Traffic Violations Classification

PROJECT REPORT

GROUP -5

GROUP MEMBERS:

1. MD IBRAHIM SIDDIK (2211632042)
2.AHMED REZWAN KARIM (2132661642)
3. AFSANUL HAQUE (2132329642)

1. Revisited Dataset & Algorithm:

i) Dataset Justification:

Reasons to select: Multiple violations (4 to be precise) and multiclass target. Ideal for supervised multiclass prediction. Also has rich features set with 43 attributes including numerical and categorical data.

Why the dataset is suitable/challenging: The dataset is suitable because of the sufficient numbers of data which provides strong predictive values. Also, the dataset has missing values and categorical values which is challenging for ML modelling.

ii) Algorithm Rationale:

Algorithms strength and weakness:

K-NN: Easy to understand and implement and models complex decision boundaries without assumption. K-NN directly supports multiclass. K-NN does not require any training time. All assumptions are done during the prediction. The weakness is K-NN requires all features to be numeric. So, encoding is required for non-numerical data.

Logistic Regression: Logistic Regression is simple & interpretable. It is very fast to train. It also scales well. It is efficient with multiclass. The weakness is just like K-NN, it requires all features to be numeric. That means categorical features must be encoded. Also, if some class appears much more frequent than others, it can bias the model towards the majority class.

Random Forest: Random forests are well suited for handling large datasets because of ensemble learning and parallelization because it builds multiple independent decision trees. It is robust to high dimensionality and handle missing values and outliers better. Also, it takes minimal processing times. The weakness is that its computational complexity, slower performance compared to simpler models and lack of interpretability.

Why Random Forest is good fit for our project: Random Forest performs better than most of the models in our scenario because all the metrics lean towards the random forest. It performed better than KNN and Logistic Regression with the highest accuracy possible. It does not need any scaling. Also, KNN and Logistic Regression is not practical for categorical data.

iii) Preliminary Plan:

- Data Loading: Used custom ARFF file parser with CSV Handling with quote character correction (“”) for field parsing.
- Feature Engineering: Used only random 1M sample. Excluded non-predictive columns (seqid, date_of_stop, time_of_stop, geolocation, driver_license, registration). Used imputation for categorical/numerical values.
- Encoding Strategy: Used label encoding. (Warning = 0, Citation = 1 etc.)
- Scaling Strategy: Used standard scaler for KNN & Logistic Regression and no scaling is required for random forest because it is a tree-based algorithm. 80- 20 stratified train-test split and maintained class distribution.

2. Data Exploration & Analysis:

- ARFF File Structure: Contains @RELATION name and @ATTRIBUTE definitions.
- Starts with @DATA followed by comma-separated values.
- Takes 50,000 records for initial analysis.

```
Loading sample data for initial analysis...
Found 43 attributes
Data starts at line 48
Found 43 attributes
Data starts at line 48
Sample dataset shape: (49374, 43)
Sample dataset shape: (49374, 43)
```

Output Interpretation:

- Found 43 attributes: The dataset has 43 different features/columns
- Data starts at line 48 First 47 lines contain metadata and attribute definitions
- Sample dataset shape: (49374, 43)"**: Successfully loaded 49,374 records with 43 features

***49,374 < 50,000 indicates some records were filtered out due to parsing issues.

```
=== DATA TYPES AND INFO ===
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 49374 entries, 0 to 49373
Data columns (total 43 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   seqid                 49374 non-null  object
 1   date_of_stop          49374 non-null  object
 2   time_of_stop          49374 non-null  object
 3   agency                49374 non-null  object
 4   subagency             49374 non-null  object
 5   description           49374 non-null  object
 6   location              49374 non-null  object
 7   latitude              49374 non-null  object
 8   longitude             49374 non-null  object
 9   accident              49374 non-null  object
 10  belts                 49374 non-null  object
 11  personal_injury       49374 non-null  object
 12  property_damage       49374 non-null  object
 13  fatal                 49374 non-null  object
 14  commercial_license    49374 non-null  object
 15  hazmat                49374 non-null  object
 16  commercial_vehicle    49374 non-null  object
 17  alcohol               49374 non-null  object
 18  work_zone             49374 non-null  object
...
 41  arrest_type           49374 non-null  object
 42  geolocation           49374 non-null  object
dtypes: object(43)
memory usage: 16.2+ MB
```

Column Categories Analysis:
1. Identifiers: seqid, geolocation
2. Temporal Data: date_of_stop, time_of_stop
3. Administrative: agency, subagency, description, location
4. Geographic: latitude, longitude, state, driver_city, driver_state, dl_state
5. Incident Details: accident, personal_injury, property_damage, fatal
6. Safety Features: belts, commercial_license, hazmat, commercial_vehicle
7. Enforcement Context: alcohol, work_zone, arrest_type
8.Search Information: search_conducted, search_disposition, search_outcome, search_reason, search_reason_for_stop, search_type, search_arrest_reason
9. Vehicle Information: vehicletype, year, make, model, color
10.Violation Details: violation_type, charge, article, contributed_to_accident
11. Demographics: race, gender

Data Richness: The 43 features provide comprehensive coverage of:
- When (temporal patterns)
- Where (geographical analysis)
- Who (demographic analysis)
- What (violation types and details)
- How (enforcement methods)
- Why (contributing factors)

Traffic Violations by Day of Week (Top Left)

- Traffic Volume: Corresponds to commuter traffic patterns
- Tuesday Peak: Possibly intensive enforcement day or statistical anomaly
- Weekend Reduction: Lower traffic volume and different enforcement priorities

Traffic Violations by Month (Top Right)

- Weather Factor: Good weather = more driving = more violations
- School Calendar: September drop may correlate with school resumption
- Tourism Impact: Summer months see increased visitor traffic

Traffic Violations by Hour of Day (Bottom Left)

- Evening Focus: Higher evening enforcement
- Safety Priority: Late evening peak may target impaired driving
- Resource Allocation: Officer deployment matches traffic patterns

Traffic Violations by Year (Bottom Right)

Dataset Scope: Sample primarily from 2019 traffic violations

- Data Collection Period: Focused time frame rather than multi-year study
- Trend Analysis Limited: Cannot assess long-term trends from this sample

Distribution of Violation Types (Top Left - Pie Chart)

Educational Approach: Police prioritize warnings over punitive measures

- Revenue vs. Education: Emphasis on behavior modification rather than revenue generation
- Automated Enforcement: ESERO (Electronic Speed Enforcement) represents modern enforcement technology

Traffic Violations by Race (Top Right - Bar Chart)

- Proportional Representation: White and Black drivers show equal violation rates
- Demographic Reflection: May reflect county population demographics
- Enforcement Equity: Relatively balanced enforcement across racial groups

Distribution by Gender (Bottom Left - Pie Chart)

Gender Disparity: Males twice as likely to receive traffic violations

- Driving Behavior: May reflect differences in driving patterns, risk-taking, or exposure
- Social Factors: Could indicate occupational driving, commuting patterns, or behavioral differences

Top 10 Vehicle Types (Bottom Right - Horizontal Bar)

- Vehicle Fleet Composition: Reflects typical suburban vehicle ownership
- Automobile Dominance: 90% of violations involve standard passenger cars
- Truck Presence: Light duty trucks represent suburban/work vehicle use
- Specialty Vehicles: Low numbers for motorcycles, RVs, trailers reflect their limited use

3. Model Results and Implementation

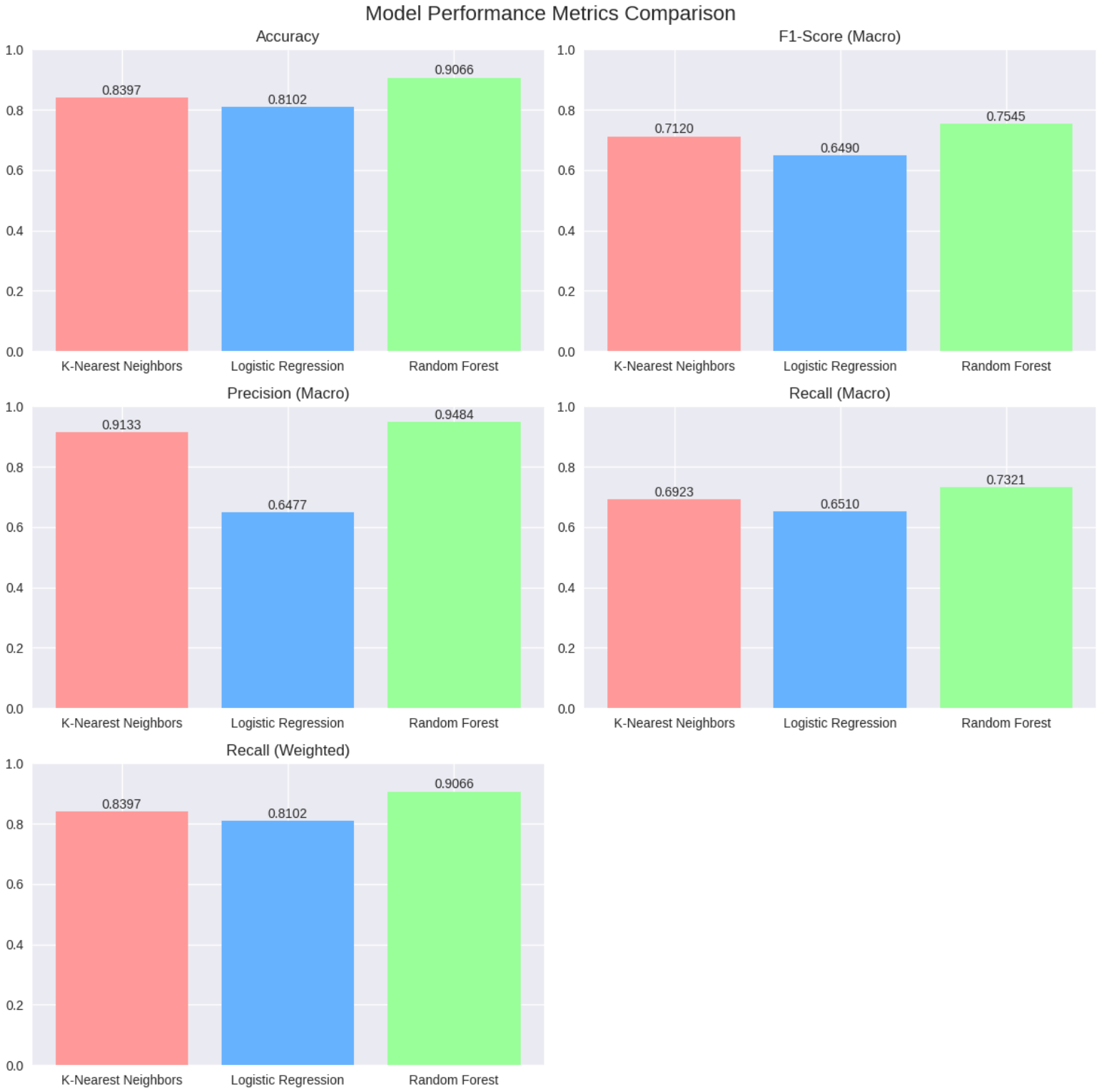

Table Summarization:

Model	Accuracy	F1-Score (Macro)	F1-Score (Weighted)	Precision (Macro)	Precision (Weighted)	Recall (Macro)	Recall (Weighted)	Train Size	Test Size
K-Nearest Neighbors	0.8397	0.7120	0.8396	0.9133	0.8408	0.6923	0.8397	800,000	200,000
Logistic Regression	0.8102	0.6490	0.8101	0.6477	0.8112	0.6510	0.8102	800,000	200,000
Random Forest	0.9066	0.7545	0.9065	0.9484	0.9078	0.7321	0.9066	800,000	200,000

Comparison and Contrast of Model Performance:

- **Ranking**: Random Forest leads with the highest accuracy (0.9066) and lowest MAE (0.2791), followed by K-Nearest Neighbors (accuracy: 0.8397), and Logistic Regression (accuracy: 0.8102).
- **Strengths**: Random Forest excels in precision (0.9484 macro) and handles class imbalance well. KNN performs decently in macro precision (0.9133), while Logistic Regression struggles with minority classes.
- **Weaknesses**: Logistic Regression has the lowest F1-macro (0.6490), indicating poor balance across classes. KNN lags in recall (0.6923)

4. Evaluation Metrics:

Metrics Used:

Accuracy, F1-Score (Macro), F1-Score (Weighted), Precision (Macro), Precision (Weighted), Recall (Macro), Recall (Weighted).

Justification:

Metrics—Accuracy, F1-Score (Macro), F1-Score (Weighted), Precision (Macro), Precision (Weighted), Recall (Macro), and Recall (Weighted)—are appropriate for classifying traffic violation dataset.

- Accuracy provides an overall performance baseline, critical for gauging general model effectiveness.
- F1-Score (Macro) and Recall (Macro) ensure balanced evaluation across potentially imbalanced violation classes (e.g., rare severe violations), preventing bias toward majority non-violation cases.
- Precision (Macro) and Recall (Macro) assess the model's ability to correctly identify violations and capture all actual violations, respectively, which is vital for public safety;
- Weighted versions (F1-Score, Precision, Recall) account for class distribution, reflecting real-world traffic data imbalances; together, they offer a comprehensive view of model performance, addressing both overall correctness and fairness across violation types.

5. Discussion & Conclusion

Model strengths and limitation:

- KNN shows excellent pattern recognition excellence with multi-class natural handling. Also captures local patterns. Limitation is computational burden with 1.5M+ records. KNN is hugely memory intensive and there is a curse of dimensionality.

- Logistic Regression is the fastest model in the whole dataset with better interpretability and memory efficient and robust baseline. Limitation is linear assumption and interaction blindness because it sometimes misses more complex patterns. Also, it is sensitive to outliers and may favor majority violation types.

- Random Forest handles mixed categorical and numerical features better. Also, it gives features importance which is the factors that matter most for traffic violations. It requires no scaling and captures complex patterns and also robust to missing values. It combines multiple decision paths for better accuracy. Limitation is hard to explain the individual predictions because black box by nature. Importantly, it is sensitive to hyperparameters. Performance depends on tree depth, nature of estimators.

Potential Improvements of Future Works:

- Extract temporal patterns (rush hour, seasonal trends), create interaction features (location × time), and derive behavioral metrics from violation history.
- combine all three models using voting classifiers, stacking, or boosting to leverage each algorithm's strengths for superior performance.
- Implement Grid Search, Random Search, or Bayesian optimization to fine-tune model parameters for optimal performance.
- Apply SMOTE, under sampling, or cost-sensitive learning to better handle minority violation types and improve recall.
- Build production pipeline with model monitoring, A/B testing, automatic retraining, and performance drift detection for continuous improvement