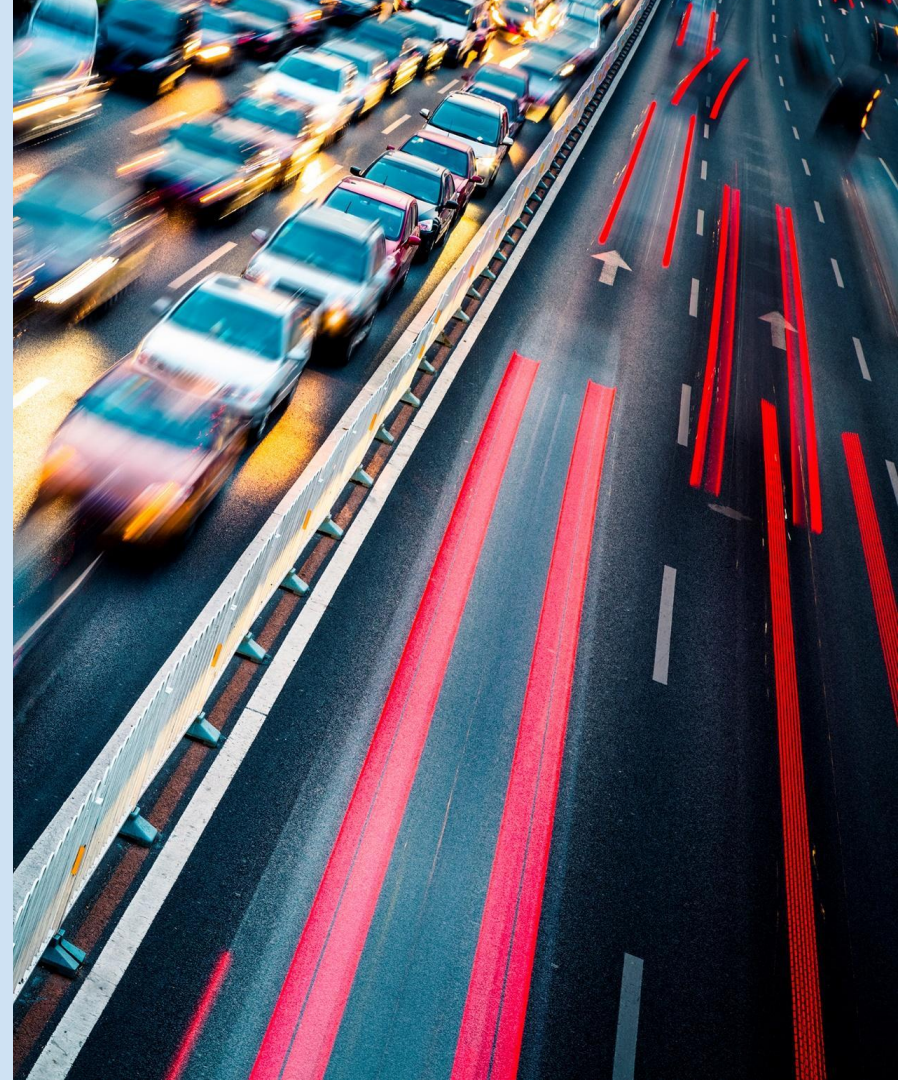


# TRAFFIC ACCIDENTS IN THE UK 2021:

*Investigating traffic accident  
seriousness in the UK and the  
variables that are more likely  
to lead to a car collision*

Ingrid Ionita, Clarissa Lo, Michelle  
Obonyano & Ayomide Olarewaju



# Task distribution



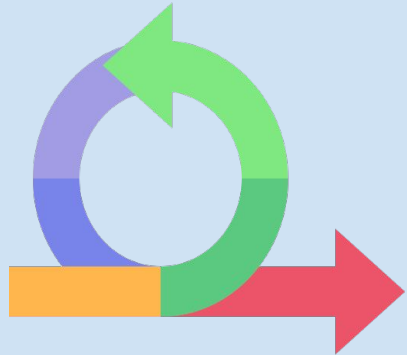
NAME	ROLE
Ingrid Ionita	Scrum master, Main data cleaner, Main analyst
Clarissa Lo	Lead visualisation, Main analyst
Ayomide Olarewaju	Lead data cleaner, Lead machine learning
Michelle Obonyano	Main analyst, Main visualisation

# *What were our challenges?*

## **Communication issues:**

- Time zone differences
- Personal responsibilities outside of the CFGdegree (childcare, university, work)

## **Difficulty on deciding on a project topic**



*Agile approach*

# *How did we overcome them?*

- Daily scrum meetings
- Ensuring scrum master met with everyone
- Scrum master representing absent team members
- Listening to opinions to include everyone's voice
- Trusted each other

# How did we make it work for us?

Trello board



Organise tasks

Google Drive

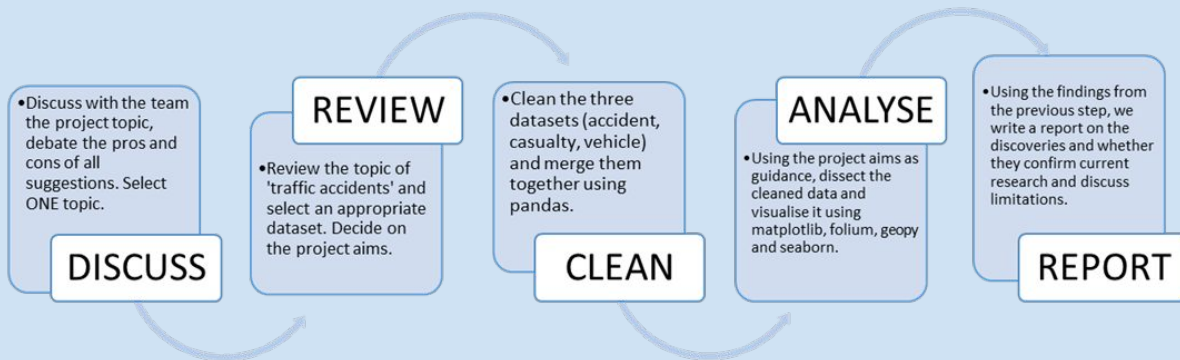


Organise documents

Slack



Communicate



# Our project: traffic accidents

Why did we choose the  
topic of traffic accidents?

There are approximately  
150,000  
people caught in a traffic accident per year.

Our project aims are:

To discover the circumstances in  
which traffic accidents are more  
likely to happen

To offer recommendations to the  
UK government in reducing the  
number of traffic accidents.

# Objectives



01 **Are certain drivers or casualties more likely to get involved in a severe car accident?**

- *These drivers and casualties would be categorised by their age and gender.*

02 **What dates are more likely to have a higher number of traffic accidents and cause more severe accidents?**

- *Can this be linked to environmental factors such as the weather and the light conditions?*

03 **What areas in the UK are more likely to have more severe traffic accidents?**

- *This can be from the road type, junction location, county.*

04 **Extension question: can machine learning be used to predict traffic accidents?**

# Road Safety Data

**Published by:** Department for Transport  
**Last updated:** 15 October 2022  
**Topic:** Transport  
**Licence:** [Open Government Licence](#)

## Summary

Road Safety Statistics [releases](#) and [guidance](#) about the data collection.

[Data download tool](#) for bespoke breakdowns of our data.

[STATS19 R package](#) developed independently of DfT, offering an alternative way to access this data for those familiar with the R language.

[View full summary.](#)

# Data sources

Our sources are from the UK government!

## Data links

### Link to the data

	Format	File added	Data preview
<a href="#">Road Safety Data - Casualty Adjustment Lookup 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Collision Adjustment Lookup 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - E-Scooter 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Casualties 1979 - 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Vehicles 1979 - 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Accidents 1979 - 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Casualties 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Vehicles 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Accidents 2021</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Accidents 2021 - Provisional Mid Year Unvalidated Data</a>	CSV	15 October 2022	<a href="#">Preview</a>
<a href="#">Road Safety Data - Vehicles 2021 - Provisional Mid Year Unvalidated Data</a>	CSV	25 November 2021	<a href="#">Preview</a>
<a href="#">Road Safety Data - Casualties 2021 - Provisional Mid Year Unvalidated Data</a>	CSV	25 November 2021	<a href="#">Preview</a>
<a href="#">Road Safety - E-Scooter 2021 - Provisional Mid Year Unvalidated Data</a>	CSV	25 November 2021	<a href="#">Preview</a>
<a href="#">Road Safety - E-Scooter 2021 - Provisional Mid Year Unvalidated Data</a>	CSV	25 November 2021	<a href="#">Preview</a>



# Data cleaning

Replacing missing /out of range data represented by (-1) with mode and mean while columns with more than 40% were dropped. This was done to prevent our data from being skewed.

## REMOVING DULPICATES

```
1 n_duplicates =df.drop(labels=["accident_index"], axis=1).duplicated().sum()  
2 print(f"You might have {n_duplicates} duplicates in your database.")
```

You might have 0 duplicates in your database.

## DETECTING COLUMNS WITH MISSING VALUES

```
1 #checking how many NaN values each column contains.  
2  
3 missing_val= df.isnull().sum().sort_values(ascending=False)  
4 percent_missing = ((missing_val/df.isnull().count()*100).sort_values(ascending=False)  
5 missing_df = pd.concat([missing_val,percent_missing], axis=1, keys=['Total', '%'],sort=False)  
6 missing_df[missing_df['Total']>=1]
```

	Total	%
location_easting_osgr	39	0.025087
location_northing_osgr	39	0.025087
longitude	39	0.025087
latitude	39	0.025087



# Drivers and casualties in a traffic accident

Pivot table:

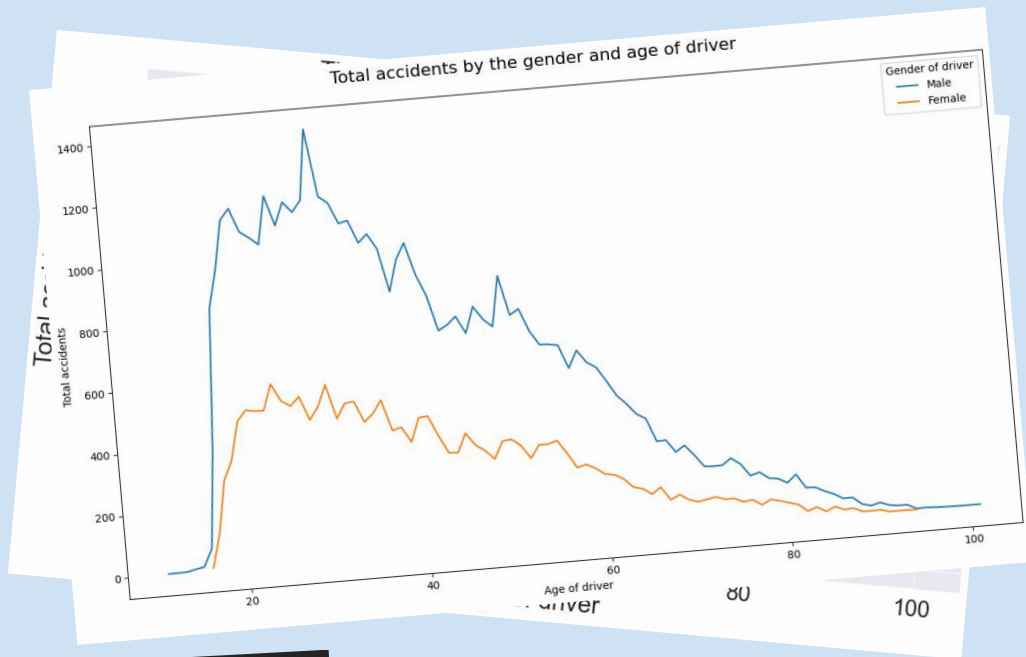
```
PIVOT_TBL_1b= GEN_AGE_AF.pivot_table(  
    index = 'age_of_casualty',  
    columns="sex_of_casualty",  
    aggfunc = "count")['casualty_severity']
```

PIVOT\_TBL\_1b

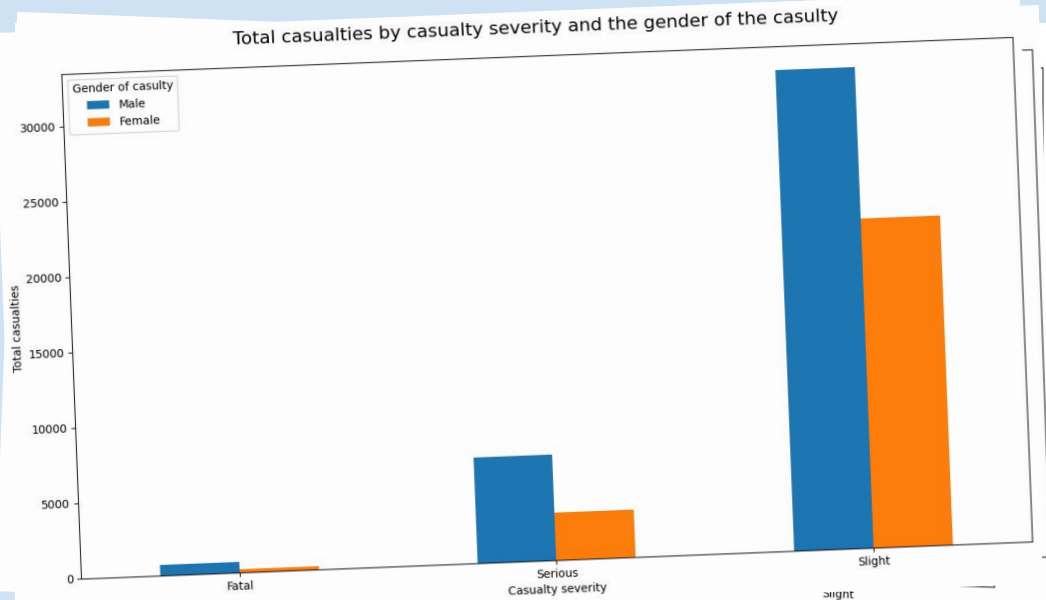
Line chart:

```
plt.plot(  
    PIVOT_TBL_1b.plot(kind="line", figsize=(15,8))  
  
    # x-axis  
    plt.xlabel('Age of casualty')  
  
    # y-axis  
    plt.ylabel('Total casualties')  
)
```

```
# other elements of the plot  
plt.title('Total casualties by the gender and age of casualty', pad=20, fontsize=16)  
plt.legend(labels=["Male", "Female"], title="Gender of casualty", loc="upper right")  
plt.show()
```



# Drivers and casualties in a traffic accident



Pivot table:

```
PIVOT_TBL_1e = GEN_AGE_AF.pivot_table(  
    index = 'casualty_severity',  
    columns="sex_of_casualty",  
    aggfunc = "count")['age_of_casualty']
```

```
P PIVOT_TBL_1e
```

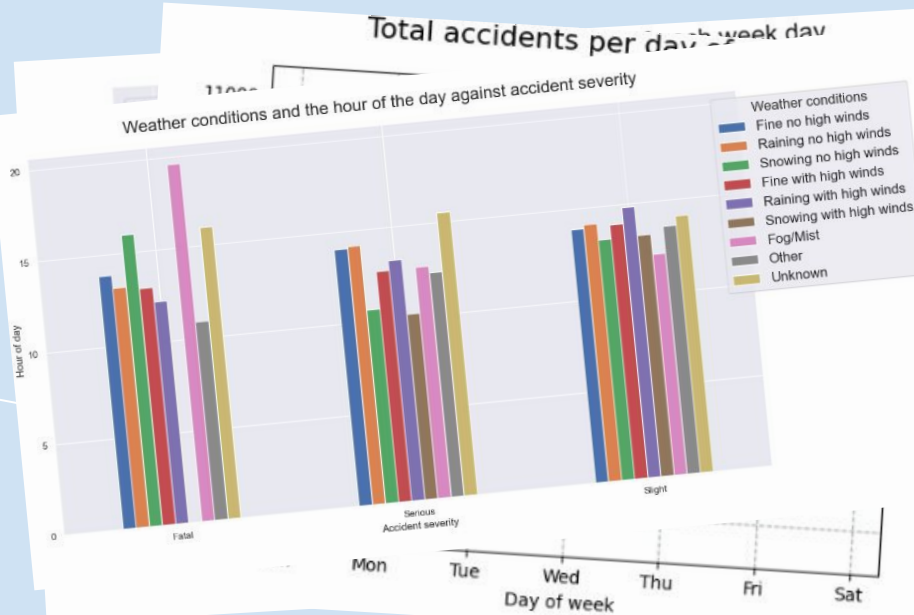
Line chart:

```
# bars  
PIVOT_TBL_1e.plot(kind="bar", figsize=(15,8))  
  
# x-axis  
plt.xlabel('Casualty severity')  
plt.xticks(range(0,3), labels=(severity), rotation = 'horizontal')  
  
# y-axis  
plt.ylabel('Total casualties')
```

```
# other elements  
plt.title('Total casualties by casualty severity and the gender of the casualty', pad=20, fontsize=16)  
plt.legend(title="Gender of casualty", labels=["Male", "Female"], loc="upper left")  
plt.show()  
plt.show()
```

# Traffic accidents and the time

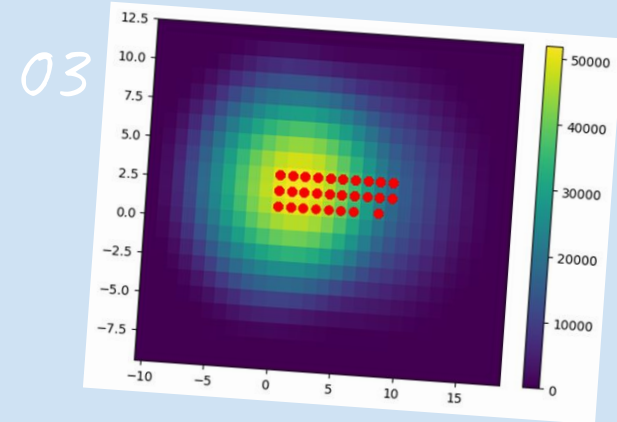
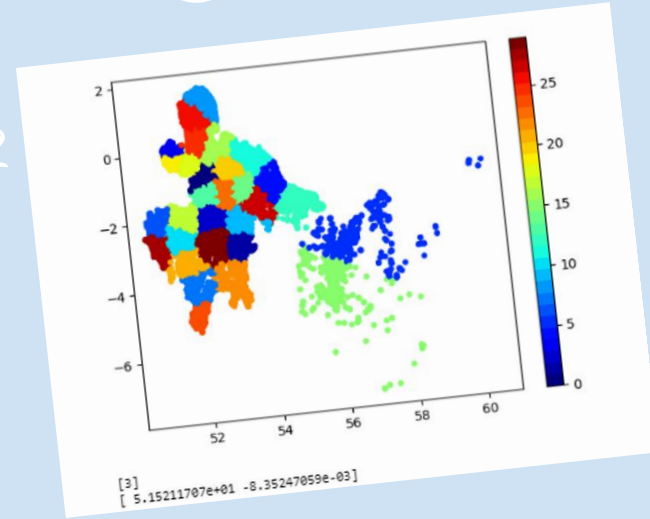
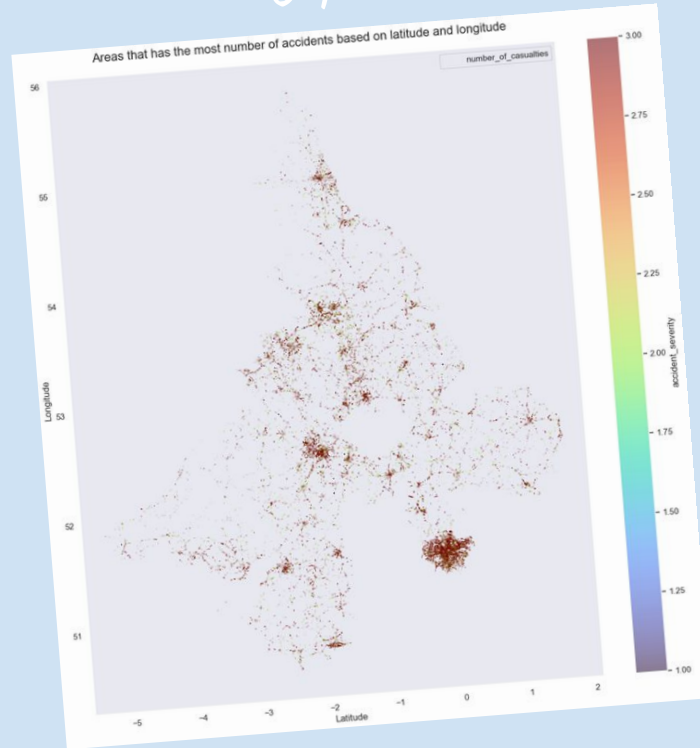
- Could be linked to fatigue related to work hours
- Wednesday, Thursday and Friday have the most traffic accidents
- Weather and light conditions variables explored, but no interesting findings resulted from it



```
## pivot table of week day and hour count ##  
  
# make a new dataframe for variables required for the pivot table  
DAY_HOUR_MONTH_DF = DF[['day_of_week', 'casualty_severity', 'wthr', 'light']]  
  
# count the number of accidents by accident severity and weather conditions ##  
  
# make a new data frame with variables required  
WTHR_LIGHT = DF[['hour', 'accident_severity', 'weather_conditions', 'light_conditions']]  
  
# make a list of the labels for each weather condition  
wthr_labels = ["Fine no high winds", "Raining no high winds", "Snowing no high winds", "Fine with high winds", "Raining with high winds", "Snowing with high winds", "Fog/Mist", "Other", "Unknown"]  
  
# pivot table with accident severity as index and weather conditions as columns  
PIVOT_TBL_2c = WTHR_LIGHT.pivot_table(index = 'accident_severity', columns = 'weather_conditions')[['hour']]  
  
## plot the number of accidents by accident severity and weather conditions ##  
  
# bar  
PIVOT_TBL_2c.plot(kind="bar", figsize=(15,8))  
  
# x-axis  
plt.xlabel('Accident severity')  
plt.xticks(range(0,3), labels=(severity), rotation = "horizontal")  
  
# y-axis  
plt.ylabel('Hour of day')  
plt.yticks(range(0,24,5), rotation = "horizontal")  
  
# other elements  
plt.title('Weather conditions and the hour of the day against accident severity', pad=20, fontsize=20)  
plt.legend(title="Weather conditions", title_fontsize='16', labels = wthr_labels, fontsize='16', loc="upper right", bbox_to_anchor=(1.25, 1))  
  
# elements of the plot  
plt.title('The total accidents by the hour of each week day', pad=20, fontsize=20)  
plt.legend(labels=days, title="Days of the week", loc="upper left")
```

# Traffic accidents and the location

1. The darker area on the graph and where its showing on the x and y axis which is -1, +1 and 51, 52
2. The figures with exponents of Ten [  $5.15211707e+01$  -  $8.35247059e-03$  ] multiplied by the required tens to get the actual longitude and latitude 51.5211707, and longitude -0.00835247 which turns out to be London
3. Junction 0-9 has high number of accident severity but higher number of accident severity with added location 10





# Nominatim API

- Nominatim's API was used to produce heat maps
- Many errors in the process
- Marker map and heatmap not working initially because too many API recalls
- Group effort solved the problem

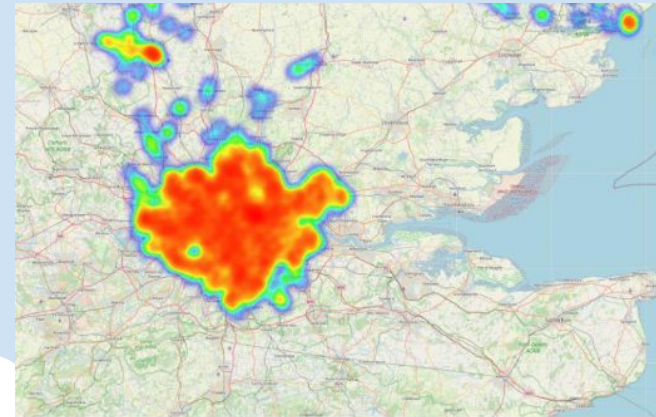
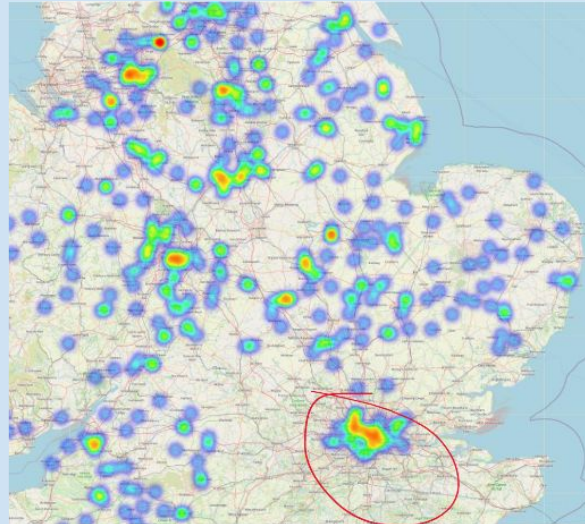
```
# Getting rid of the other values, fatal and serious, to only keep the fatal accidents.  
dv3 = [1, 2]  
SLIGHT = DF[DF.accident_severity.isin(dv3) == False]
```

```
# Creating a new map variable for the serious accidents.  
SLIGHT_UK = folium.Map(location = [54.76999101318324, -2.8478385244334445], zoom_start = 6)
```

```
# Using the Lambda function to combine the Latitude and Longitude into one column.  
slight_latlon = SLIGHT.apply(lambda row: (str(row.latitude),str(row.longitude)),axis=1)
```

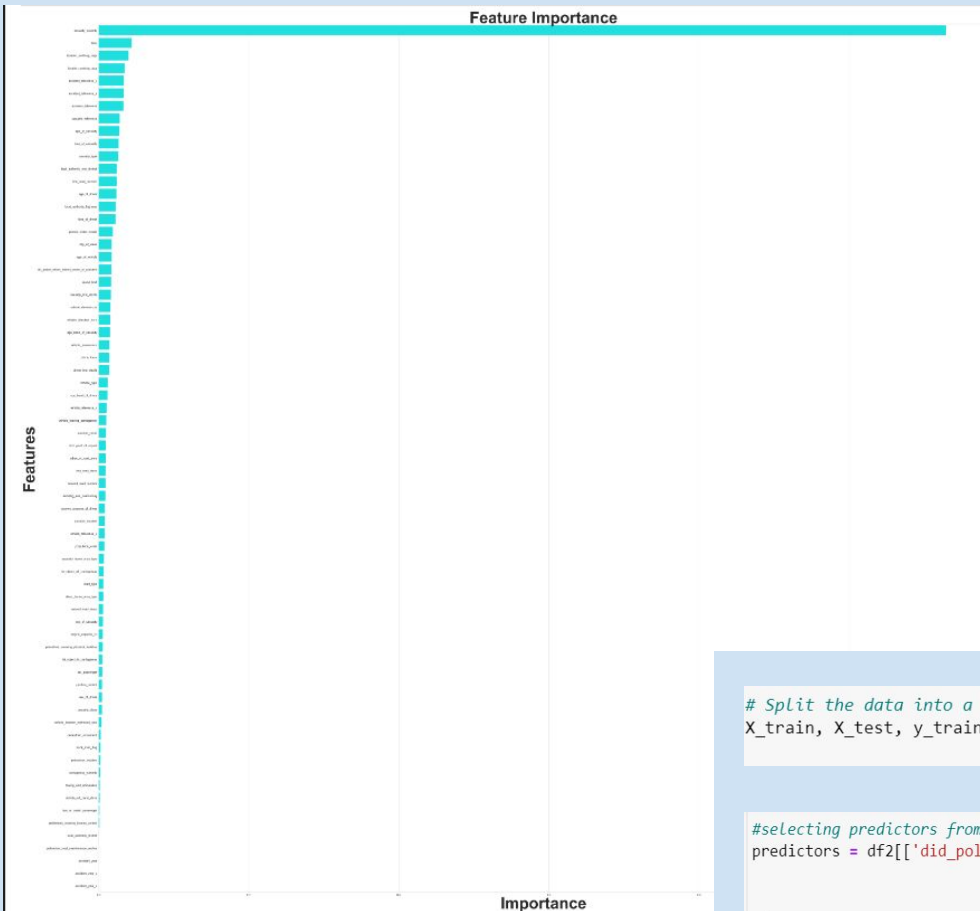
```
#Plotting the Latitude and Longitude on the map.  
HeatMap(slight_latlon).add_to(SLIGHT_UK)
```

```
# Calling the map to see the visualisation.  
SLIGHT_UK
```



# Machine learning

- Feature selection from our 69 columns using random forest but observed no correlation
- Choose specific predictors
- Train\_test\_split : 80/20 used
- Experimented with 4 algorithms : Random forest, Decision Tree, Logistic Regression and K-NN
- Evaluation metrics was Accuracy & confusion matrix performance metrics
- Result



*# Split the data into a training and test set.*

```
X_train, X_test, y_train, y_test = train_test_split(predictors.values,
                                                    df2['accident_severity'].values, test_size=0.20, random_state=99)
```

*#selecting predictors from the above features*

```
predictors = df2[['did_police_officer_attend_scene_of_accident', 'age_of_driver', 'vehicle_type', 'age_of_vehicle',
                  'engine_capacity_cc', 'day_of_week', 'road_type', 'casualty_severity',
                  'sex_of_driver', 'speed_limit', 'pedestrian_location', 'junction_control', 'urban_or_rural_area']]
```

# Machine learning

- With a 98% accuracy rate, the balanced dataset of the Random Forest model is the best classifier for predicting accidents.

## Unbalanced dataset

```
2    120493
1     31981
0      2944
Name: accident_severity, dtype: int64
```

## Balanced dataset

```
2    120493
1    120493
0    120493
dtype: int64
```

## UNBALANCED DATASET

MODELS	Accuracy	Precision	Recall	F1-score
Random Forest	96.4	96.4	96.4	96.4
Decision Tree	93.8	94.0	93.8	93.8
K-NN	94.2	94.3	94.2	93.7
Logistic Regression	84.5	83.0	85.0	83.0

## BALANCED (SMOTE) DATASET

MODELS	Accuracy	Precision	Recall	F1-score
Random Forest	98.0	98.0	98.0	98.0
Decision Tree	96.8	96.8	96.8	96.8
K-NN	80.4	80.2	80.4	80.0
Logistic Regression	93.0	93.0	93.0	93.0



# Conclusion

Our findings:

- Men are more likely to be in a car accident at all level of severities
- Age?
- Wednesday, Thursday and Friday have the most number of traffic accidents
- Summer and spring months ( May - August) have a surge in traffic accidents
- London has the highest number of accidents
- Junctions seem to be the most dangerous for traffic accidents
- Machine learning is a good tool for predicting and ultimately avoiding road traffic accident severity.

Based on these we recommended the following:

- Traffic congestion in the urban areas should be reduced
- Speed and highway cameras should be installed to monitor vehicles exceeding speed limit
- Educate people on how to drive safely.
- Implement proper traffic management services and Machine Learning in making predictions on road traffic accidents.

*Thank you!*