Data Analytics

Road accidents analysis

Adel Ait Slimane

July, 2024

Table of content

Introduction

Studying the impact of car accidents on traffic in the United States is crucial for understanding the dynamics of roadway safety, traffic flow, and urban planning. Car accidents, ranging from minor fender benders to major collisions, can significantly disrupt traffic patterns, leading to congestion, delays, and secondary accidents. Analyzing these incidents provides valuable insights into the causes and consequences of traffic disruptions, allowing for the development of more effective traffic management strategies and safety measures. By examining various factors such as accident frequency, location, time of day, and severity, we can identify trends and implement solutions aimed at reducing accidents and minimizing their impact on traffic flow. This research is not only vital for enhancing road safety but also for improving the overall efficiency and reliability of transportation networks, ultimately contributing to a safer and more efficient travel experience for all road users.

The goal of this project is to analyze trends in car accidents over time and localization and evaluate their impact on traffic.

In order to achieve the project, I will follow these steps:
- Search project topic
- Trello planning creation
- Collect data
- Exploratory data analysis in Python (data cleaning, data wrangling, data visualization)
- Creation of a database using MySQL
- Add data to database and create Entity Relationship Diagram
- Expose data by creating an API
- Do web scraping to collect information
- Train and test model with machine learning

Data collection and data sources

Flat files:
The main dataset was found on Kaggle under the name US Accidents (2016 – 2023). A lighter, easier version to work with focused on March 2023 from this dataset exists but I wanted the largest information source for accuracy and exhaustive purposes regarding analysis and machine learning. The lighter version was only used in MySQL

API:
An API was created to facilitate data access.

Web scraping:
In correlation with the API, Wikipedia website information about population was scraped to scale high occurrences of accidents.

Data cleaning and exploratory data analysis

The flat file from Kaggle was used to conduct data cleaning and exploratory data analysis.
For better understanding, below are main numbers, graphs and visualizations.

Initial number of accidents

```
display(df.describe())
```

```
(7728394, 46)
```

Data types

```
df.dtypes
```

```
ID                     object
Source                 object
Severity                int64
Start_Time             object
End_Time               object
Start_Lat             float64
Start_Lng             float64
End_Lat               float64
End_Lng               float64
Distance(mi)          float64
Description            object
Street                 object
City                   object
County                 object
State                  object
Zipcode                object
Country                object
Timezone               object
Airport_Code           object
Weather_Timestamp      object
Temperature(F)        float64
Wind_Chill(F)         float64
Humidity(%)           float64
Pressure(in)          float64
Visibility(mi)        float64
Wind_Direction         object
Wind_Speed(mph)       float64
Precipitation(in)     float64
Weather_Condition      object
Amenity                 bool
Bump                    bool
Crossing                bool
```

Checking nulls

```
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
Description                 5
Street                  10869
City                      253
County                      0
State                       0
Zipcode                  1915
Country                     0
Timezone                 7808
Airport_Code            22635
Weather_Timestamp      120228
Temperature(F)         163853
Wind_Chill(F)         1999019
Humidity(%)            174144
Pressure(in)           140679
Visibility(mi)         177098
Wind_Direction         175206
Wind_Speed(mph)        571233
Precipitation(in)     2203586
Weather_Condition      173459
Amenity                     0
Bump                        0
Crossing                    0
```

Number of accidents after null deletion

```
df = df.dropna()
print(df.shape)
```

```
(3554549, 46)
```

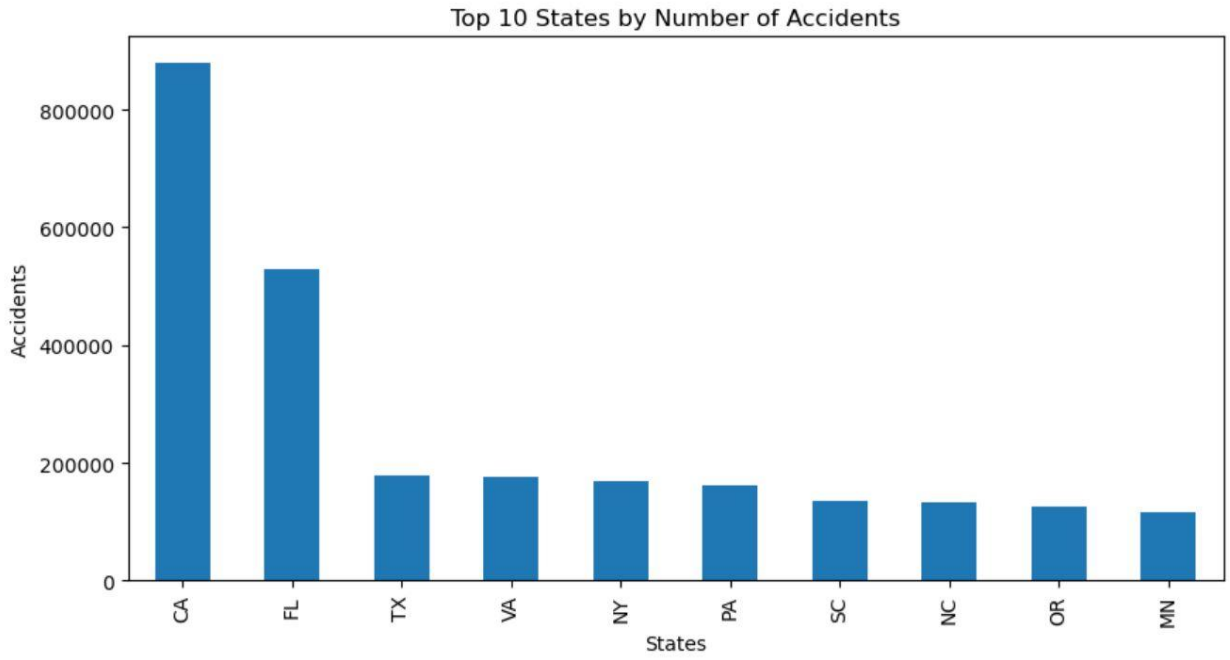
Checking for duplicates

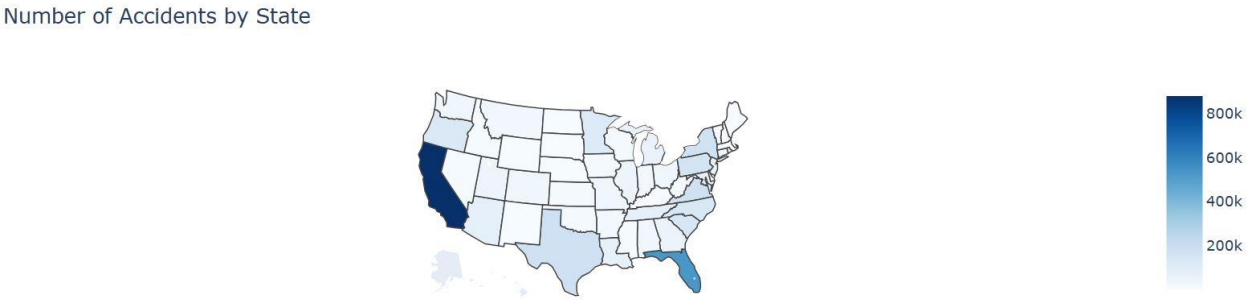
```
df.duplicated().any()
```

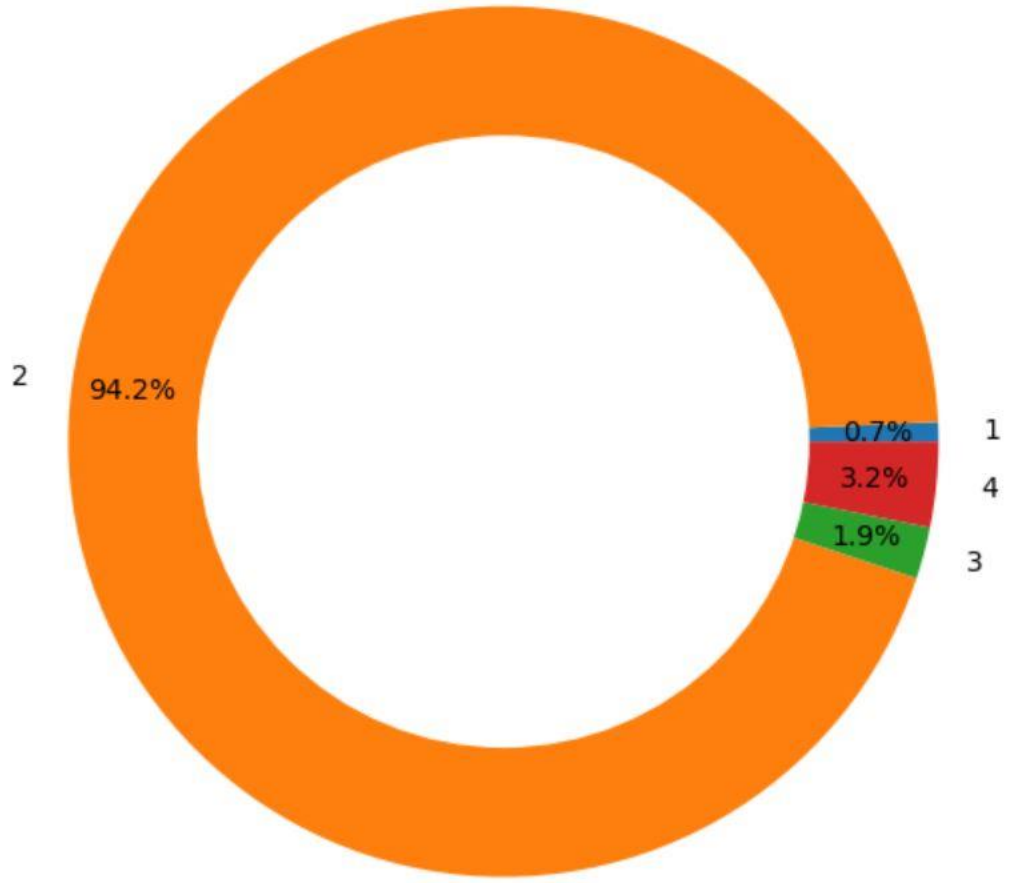
```
False
```

Visualization

Top ten accidents by state graph and map






Accidents by Severity
2
94.2%
0.7%
1
3.2%
4
1.9%
3

Database type selection

Choosing the right database management system is a critical decision for any project, as it directly impacts performance, scalability, and maintainability. My selection of MySQL as the preferred database type is driven by several key factors that align with the specific needs of the project. MySQL stands out due to its robust performance, extensive community support, flexibility, and proven reliability in handling large-scale applications.

First and foremost, MySQL is renowned for its exceptional performance, especially in read-heavy operations. Its optimized storage engines provide fast query processing and efficient storage management, making it an ideal choice for applications requiring quick and reliable access to data. Additionally, MySQL's support for indexing, partitioning, and advanced querying capabilities ensures that it can handle complex queries and large datasets with ease.

Another significant advantage of MySQL is its extensive community support and widespread adoption. Being one of the most popular open-source databases, MySQL benefits from a vast ecosystem of developers, tools, and resources. This widespread use translates into a wealth of knowledge, tutorials, and third-party integrations, facilitating easier development and troubleshooting. The active community also means regular updates and improvements, ensuring that MySQL remains secure and up-to-date with the latest features.

Flexibility is another compelling reason for choosing MySQL. It supports various storage engines, allowing users to tailor the database to their specific needs. Whether the project requires transactional support, full-text search capabilities, or memory-optimized tables, MySQL provides the necessary options to customize the database environment. Additionally, MySQL's compatibility with various platforms and programming languages like Python in this project makes it versatile and adaptable to different development environments.

Finally, MySQL's reliability and maturity as a database solution cannot be overstated. With a track record spanning over two decades, MySQL has proven its stability and robustness in numerous high-profile applications.

In summary, the decision to use MySQL is based on its superior performance, extensive community support, flexibility, and proven reliability. These attributes make MySQL an excellent choice for developing a robust, scalable, and efficient database solution that can meet the demands of modern applications.

MySQL queries and ERD

Table queries

```
182 •  select avg(`Distance(mi)`) from accident;
183
```

Result Grid | Filter Rows: | Export:

avg(`Distance(mi)`)
0.526420922417855

```
193 •  SELECT COUNT(DISTINCT Weather_Condition) FROM accidents;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

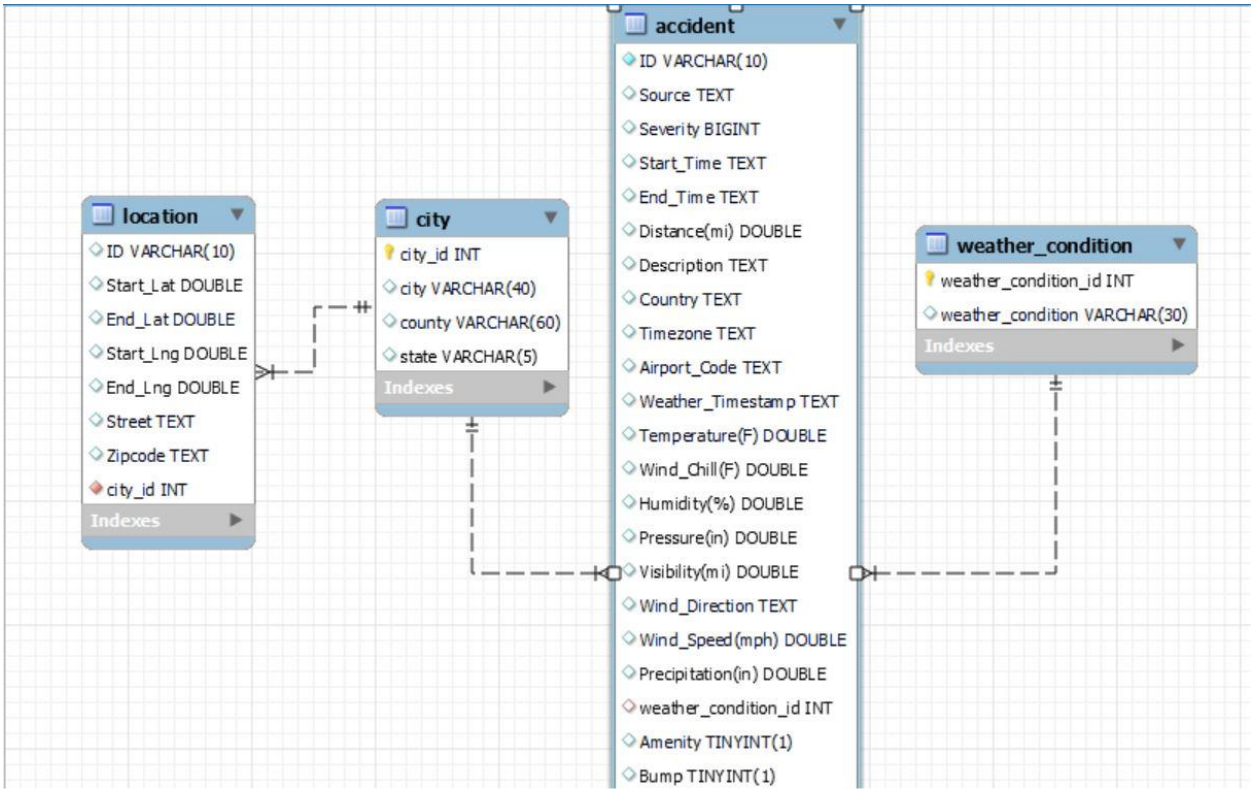
COUNT(DISTINCT Weather_Condition)
108

```
189 •  select `Precipitation(in)` from accidents
190    order by `Precipitation(in)` desc
191    limit 10;
192
```

Result Grid | Filter Rows: | Export:

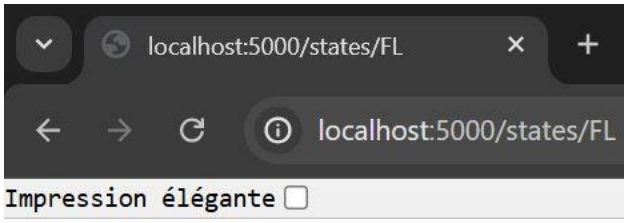
Precipitation(in)
10.13
10.09
9.99
9.99
9.99
9.99
9.99
9.99
9.99
9.99

Diagram


location
ID VARCHAR(10)
Start_Lat DOUBLE
End_Lat DOUBLE
Start_Lng DOUBLE
End_Lng DOUBLE
Street TEXT
Zipcode TEXT
city_id INT
Indexes
city
city_id INT
city VARCHAR(40)
county VARCHAR(60)
state VARCHAR(5)
Indexes
accident
ID VARCHAR(10)
Source TEXT
Severity BIGINT
Start_Time TEXT
End_Time TEXT
Distance(mi) DOUBLE
Description TEXT
Country TEXT
Timezone TEXT
Airport_Code TEXT
Weather_Timestamp TEXT
Temperature(F) DOUBLE
Wind_Chill(F) DOUBLE
Humidity(%) DOUBLE
Pressure(in) DOUBLE
Visibility(mi) DOUBLE
Wind_Direction TEXT
Wind_Speed(mph) DOUBLE
Precipitation(in) DOUBLE
weather_condition_id INT
Amenity TINYINT(1)
Bump TINYINT(1)
weather_condition
weather_condition_id INT
weather_condition VARCHAR(30)
Indexes

API

Top 20 cities




```
{
  "cities": [
    {
      "accidents": 24282,
      "city": "Miami"
    },
    {
      "accidents": 7254,
      "city": "Greenville"
    },
    {
      "accidents": 6985,
      "city": "Orlando"
    },
    {
      "accidents": 5378,
      "city": "Jacksonville"
    },
    {
      "accidents": 2925,
      "city": "Kissimmee"
    },
    {
      "accidents": 2380,
      "city": "San Antonio"
    },
    {
      "accidents": 2126,
      "city": "Sarasota"
    },
    {
      "accidents": 1993,
      "city": "Tampa"
```

Web Scraping

I scraped Wikipedia in order to get the full state name instead of just having the two letters code and get population and area size to make further exploration possible (ie accident per capita)

```
soup = BeautifulSoup(response.content, 'html.parser')
main_section = soup.find('div',class_="mw-content-ltr mw-parser-output")
tables = main_section.find_all('table')
states_table = tables[1]

state = []
state_code = []
population = []
area = []
wiki_link = []

for row in states_table.find_all('tr')[2:]:
    state.append(row.find('a').text.strip('\n'))
    wiki_link.append('https://en.wikipedia.org'+row.find('a').get('href'))
    table_data = row.find_all('td')
    state_code.append(table_data[0].text)
    population.append(int(table_data[4].text.strip('\n').replace(',','')))
    area.append(int(table_data[-2].text.replace(",","")))

state_df = pd.DataFrame({'state_name':state, 'State':state_code,'wiki_link':wiki_link,'population':population,'area_km2':area})
```

```
state_df
```

	state_name	State	wiki_link	population	area_km2
0	Alabama	AL	https://en.wikipedia.org/wiki/Alabama	5024279	135767
1	Alaska	AK	https://en.wikipedia.org/wiki/Alaska	733391	1723337
2	Arizona	AZ	https://en.wikipedia.org/wiki/Arizona	113990	295234
3	Arkansas	AR	https://en.wikipedia.org/wiki/Arkansas	53179	137732
4	California	CA	https://en.wikipedia.org/wiki/California	39538223	423967

```
result = pd.merge(df, state_df, on='State')
result
```

Signal	Turning_Loop	Sunrise_Sunset	Civil_Twilight	Nautical_Twilight	Astronomical_Twilight	state_name	wiki_link	population	area_km2
False	False	Night	Night	Night	Night	Ohio	https://en.wikipedia.org/wiki/Ohio	44826	116098
False	False	Night	Night	Night	Day	Ohio	https://en.wikipedia.org/wiki/Ohio	44826	116098

Machine Learning

The necessity of using machine learning (ML) has become increasingly apparent in today's data-driven world. Machine learning, a subset of artificial intelligence, enables systems to learn from data, identify patterns, and make decisions with minimal human intervention. This capability is essential for handling the vast amounts of data generated daily, turning it into actionable insights and driving smarter decision-making.

Machine learning is crucial for various applications across industries. In healthcare, it enables early disease detection and personalized treatment plans. In finance, it improves fraud detection and risk management. Retail businesses use ML for customer segmentation and inventory optimization, while manufacturers leverage it for predictive maintenance and quality control. Here it will be used to predict road accidents impact on traffic. Additionally, machine learning enhances user experiences through personalized recommendations, automated customer support, and efficient data processing.

The necessity of machine learning stems from its ability to automate complex tasks, improve accuracy and efficiency, and uncover insights that were previously inaccessible. As the volume and complexity of data continue to grow, the role of machine learning in harnessing this data to drive innovation and efficiency becomes indispensable.

Final presentation shows different steps an techniques to select the best parameter.

GDPR

This part precises data collection practices and the strict adherence to the General Data Protection Regulation (GDPR). I was fully committed to protecting the privacy and personal data of individuals. I confirm that no personal information was used during our data collection processes for this project.

The data collection was meticulously designed and implemented to ensure that all data gathered was strictly non-personal and anonymized. The following points highlight my compliance measures:

Only non-personal, anonymized, and aggregate data were collected.
No identifiers such as names, addresses, phone numbers, email addresses, social security numbers, or any other information that could be used to identify individuals were collected.

Data was gathered through automated processes that ensured the exclusion of any personal information.
Technical safeguards were employed to strip any incidental personal data that might have been inadvertently collected during the data-gathering phase.

All data was processed in a manner that maintained anonymity and prevented re-identification.
Data storage systems were designed to handle only anonymized datasets, with rigorous access controls to ensure data integrity and confidentiality.

I take the privacy and protection of data very seriously. I ensure that all my data collection and processing activities comply with GDPR and that personal information is never used unless explicitly permitted and safeguarded. The measures outlined above demonstrate my commitment to privacy and data protection.