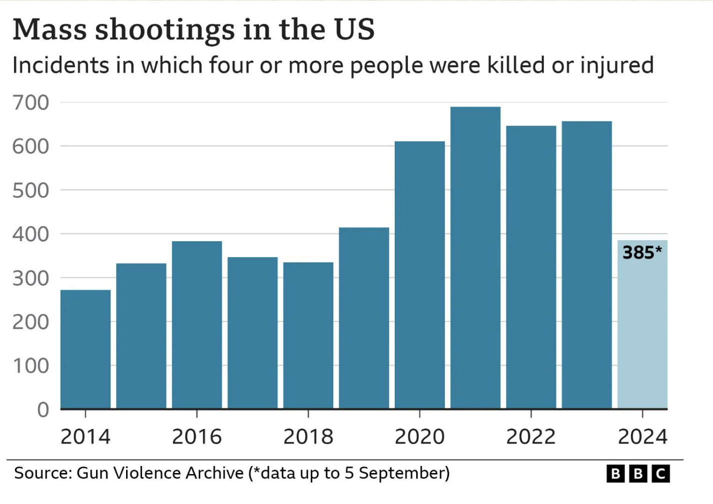
Gun Violence Prediction

By: Sarthak Sethi

WHY I CHOOSE THIS TOPIC

- Gun violence is a huge problem within the US
  - Over 385 mass shootings in the US alone in 2024 (up to sep 20)
  - More than a single mass shooting everyday

$$\frac{\text{385 mass shootings}}{\text{262 days (jan 1 to sep 20)}} = \text{1.4-ish mass shootings a day!}$$

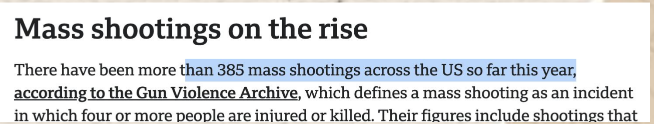


Mass shootings in the US

Incidents in which four or more people were killed or injured

Source: Gun Violence Archive (*data up to 5 September)

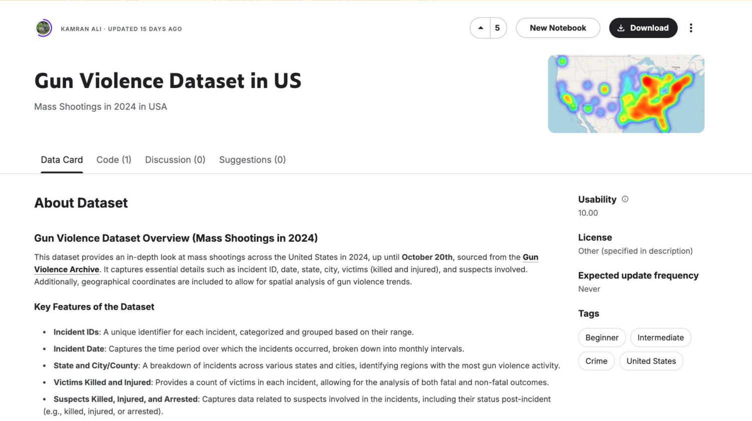
BBC



Mass shootings on the rise

There have been more than 385 mass shootings across the US so far this year, **according to the Gun Violence Archive**, which defines a mass shooting as an incident in which four or more people are injured or killed. Their figures include shootings that

THE DATA SET



Gun Violence Dataset in US

Mass Shootings in 2024 in USA

About Dataset

Gun Violence Dataset Overview (Mass Shootings in 2024)

This dataset provides an in-depth look at mass shootings across the United States in 2024, up until **October 20th**, sourced from the **Gun Violence Archive**. It captures essential details such as incident ID, date, state, city, victims (killed and injured), and suspects involved. Additionally, geographical coordinates are included to allow for spatial analysis of gun violence trends.

Key Features of the Dataset

- **Incident IDs**: A unique identifier for each incident, categorized and grouped based on their range.
- **Incident Date**: Captures the time period over which the incidents occurred, broken down into monthly intervals.
- **State and City/County**: A breakdown of incidents across various states and cities, identifying regions with the most gun violence activity.
- **Victims Killed and Injured**: Provides a count of victims in each incident, allowing for the analysis of both fatal and non-fatal outcomes.
- **Suspects Killed, Injured, and Arrested**: Captures data related to suspects involved in the incidents, including their status post-incident (e.g., killed, injured, or arrested).

Usability
10.00

License
Other (specified in description)

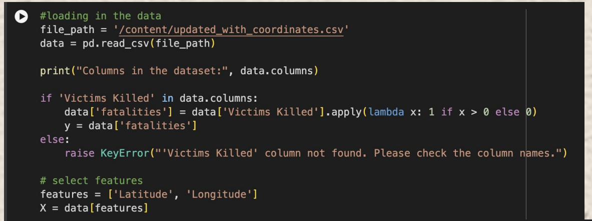
Expected update frequency
Never

Tags
Beginner Intermediate Crime United States

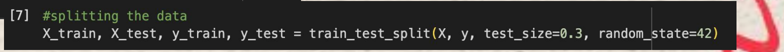
- Got this data set from Kaggle
- Covers 2024 U.S. mass shootings (up to Oct 20).
- Includes incident IDs, dates, locations, victims, and suspects.
- Geographical data supports spatial trend analysis.

HOW I ANALYZED IT

- Imported data and verified structure
- Cleaned data, addressed missing value
- Selected relevant columns
- Divided data into training and test sets.

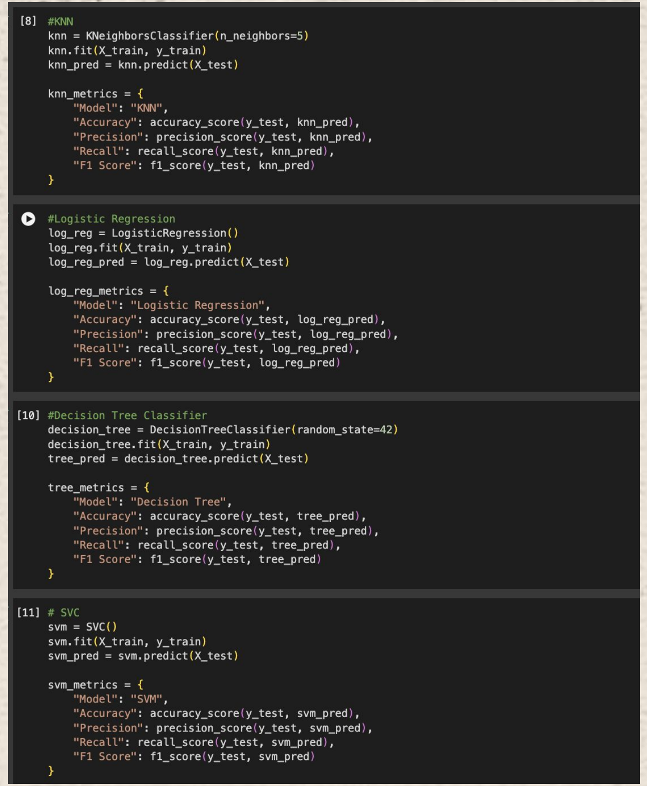


```python
#loading in the data
file_path = '/content/updated_with_coordinates.csv'
data = pd.read_csv(file_path)

print("Columns in the dataset:", data.columns)

if 'Victims Killed' in data.columns:
    data['fatalities'] = data['Victims Killed'].apply(lambda x: 1 if x > 0 else 0)
    y = data['fatalities']
else:
    raise KeyError("'Victims Killed' column not found. Please check the column names.")

# select features
features = ['Latitude', 'Longitude']
X = data[features]
```



```python
[7] #splitting the data
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

MODELS USED AND WHY?



```
[8]  #KNN
     knn = KNeighborsClassifier(n_neighbors=5)
     knn.fit(X_train, y_train)
     knn_pred = knn.predict(X_test)

     knn_metrics = {
         "Model": "KNN",
         "Accuracy": accuracy_score(y_test, knn_pred),
         "Precision": precision_score(y_test, knn_pred),
         "Recall": recall_score(y_test, knn_pred),
         "F1 Score": f1_score(y_test, knn_pred)
     }
```

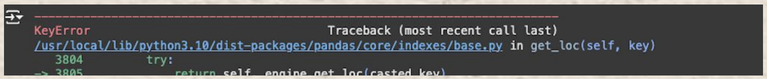
```
     #Logistic Regression
     log_reg = LogisticRegression()
     log_reg.fit(X_train, y_train)
     log_reg_pred = log_reg.predict(X_test)

     log_reg_metrics = {
         "Model": "Logistic Regression",
         "Accuracy": accuracy_score(y_test, log_reg_pred),
         "Precision": precision_score(y_test, log_reg_pred),
         "Recall": recall_score(y_test, log_reg_pred),
         "F1 Score": f1_score(y_test, log_reg_pred)
     }
```

```
[10] #Decision Tree Classifier
     decision_tree = DecisionTreeClassifier(random_state=42)
     decision_tree.fit(X_train, y_train)
     tree_pred = decision_tree.predict(X_test)

     tree_metrics = {
         "Model": "Decision Tree",
         "Accuracy": accuracy_score(y_test, tree_pred),
         "Precision": precision_score(y_test, tree_pred),
         "Recall": recall_score(y_test, tree_pred),
         "F1 Score": f1_score(y_test, tree_pred)
     }
```

```
[11] # SVC
     svm = SVC()
     svm.fit(X_train, y_train)
     svm_pred = svm.predict(X_test)

     svm_metrics = {
         "Model": "SVM",
         "Accuracy": accuracy_score(y_test, svm_pred),
         "Precision": precision_score(y_test, svm_pred),
         "Recall": recall_score(y_test, svm_pred),
         "F1 Score": f1_score(y_test, svm_pred)
     }
```

- K-Nearest Neighbors (KNN)
- Logistic Regression
- Decision Tree
- Support Vector Machine (SVM)

CHALLENGES FACED

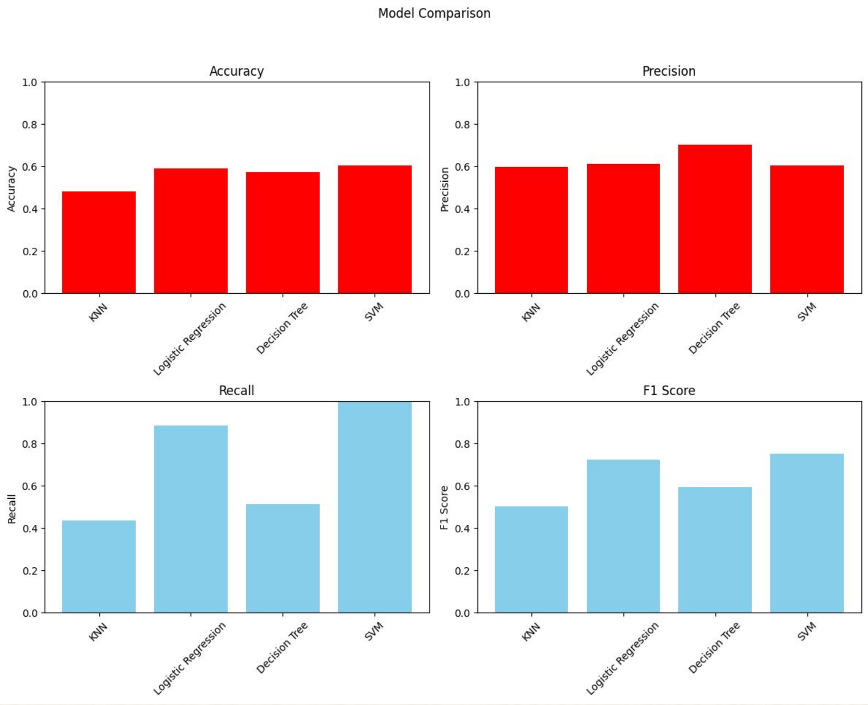


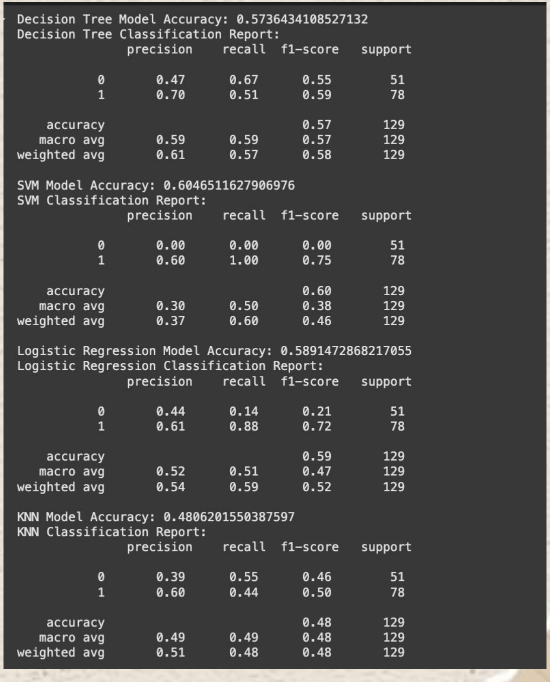
```
--------------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
/usr/local/lib/python3.10/dist-packages/pandas/core/indexes/base.py in get_loc(self, key)
   3804         try:
-> 3805             return self._engine.get_loc(casted_key)
```

- Data quality issues required preprocessing

- Precision limitations in models due to imbalanced data

THE RESULTS





```
Decision Tree Model Accuracy: 0.5736434108527132
Decision Tree Classification Report:
              precision    recall  f1-score   support

           0       0.47      0.67      0.55        51
           1       0.70      0.51      0.59        78

    accuracy                           0.57       129
   macro avg       0.59      0.59      0.57       129
weighted avg       0.61      0.57      0.58       129

SVM Model Accuracy: 0.6046511627906976
SVM Classification Report:
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        51
           1       0.60      1.00      0.75        78

    accuracy                           0.60       129
   macro avg       0.30      0.50      0.38       129
weighted avg       0.37      0.60      0.46       129

Logistic Regression Model Accuracy: 0.5891472868217055
Logistic Regression Classification Report:
              precision    recall  f1-score   support

           0       0.44      0.14      0.21        51
           1       0.61      0.88      0.72        78

    accuracy                           0.59       129
   macro avg       0.52      0.51      0.47       129
weighted avg       0.54      0.59      0.52       129

KNN Model Accuracy: 0.4806201550387597
KNN Classification Report:
              precision    recall  f1-score   support

           0       0.39      0.55      0.46        51
           1       0.60      0.44      0.50        78

    accuracy                           0.48       129
   macro avg       0.49      0.49      0.48       129
weighted avg       0.51      0.48      0.48       129
```

NEXT STEPS

STEP 1

Explore additional machine learning models

STEP 2

Enhance data preprocessing techniques

STEP 3

Experiment with ensemble methods for improved accuracy

STEP 4

Expand features and test model stability on larger/more datasets

THANK YOU FOR LISTENING