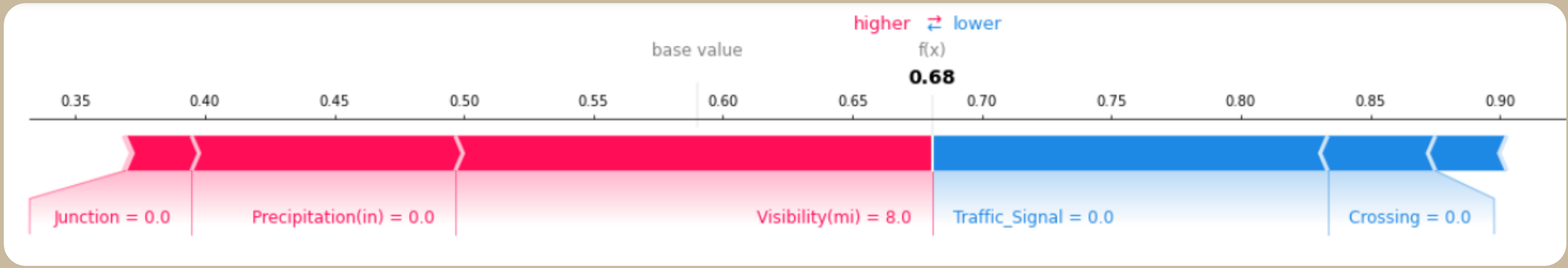
XGBClassifier

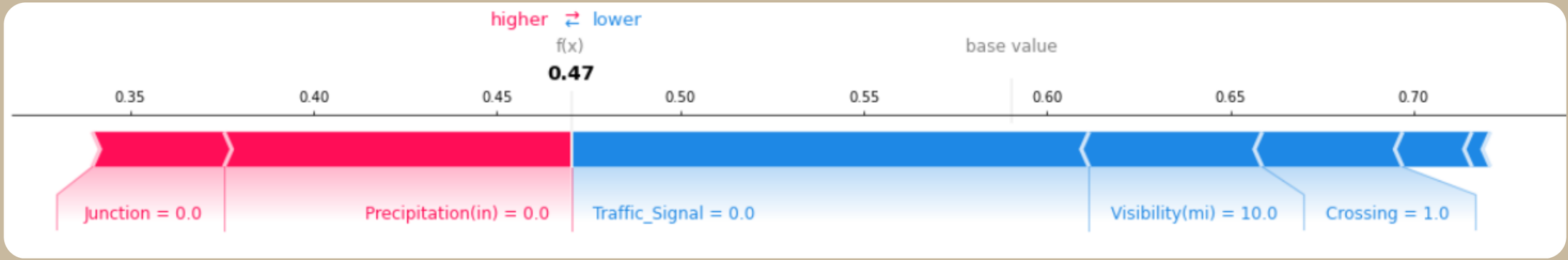
```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, early_stopping_rounds=None,
              enable_categorical=False, eval_metric=None, feature_types=None,
              gamma=None, gpu_id=None, grow_policy=None, importance_type=None,
              interaction_constraints=None, learning_rate=None, max_bin=None,
              max_cat_threshold=None, max_cat_to_onehot=None,
              max_delta_step=None, max_depth=None, max_leaves=None,
              min_child_weight=None, missing=nan, monotone_constraints=None,
              n_estimators=100, n_jobs=None, num_parallel_tree=None,
```

Accuracy: 0.47244034459098877

Shapley values

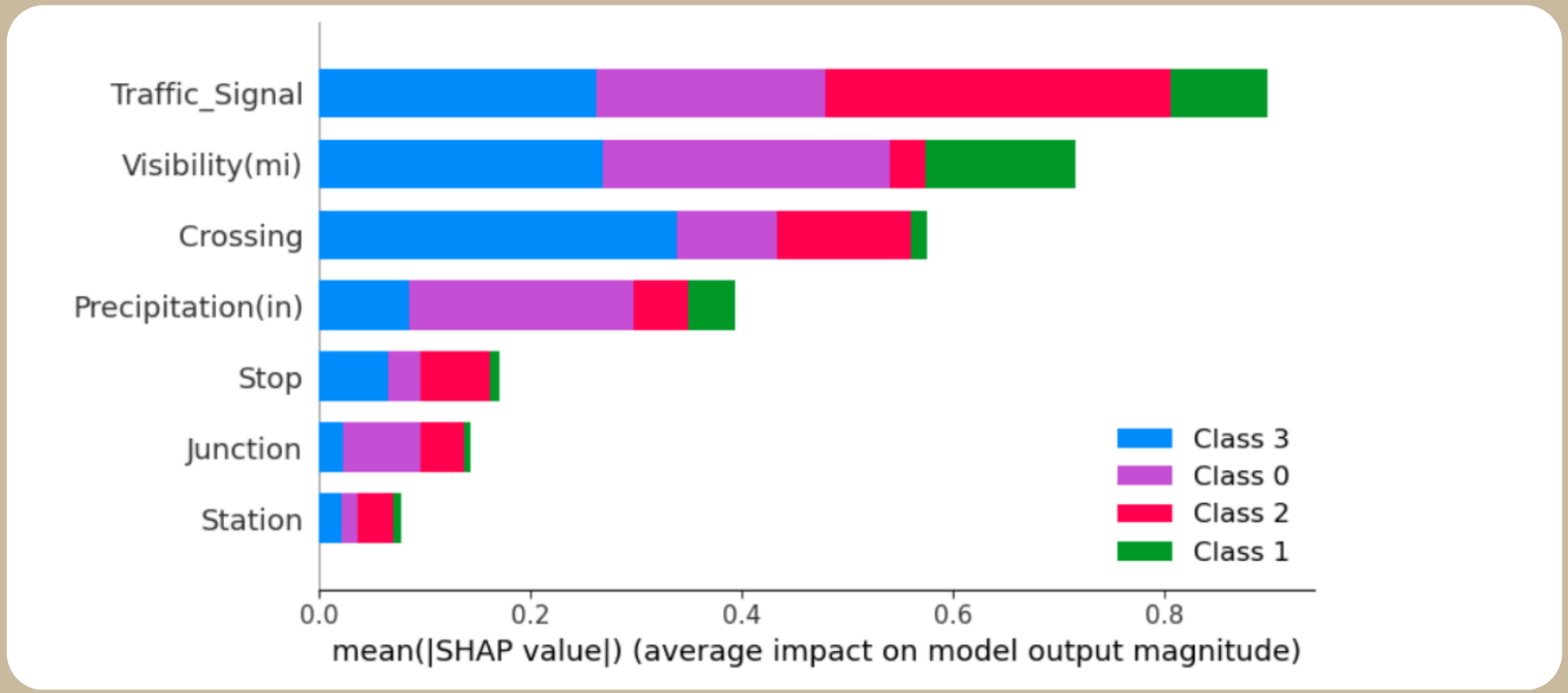
Examples of forceplot





Shapley values

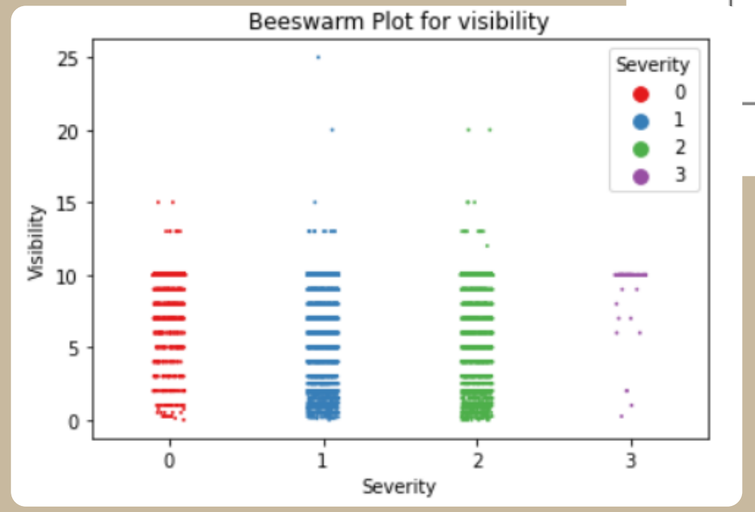
Identify important variables

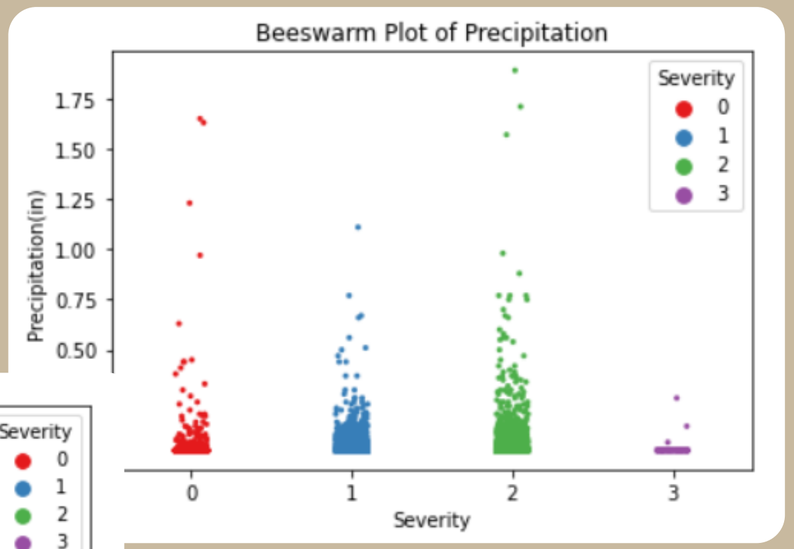


Shapley values

Beeswarm plots

Binary values -> impossible to represent in a stripplot

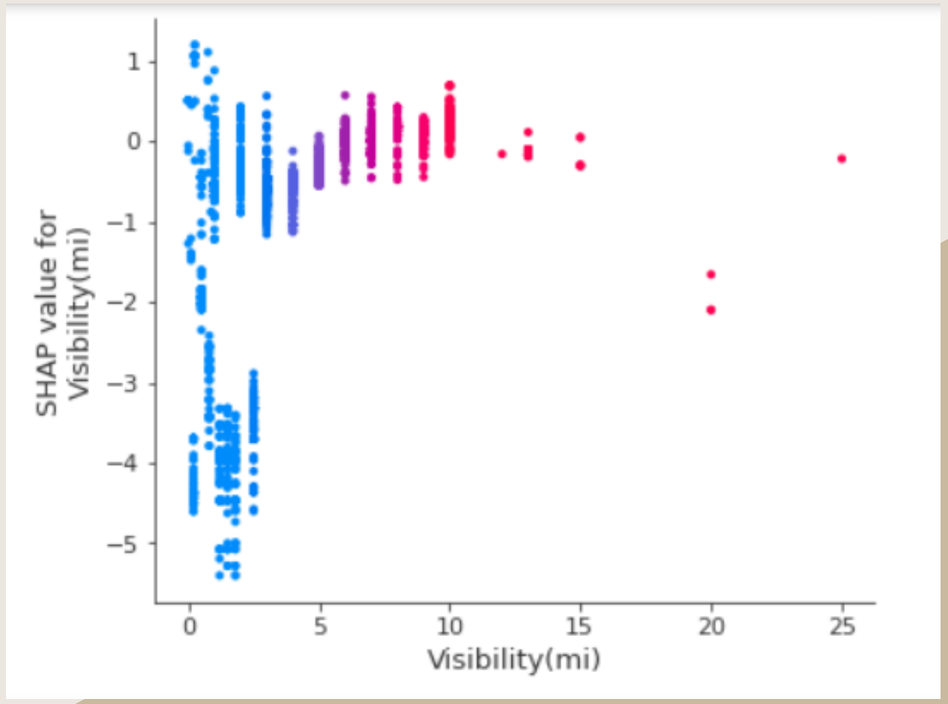




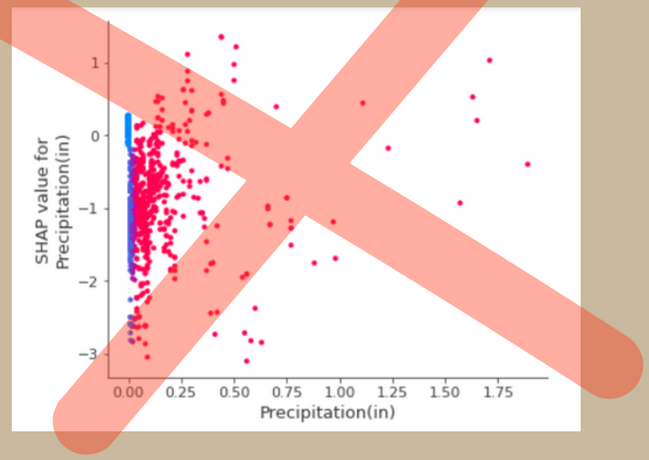
Shapley values

Dependance plots

Dependance plot of Visibility



Dependance plot of Precipitation



Clustering of Shapley values