



# Mass Shooting Prediction

By: Sarthak Sethi  
CS171 - Prof. Sengupta

XXXX







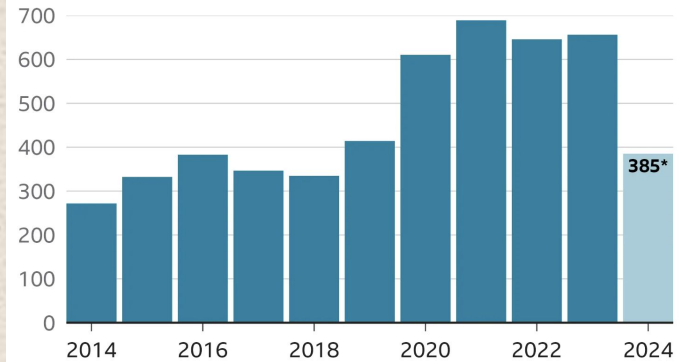
# WHY I CHOOSE THIS TOPIC

- **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**

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

## 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

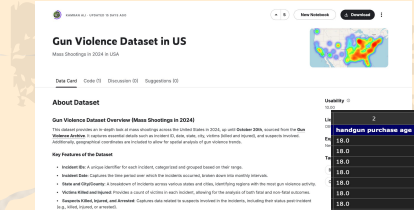
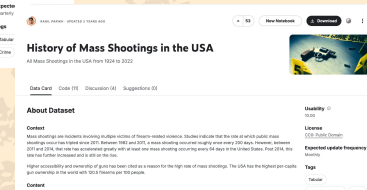
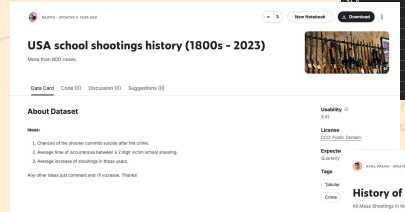
# THE SOLUTION



- Use preexisting Data on mass shootings within the United States
  - See if there any similarities
  - Anything that could help us predict, or know what might cause a shooting
    - Even better if we could know exactly what type of gun the perpetrator might have based off data
    - And where it might occur to have extra police stationed there to more safety to public



## THE DATA

[illegible]

- Multiple different Kaggle Data Set
  - Verified multiple incidents manually
- LLM Model - Perplexity
  - Population
  - Crime Rate
  - Gun Laws
    - How old to buy
      - Handgun
      - Long gun
    - Open carry or not
- ChatGPT

# HOW I ANALYZED IT

- Data Integration and Cleaning
- Model Training and Evaluation
- Unseen Data Testing
- Insights and Visualizations



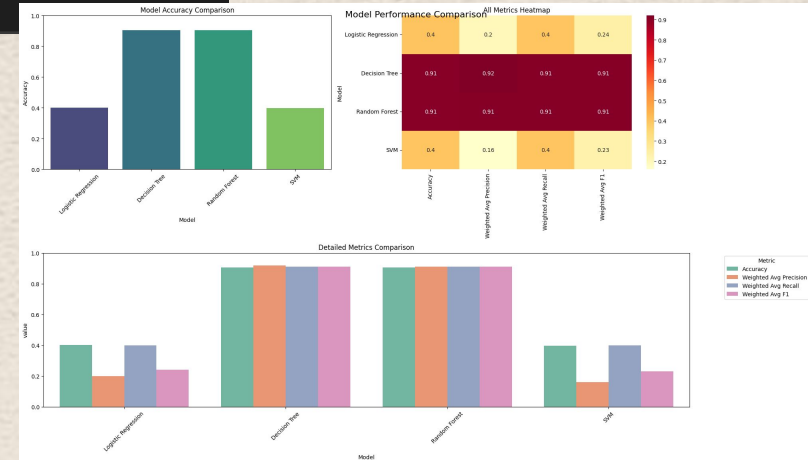
# MODELS USED

- Decision Tree
- Random Forest
- SVM
- Logistic Regression
- KNN - Dropped since its a large dataset!

```
#test the data

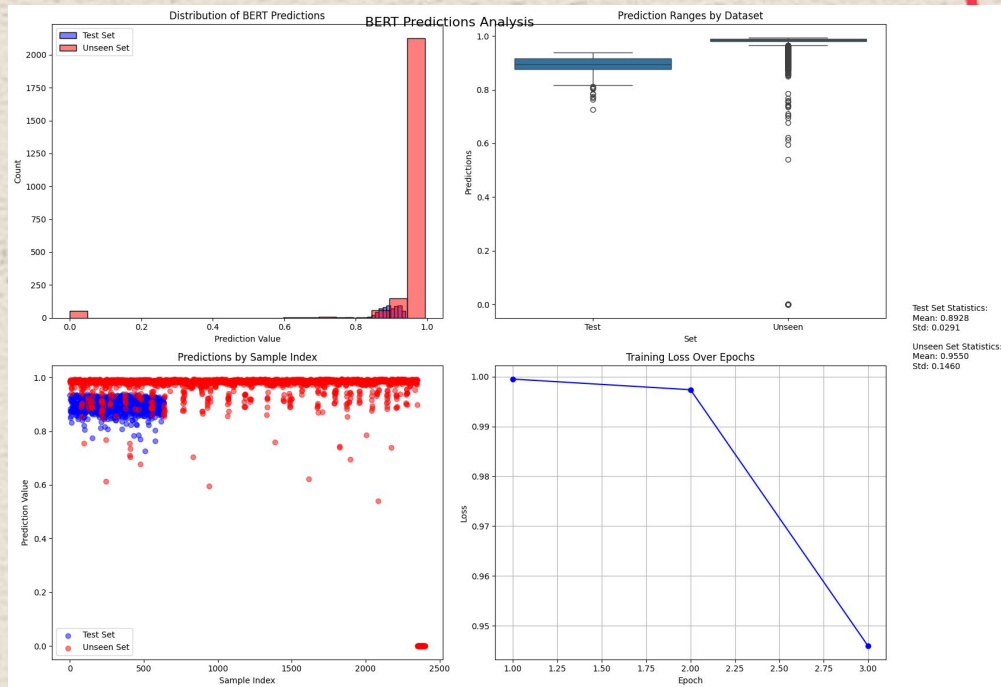
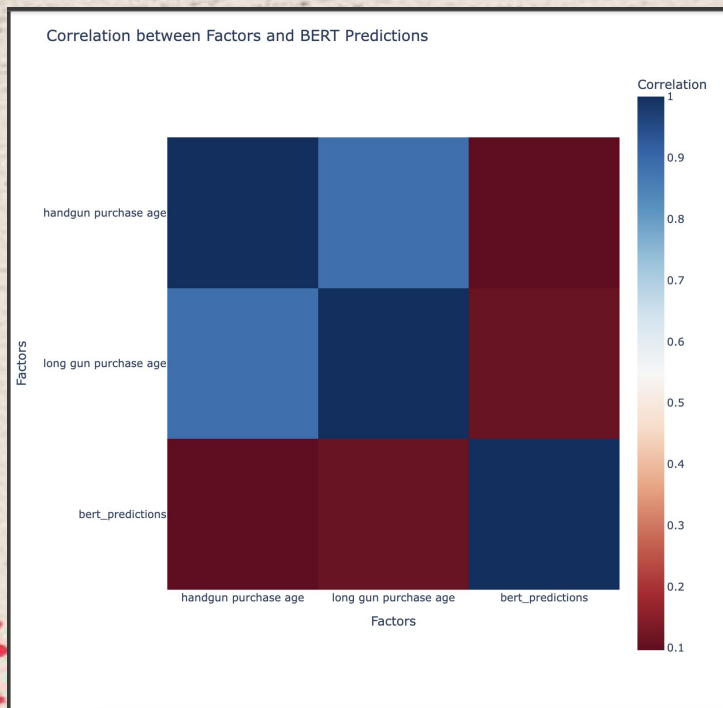
# Initialize models
models = {
    "Logistic Regression": LogisticRegression(),
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(),
    "SVM": SVC()
}

# Train and evaluate each model
results = {}
for name, model in models.items():
    # Train the model
    model.fit(X_train, y_train)
    # Predict on the test set
    y_pred = model.predict(X_test)
    # Evaluate the model
    acc = accuracy_score(y_test, y_pred)
    print(f"\n{name} Results:")
    print(f"Accuracy: {acc:.4f}")
    print(classification_report(y_test, y_pred))
    results[name] = acc
```





# BERT - MODEL



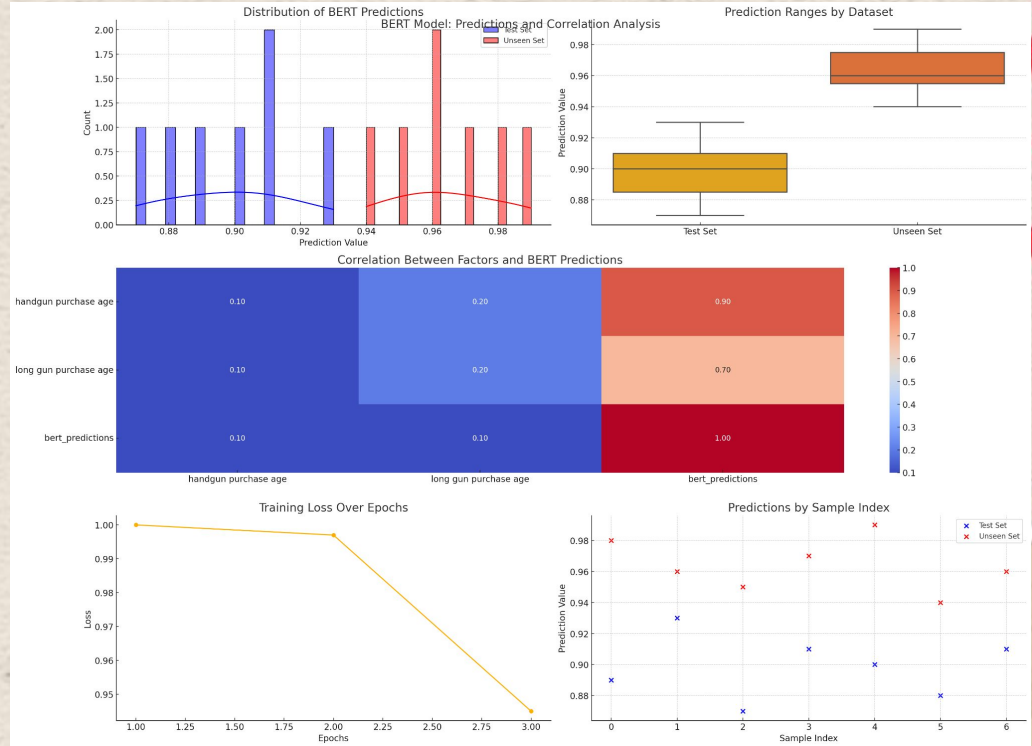
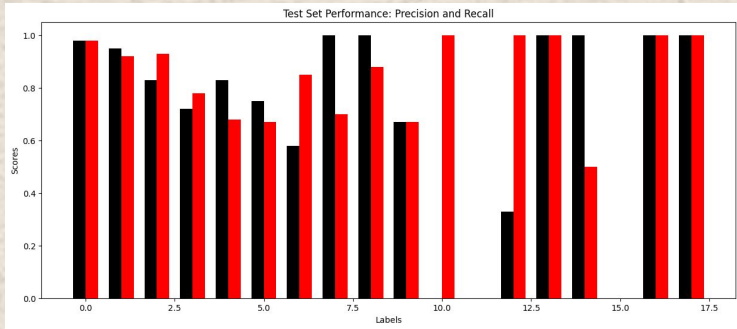
# CHALLENGES FACED



- GPU limitations, for BERT (had to buying colab points)
- Data Cleaning, and problems with working with .csv files
- Heatmap, not working even after help with LLMs on my code
- Had to scrape ideas of a front end and google maps API, due to time and not buying points sooner
- Tried using other heavy models like BERT, but didn't work



# CURRENT RESULTS





**THANK YOU FOR  
LISTENING**

