NFL Fan Arrests Data Project

[nfl-arrests.csv]
[John Draa]

What is the dataset?

- Source of the dataset is from Kaggle: https://www.kaggle.com/datasets/washingtonpost/nfl-arrests
- The dataset is a csv file that contains 1,006 data points
- Data contains fan arrest numbers at NFL games from 2011-2015
- Each row represents a single game with the following key variables:
    - **week_num:** What week of the season game was played in
    - **home_team/away_team**: Whether the team was at home stadium or not
    - **home_score/away_score:** Scores for the two teams
    - **OT_flag:** Whether the game went into overtime or not
    - **arrests**: Number of fan arrests

2011	1	Sunday	1:15:00 PM	Arizona	Carolina
2011	4	Sunday	1:05:00 PM	Arizona	New York
2011	7	Sunday	1:05:00 PM	Arizona	Pittsbu
2011	9	Sunday	2:15:00 PM	Arizona	St. Lou
2011	13	Sunday	2:15:00 PM	Arizona	Dallas
2011	14	Sunday	2:05:00 PM	Arizona	San Fra
2011	15	Sunday	2:15:00 PM	Arizona	Clevela
2011	17	Sunday	2:15:00 PM	Arizona	Seattle
2012	1	Sunday	1:25:00 PM	Arizona	Seattle
2012	3	Sunday	1:05:00 PM	Arizona	Philadel
2012	4	Sunday	1:05:00 PM	Arizona	Miami
2012	6	Sunday	1:05:00 PM	Arizona	Buffalo
2012	8	Monday	5:30:00 PM	Arizona	San Fra
2012	12	Sunday	2:25:00 PM	Arizona	St. Lou
2012	15	Sunday	2:05:00 PM	Arizona	Detroit
2012	16	Sunday	2:25:00 PM	Arizona	Chicago

What am I trying to find?

- Main Question: “Can I predict the number of fan arrests at an NFL game based on the team, stadium, and game context?”
- Classification problem, determined by arrest levels (low, medium, high)
- Patterns I expect to find:
    - Certain teams or stadiums will show consistently higher arrest rates
    - Arrest rates will also be based on the context of the game, such as a rivalry game, the final score of the game, how late in the season/close to playoffs the game is

What are the benefits?

- Predicts high arrest games in advance, so stadium security and local police can be better prepared
- Planning around high risk matchups reduces disruptions and could help create a safer environment for fans
- Insights on arrest factors would support evidence based adjustments on stadium policies (alcohol sales, entry point security)
- Enhance community relations and public perception of the league through transparent security planning
- Same framework could be applied to other sports leagues and large events

Cleaning and Preparing Data

- Removed duplicate rows
- Dropped rows with missing arrests and reset index
- Standardized text fields (day_of_week, division_game)
- Added “No OT” flag, previously there was only OT flag
- Identified and converted object datatype columns
- Checked for missing values and data types
- Identified object data type columns
- Converted object columns to numeric types
- Created new outcome columns (away_team_won, game_outcome)
- Created additional features (score_diff, rivalry_status)
- Encoded categorical variables as needed (after charts)

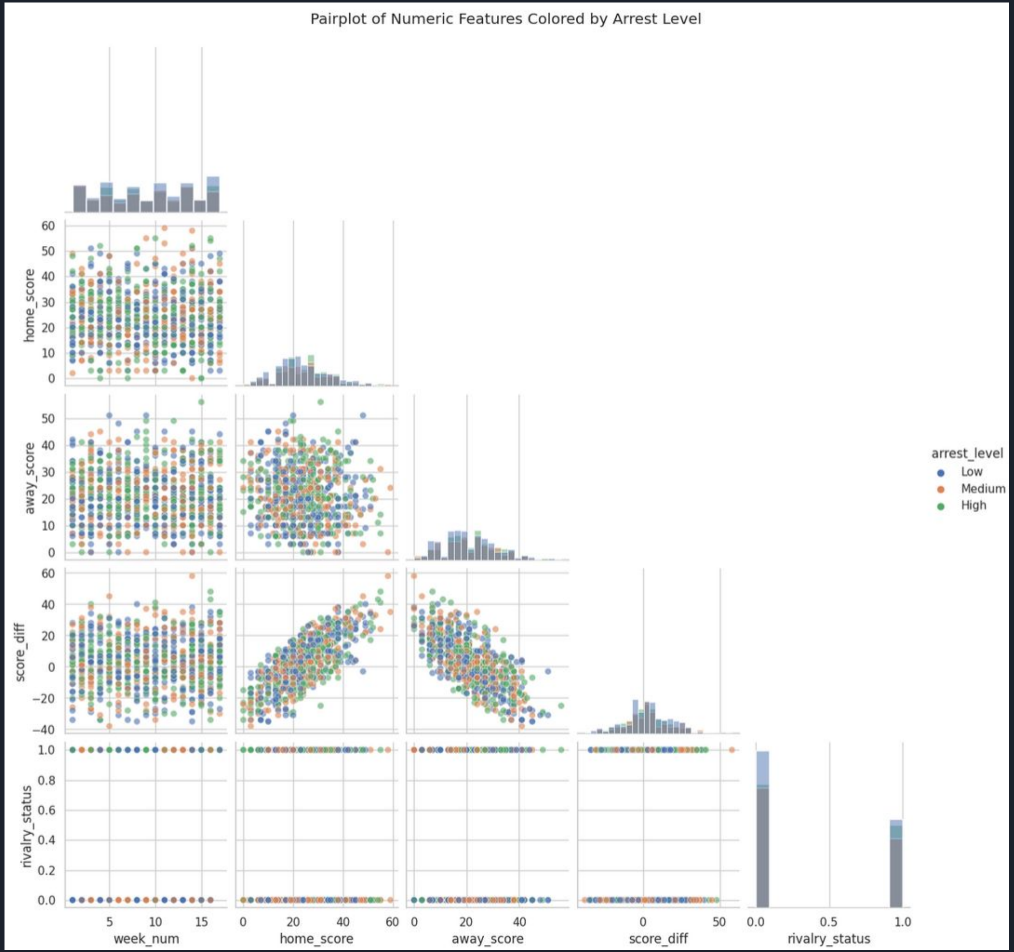
```python
df = arrests_df.copy()

#remove duplicates and missing values
df.drop_duplicates(inplace=True)
df.dropna(subset=['arrests'], inplace=True)
df.reset_index(drop=True, inplace=True)
print("Missing values per column:\n", df.isna().sum())

#standardize text fields and make new columns
df['day_of_week']   = df['day_of_week'].str.strip().str.title()
df['division_game'] = df['division_game'].str.strip().str.lower()
df['OT'] = df['OT_flag'].fillna("No").apply(lambda x: "OT" if str(x).strip().upper()=="OT" else "No OT")
df['rivalry_status'] = df['division_game'].map({'y': 1, 'n': 0})
df['score_diff'] =  df['home_score'] - df['away_score']
df['away_team_won'] = df['away_score'] > df['home_score']
df['game_outcome']  = df['away_team_won'].map({True: 'Away Win', False: 'Home Win'})

#check info()
df.info()
print(df.isna().sum())

#check for obj datatype
obj_cols = df.select_dtypes(include='object').columns.tolist()
print("Object-dtype columns:", obj_cols)

#convert obj to numeric
numeric_like_cols = [col for col in obj_cols if df[col].dropna().astype(str).str.fullmatch(r'-?\d+(\.\d+)?').all()]
for col in numeric_like_cols:
    df[col] = pd.to_numeric(df[col], errors='coerce')

#final inspection
df.info()
print("Missing values per column:\n", df.isna().sum())

#create the target for exploration
df['arrest_level'] = pd.qcut(df['arrests'], q=3, labels=['Low','Medium','High'])
```

Descriptive Analysis

- Week Number
  - Games spread evenly across weeks 1-17, no clear hot weeks for arrests
- Day of Week and Game Time
  - Most games are on sunday afternoon; also midweek and primetime games but arrest counts are similar across days and times
- Home and Away Scores
  - Average points around 24-27, teams score roughly equally on average
- Overtime
  - Small amount of games went into OT, these show slightly higher arrests
- Score Difference
  - Distribution is mostly centered around close margins of 0-7 points, fewer blowouts
- Division/Rivalry Games
  - Half of the games are within the same division, these have a higher median arrest count
- Arrests
  - Right skewed distribution with a median of 3, mean slightly higher. Most games see few arrests, but a tail of high incident games exist
- Game Outcome
  - Home wins roughly 55%, away wins 45%, home team victories correlate with marginally higher arrests

Pair Plot

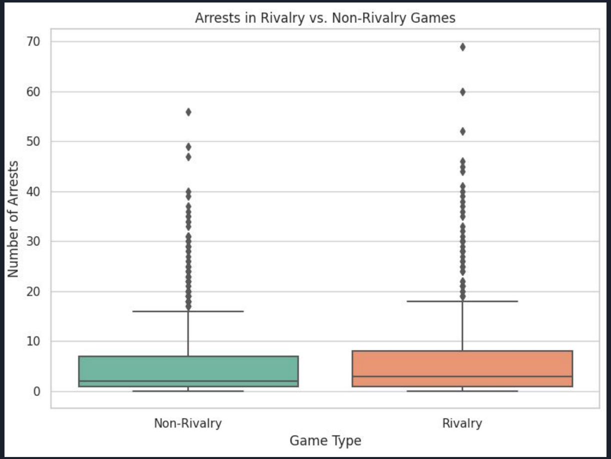
- Close games drive arrests: High arrests cluster around small score margins
- Balanced scoring matters: Both teams scoring well, but closely, leads to more incidents
- Rivalry effect: Rivalry matchups appear more often in the high arrest category
- No season time bias: Arrrest levels are evenly spread across weeks, so timing in the season plays a minor role



Arrests By Rivalry Games

Analysis & Patterns Observed:

- Rivalry games show a **higher median** number of arrests compared to non-rivalry games.
- The data also shows a **greater spread and more outliers** in rivalry games, indicating that some rivalry games escalate more than others.
- This pattern supports the idea that the **intensity of rivalry matchups** contributes to an increase in disruptive fan behavior.



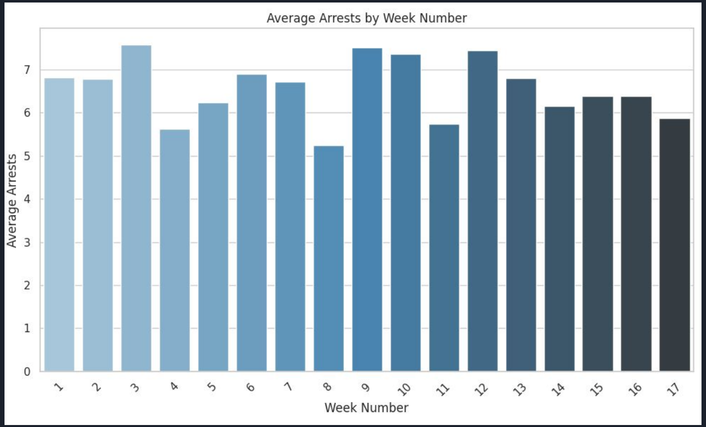
Anomalies Noted:

- A few **non-rivalry games show unusually high arrest counts**, possibly due to other high-stakes factors (playoffs, team history, or incidents not captured in the rivalry flag).
- Some rivalry games have surprisingly low arrests, suggesting **contextual factors** (time of day, stadium policies, weather) may also play a role.

Arrests By Week

Analysis & Patterns Observed:

- **No clear upward trend** across the season. Arrests **fluctuate week to week**.
- Weeks 1-3 begins **surprisingly high**, likely due to opening-week excitement and large fan turnouts.
- Several **late-season weeks (14-17)** show **lower arrest averages**, contradicting the initial hypothesis.



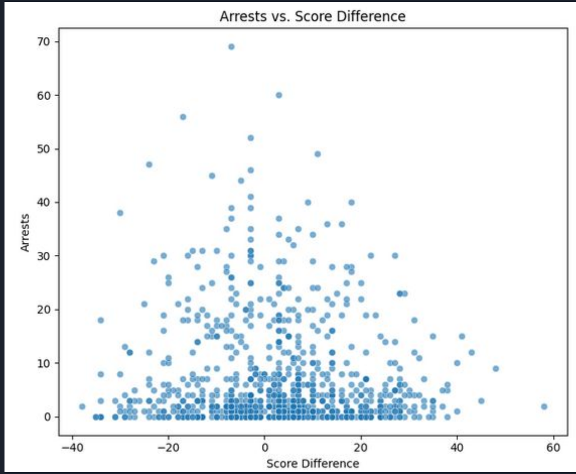
Anomalies Noted:

- Some **mid-season spikes** in arrests (Week 9) might be explained by **rivalry games**, **nationally televised matchups**, or **holidays**, but that's not all visible from this dataset alone.
- **Low Week 17 arrests** are unexpected and might reflect **rested starters**, **less crowded stadiums**, or **lower stakes for eliminated teams.**
- Arrests aren't always steadily increasing — suggesting other variables (rivalries, team performance, location) contribute more than how close to playoffs the season is.

Arrests By Score

Analysis & Patterns Observed:

- Score_diff is centered around 0, with negative values indicating an away-team win and positives values a home-team win
- The plot shows a **negative relationship**: **arrests are more common** when the score difference is near 0.
- As the margin grows, arrests decline - fan engagement drops off in blowouts, but the drop is steeper on the away side.
- Arrests peak when score_diff is small, but with slight asymmetry:
  - Close home wins (small positive) show slightly less arrests than equally close away games



Anomalies Noted:

- A few large positive diffs (home blowout) generate high arrests - rivalry games or high tension matchups
- A few large negative diffs (away blowout) also see spikes, possibly from upset-driven celebrations or dissatisfied home fans
- A few close games have low arrests, hinting at other factors like weekday games or weather

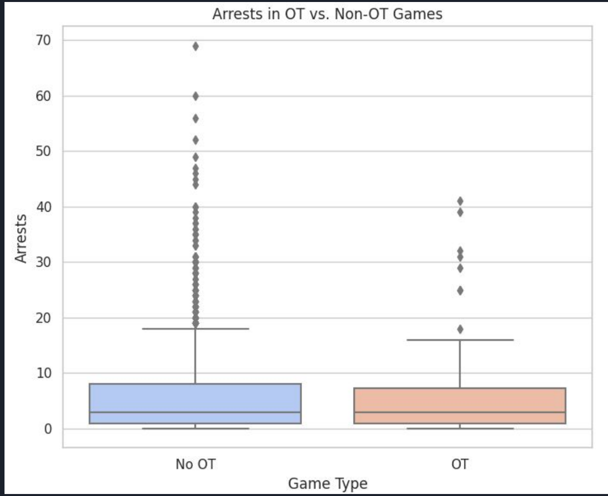
Arrests By Overtime Games

Analysis & Patterns Observed:

- **There is no significant difference in medians** between OT and non-OT games — both are centered around similar arrest counts.

- **Number of extreme outliers is higher in non-OT games**, including arrests well above 40 and 60.

- **The distributions overlap heavily**, suggesting that simply going to OT does not consistently lead to more arrests.



Anomalies Noted:

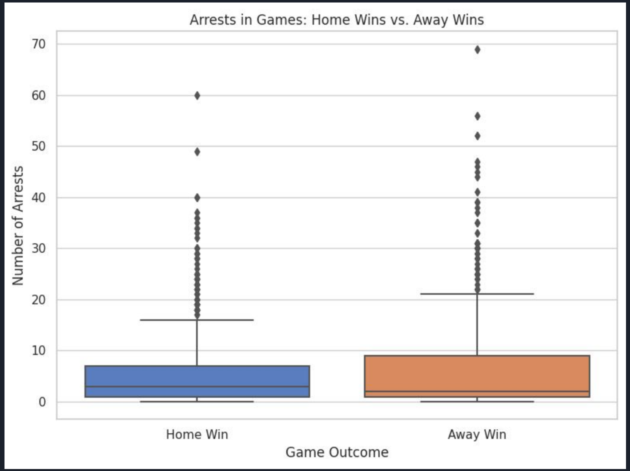
- Some **non-OT games had exceptionally high arrest counts**, again indicating other higher influencing factors such as rivalries or crowd size.

- A few OT games had arrest spikes, but they were **less extreme** than non-OT outliers.

Arrests By Away vs Home Win

Analysis & Patterns Observed:

- Arrests are generally **higher when the away team wins**:
  - The **median arrest count** is visibly higher for away wins.
  - The **interquartile range** is also larger for away wins, showing more variability.
  - There are **more extreme arrest outliers** in away wins — including the highest observed arrest counts in the dataset.
- This supports the idea that **home team losses** create a **more volatile atmosphere**, especially in competitive or emotional games.



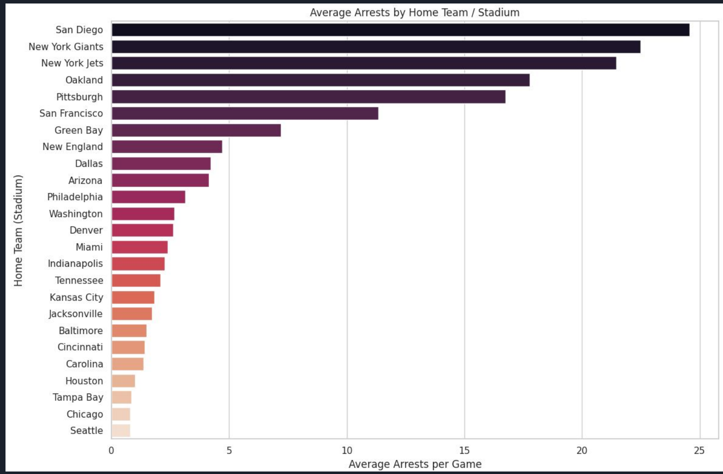
Anomalies Noted:

- Some home wins still had high arrest counts, likely due to other risk factors such as **rivalry games**, **close scores**, or **high attendance**.
- A few away wins had low arrests — suggesting the **matchup intensity or stakes** also matter.

Average Arrests By Home Team/Stadium

Analysis & Patterns Observed:

- **San Diego, the New York Giants, and the New York Jets** (Jets and Giants play at the same stadium) top the list with the highest average arrests per game.
- Other high-arrest stadiums include **Oakland, Pittsburgh, and San Francisco**, suggesting regional or city-linked patterns.
- **Many of the lowest-arrest stadiums** (e.g., Seattle, Chicago, Tampa Bay) consistently fall under 1 arrest per game on average.



Anomalies Noted:

- Some teams like **San Diego and both New York teams** have unusually high averages potentially due to **team performance**, **reporting consistency**, **urban setting**, or **larger stadium capacity**.
- Teams in historically quieter stadiums (e.g., Jacksonville, Carolina) have much lower averages, likely due to playing fewer high-tension games and possible underreporting.

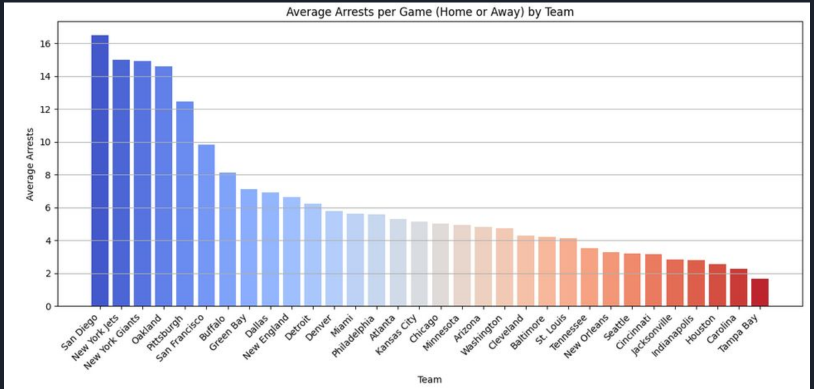
Average Arrests Per Team (Home + Away)

Analysis and Patterns Observed:

- Averages arrests across all games a team appears in, home or away
- San Diego and the NY teams remain at the top, consistent with home team chart
- Focuses more on fanbase behavior, removing local stadium effects
- Some teams dropped in ranking, showing stadium or city factors may boost arrests
- Others rose, suggest fan behavior can travel even when the team plays away

Anomalies Noted:

- Some teams showed low home arrests but higher overall — pointing to intense away games or traveling fan issues.
- A few teams stayed low in both charts, suggesting consistently calmer games regardless of location.



Hypothesis

- More arrests happen during rivalry games as they are more emotionally charged
- Close scored games have more arrests, again because of increased tensions and emotions
- Away team wins will cause more arrests, as there are generally more home team fans at a game
- **Overtime games were expected to increase arrests**, but the data showed little difference — other factors likely matter more.
- Specific stadiums and teams will consistently have higher average arrests, due to factors like local law enforcement policies and regional fan behavior

Outliers

- Several outliers in arrest counts (40+)
- These outliers were not always tied to rivalry or OT games, suggesting unseen factors
- Anomalies include unexpected spikes in mid season weeks and non rivalry matchups
- Possible cause: holidays, national broadcasts, crowd size, unrecorded incidents
- Outliers were kept to maintain real world complexity for model training

Problems with Dataset

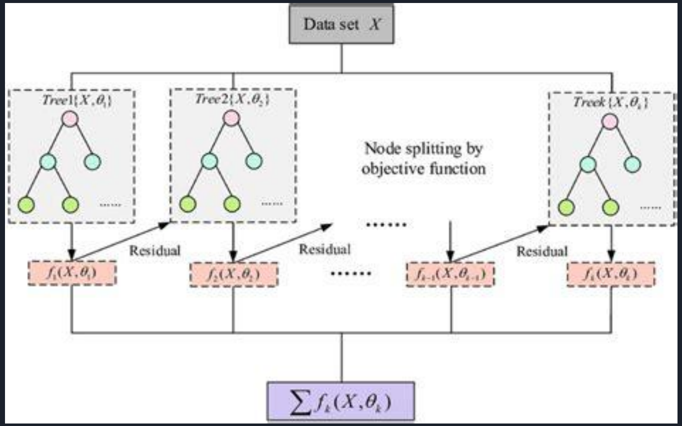
- Incomplete Coverage: Not all jurisdictions provided full data. Cleveland and New Orleans submitted no data, while others like BUF, MIA, OAK, and STL only sent partial or summarized figures.
- Inconsistent Reporting: Each police department uses different formats and reporting criteria, making direct comparisons across stadiums hard to standardize.
- Missing Contextual Data; Key arrest details like cause, location (inside stadium vs parking lot) or demographics are often missing. For example, Detroit, Minneapolis, and Atlanta excluded parking lot arrests.
- Non-uniform Time: St. Louis provided only yearly totals instead of game by game data, limiting time based analysis.
- Potential Bias in Arrest Counts: Arrest rates may reflect differences in policing intensity or policies across jurisdictions, not just fan behavior.

Potential Improvements

- Standardize reporting formats across jurisdictions
- Ensure full participation from all stadiums and police departments
- Include more detailed arrest info (cause, location, demographics)
- Ensure consistent time intervals, ideally game by game
- Ensure arrests are tracked for both inside and outside stadium
- Add other contextual data like crowd size, weather, and alcohol sales for better analysis

Why did I choose XGBoost?

- Categorical target with 3 classes (Low, Medium, High)
- Data showed non linear patterns (rivalry, score_diff, game_outcome)
- Needed model that handled complex feature interactions
- XGBoost performed best overall, especially on difficult Class 2 (high arrests)



XGBoost Setup and Feature Engineering

- Target Variable:
  - arrest_level (low, medium high), encoded with LabelEncoder for model input
- Feature Engineering:
  - Score_diff = home score - away score
  - Rivalry_status (1 if division game, 0 otherwise) - shows game intensity
- Encoding:
  - One-hot encoding applied to home_team, away_team, day_of_week, division_game, OT
- Train/Test Split:
  - 80% for training, 20% for testing
  - Split made sure all 3 arrest levels were evenly represented in both sets
  - random_state=42 for reproducibility
- Hyperparameter Tuning:
  - Used GridSearchCV to tune XGBoost
  - Tested combinations of depth, learning rate, estimators and more
  - Picked setup with best accuracy according to my GridSearch:
    - 'colsample_bytree': 1.0,
    - 'learning_rate': 0.1,
    - 'max_depth': 3,
    - 'n_estimators': 100,
    - 'subsample': 0.8

```
#hyperparameter tuning + grid search
param_grid = {'n_estimators': [100, 200, 300],'max_depth':[3, 5, 7],'learning_rate':[0.01, 0.1],'subsample':[0.8, 1.0],'colsample_bytree': [0.8, 1.0]}

grid = GridSearchCV(estimator=XGBClassifier(objective='multi:softprob',use_label_encoder=False,eval_metric='mlogloss',random_state=42),param_grid=param_grid,cv=cv,scoring='accuracy',n_jobs=-1,verbose=1)
grid.fit(X_train, y_train)

best_xgb = grid.best_estimator_
print("Best XGBoost params:", grid.best_params_)
```

```
Fitting 5 folds for each of 72 candidates, totalling 360 fits
Best XGBoost params: {'colsample_bytree': 1.0, 'learning_rate': 0.1, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.8}
```

XGBoost Evaluation

- Tuned XGBoost performed best overall
  - Accuracy: 0.665 (highest)
  - F1 score for class 2 (high arrests): 0.44, best at handling the hardest class
- Random Forest came close in accuracy (0.655)
  - Lower F1 on class 2 (0.38)
- Logistic Regression was simpler and consistent
  - Accuracy: 0.644
  - Decent on class 2 (0.41)
- Stock XGBoost underperformed
  - Accuracy was < 0.60
  - Struggled on Class 2
- Ensemble model showed slightly improved stability
  - Balanced predictions across all classes, accuracy comparable to top models, but no major gain over tuned XGBoost

```
#tuned XGBoost test
y_t_pred  = best_xgb.predict(X_test)
y_t_proba = best_xgb.predict_proba(X_test)

print("=== Tuned XGBoost ===")
print("Accuracy:  ", accuracy_score(y_test, y_t_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_t_pred))
print("Classification Report:\n", classification_report(y_test, y_t_pred))
```

```
=== Tuned XGBoost ===
Accuracy:   0.6649484536082474
Confusion Matrix:
 [[47  4 11]
 [ 5 60 10]
 [ 5 30 22]]
Classification Report:
               precision    recall  f1-score   support

           0       0.82      0.76      0.79        62
           1       0.64      0.80      0.71        75
           2       0.51      0.39      0.44        57

    accuracy                           0.66       194
   macro avg       0.66      0.65      0.65       194
weighted avg       0.66      0.66      0.66       194
```

```
#ensemble
ensemble = VotingClassifier(estimators=[("logreg", logreg), ("rf", rf), ("xgb", best_xgb)],voting='soft',n_jobs=-1)
ensemble.fit(X_train, y_train)

y_e_pred  = ensemble.predict(X_test)
y_e_proba = ensemble.predict_proba(X_test)

print("=== Ensemble (RF + Tuned XGB) ===")
print("Accuracy:  ", accuracy_score(y_test, y_e_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_e_pred))
print("Classification Report:\n", classification_report(y_test, y_e_pred))
```

```
=== Ensemble (RF + Tuned XGB) ===
Accuracy:   0.6804123711340206
Confusion Matrix:
 [[50  2 10]
 [ 4 61 10]
 [ 7 29 21]]
Classification Report:
               precision    recall  f1-score   support

           0       0.82      0.81      0.81        62
           1       0.66      0.81      0.73        75
           2       0.51      0.37      0.43        57

    accuracy                           0.68       194
   macro avg       0.66      0.66      0.66       194
weighted avg       0.67      0.68      0.67       194
```

```
#cross validation
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
models_cv = {"LogisticRegression": make_pipeline(StandardScaler(),LogisticRegression(solver='saga',max_iter=5000,multi_class='multinomial')),"RandomForest": baseline_models["RandomForest"],"XGBoost": vanilla_xgb}

print("=== 5-Fold CV (Accuracy ± Std) ===")
for name, model in models_cv.items():
    scores = cross_val_score(model, X_train, y_train, cv=cv, scoring='accuracy')
    print(f"{name}: {scores.mean():.3f} ± {scores.std():.3f}")
```

```
=== 5-Fold CV (Accuracy ± Std) ===
LogisticRegression: 0.606 ± 0.022
RandomForest: 0.607 ± 0.027
XGBoost: 0.574 ± 0.032
```

Implications

- Arrest levels at NFL games can be predicted in advance using game context and team info
- Rivalry games, close scores, and away team wins are key risk factors
- Stadium security and law enforcement can use this model to identify high risk games early
- Helps guide staffing, alcohol policy, and crowd management decisions
- The same framework can be applied to other sports leagues or large public events
- Supports data driven approaches to improve fan safety and event planning