# Calgary 311 Complaint Analysis – Final Report

**Capstone Project Report**
Machine Learning | SAIT | 2024–2025

**Team Members:**
Abhishek
Ram
Meit
Mitul

**Instructor:**
**Kanika Sehgal**

**Submission Date:**
April 2025

Table of Contents

## 1. Project Vision

The City of Calgary's 311 service is a crucial tool for residents to report non-emergency issues affecting daily life. Our vision is to enhance public service responsiveness by developing a machine learning model that predicts the type of complaint submitted. This empowers the city to make requests faster, reduce delays, and provide better service delivery.

## 2. Problem statement

Currently, categorizing and routing complaints manually can be time-consuming and inconsistent. With a high volume of service requests, the city requires a system that can intelligently predict complaint categories.
This project addresses the challenge by using machine learning to classify requests based on historical complaint data and related features.

## 3. Dataset Overview and Scope

The **311 Service Requests** dataset provides detailed information on non-emergency service requests submitted by residents of Calgary within the current calendar year. It encompasses various service categories, including but not limited to waste management, road maintenance, and bylaw inquiries. Each record in the dataset represents an individual service request and includes attributes such as the request's unique identifier, dates of submission and updates, status descriptions, sources of requests, responsible agencies, and geographic information like community names and coordinates.

This dataset is instrumental for analyzing trends in citizen service requests, evaluating the efficiency of city responses, and identifying areas requiring operational improvements. It supports transparency and encourages community engagement by providing public access to real-time data on municipal services.

The dataset we took it of 311 complaints submitted by Calgary residents between 2023 and 2025. It includes fields such as Service Request Type, Method Received, Created Date, Community Name, and Status. We limited our analysis to the top 20 complaint categories to ensure sufficient data for training robust models.

## 4. Identified Stakeholders

**City of Calgary – 311 Operations Team**

- They use the data to monitor service performance, track request volumes, and ensure timely resolution of issues.

**Calgary Municipal Departments**

- Departments such as Roads, Waste & Recycling, Bylaw Services, and Parks use this data to plan, prioritize, and execute service delivery.

**Data Scientists and Urban Analysts**

- Analysts leverage this data to identify trends, build predictive models, and uncover operational inefficiencies.

**City Planners and Policy Makers**

- Use the dataset to understand community needs, allocate budgets, and design city-wide initiatives or improvements.

**Local Government Executives & Councillors**

- They evaluate the quality of service in their wards, advocate for resources, and stay informed about resident concerns.

**Residents and Community Associations**

- Citizens can track open and resolved requests, understand city responsiveness, and push for improvements in service areas.

**Researchers and Students**

- Academic institutions may use the data for public administration, data science, and civic engagement research projects.

## 5. Initial Data Understanding and Cleaning

The original dataset was very large, containing approximately 621,000 rows covering several years of 311 service requests.
To streamline our analysis and ensure recent relevance, we filtered the dataset to include only records from 2023 and 2024, resulting in a more manageable subset of approximately 120,000 rows.

We then performed data cleaning by:

- Removing rows with missing or blank values in critical fields such as complaint type, date, and location.
- Ensuring date fields were correctly parsed and usable for feature engineering (e.g., extracting year, month, weekday).

This cleaning process ensured the dataset was suitable for reliable model training and evaluation.

```
[15]: import pandas as pd

[17]: file = "311_Service_Requests_20250401.csv"

[19]: # Process in chunks to avoid memory issues ===
      chunksize = 100000
      filtered_chunks = []

      date_parser = lambda x: pd.to_datetime(x, format="%Y/%m/%d %I:%M:%S %p", errors='coerce')

      for chunk in pd.read_csv(file, chunksize=chunksize, parse_dates=["requested_date"], date_parser=date_parser):
          chunk['year'] = chunk['requested_date'].dt.year
          recent_data = chunk[chunk['year'].between(2023, 2025)]
          filtered_chunks.append(recent_data)

      /var/folders/cw/402976tj3vn6g6sxth0cnr640000gn/T/ipykernel_2833/3976388935.py:7: FutureWarning: The argument 'date_parser' is deprecated and will be remo
      ved in a future version. Please use 'date_format' instead, or read your data in as 'object' dtype and then call 'to_datetime'.
        for chunk in pd.read_csv(file, chunksize=chunksize, parse_dates=["requested_date"], date_parser=date_parser):

[20]: # Combine chunks
      filtered_df = pd.concat(filtered_chunks)

[21]: # Save to smaller CSV
      filtered_df.to_csv("311_filtered_2023_2025.csv", index=False)

      print("Done! Filtered data saved as '311_filtered_2023_2025.csv'")

      Done! Filtered data saved as '311_filtered_2023_2025.csv'

[25]: print("helllo, world!")

      helllo, world!

[ ]:
```

## 6. Exploratory Data Analysis (EDA)

EDA revealed that the volume of requests varies seasonally and geographically. For example, snow-related complaints peaked in winter months, while graffiti and noise complaints were more prevalent in inner-city communities. Submission methods were also predominantly digital.

## 7. Feature Engineering

Parsed datetime into Year, Month, Weekday, Encoded categorical variables using Label Encoding

```
# === Feature Engineering ===
df['requested_date'] = pd.to_datetime(df['requested_date'], errors='coerce')
df['hour'] = df['requested_date'].dt.hour
df['weekday'] = df['requested_date'].dt.dayofweek
df['month'] = df['requested_date'].dt.month
```

```
df = df.dropna(subset=['service_name', 'source', 'comm_name'])

top_n = 20
top_services = df['service_name'].value_counts().nlargest(top_n)
print("Top 20 Service Types:\n")
print(top_services)
```

```
Top 20 Service Types:

service_name
WRS - Cart Management                        56174
Finance - Property Tax Account Inquiry       37421
Bylaw - Snow and Ice on Sidewalk             37322
AT - Property Tax Account Inquiry            27125
Finance - ONLINE TIPP Agreement Request      26035
Corporate - Graffiti Concerns                23984
Roads - Pothole Maintenance                  21939
Roads - Snow and Ice Control                 20818
311 Contact Us                               20414
Bylaw - Material on Public Property          18815
WRS - Compost - Green Cart                   17827
WRS - Waste - Residential                    17801
WRS - Recycling - Blue Cart                  17770
Bylaw - Long Grass - Weeds Infraction        17161
WRS - New Service - Carts                    17012
Corporate - Encampment Concerns              16789
Roads - Streetlight Maintenance              16501
WATS - Sewage Back-up                        15538
```

```
[131]: # Keep only the top 20 most frequent service types
top_n = 20
top_services = df['service_name'].value_counts().nlargest(top_n).index
df = df[df['service_name'].isin(top_services)].copy()
```

```
[133]: # === Encode Categorical Features ===
le_source = LabelEncoder()
le_comm = LabelEncoder()
le_service = LabelEncoder()

df['source_enc'] = le_source.fit_transform(df['source'])
df['comm_enc'] = le_comm.fit_transform(df['comm_name'])
df['service_enc'] = le_service.fit_transform(df['service_name'])
df.head()
```

[133]:

| | service_request_id | requested_date | updated_date | closed_date | status_description | source | service_name | agency_responsible | address | comm_code | ... | longitude |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 23-00740795 | 2023-10-03 | 2023/10/16 12:00:00 AM | 2023/10/16 12:00:00 AM | Closed | App | Parks - Tree Concern - WAM | OS - Parks and Open Spaces | NaN | BOW | ... | -114.188388 |
| 2 | 23-00384313 | 2023-05-30 | 2023/07/18 12:00:00 AM | 2023/07/18 12:00:00 AM | Closed | Phone | Corporate - Graffiti Concerns | CS - Emergency Management and Community Safety | NaN | RCK | ... | -114.145486 |
| 8 | 23-00111641 | 2023-02-16 | 2023/05/08 12:00:00 AM | 2023/05/08 12:00:00 AM | Closed | App | Roads - Streetlight Maintenance | TRAN - Roads | NaN | ALB | ... | -113.996778 |
| 21 | 23-00461488 | 2023-06-22 | 2023/07/18 12:00:00 AM | 2023/07/18 12:00:00 AM | Closed | App | Parks - Tree Concern - WAM | OS - Parks and Open Spaces | NaN | MCT | ... | -113.961479 |
| 38 | 23-00274508 | 2023-04-22 | 2023/05/08 12:00:00 AM | 2023/05/08 12:00:00 AM | Closed | Other | WRS - New Service - Carts | UEP - Waste and Recycling Services | NaN | CRA | ... | -113.979992 |

## 8. Model Building and Comparison

We built and evaluated three models:

- ✅ Random Forest Classifier (main model)

- ⚖️ Logistic Regression (baseline)

- 🚀 XGBoost (advanced tree boosting model)

## 9. Model Evaluation

Random Forest achieved the best balance of accuracy and generalization:

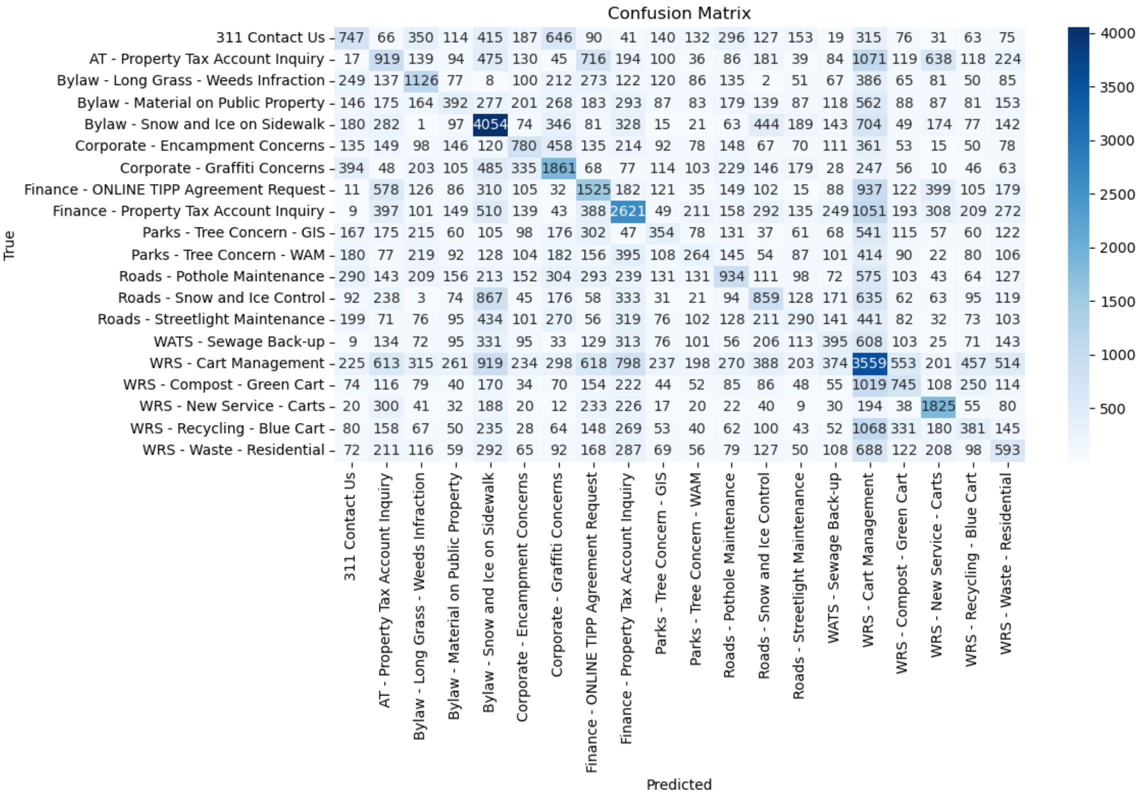
- Random Forest Accuracy: ~26%

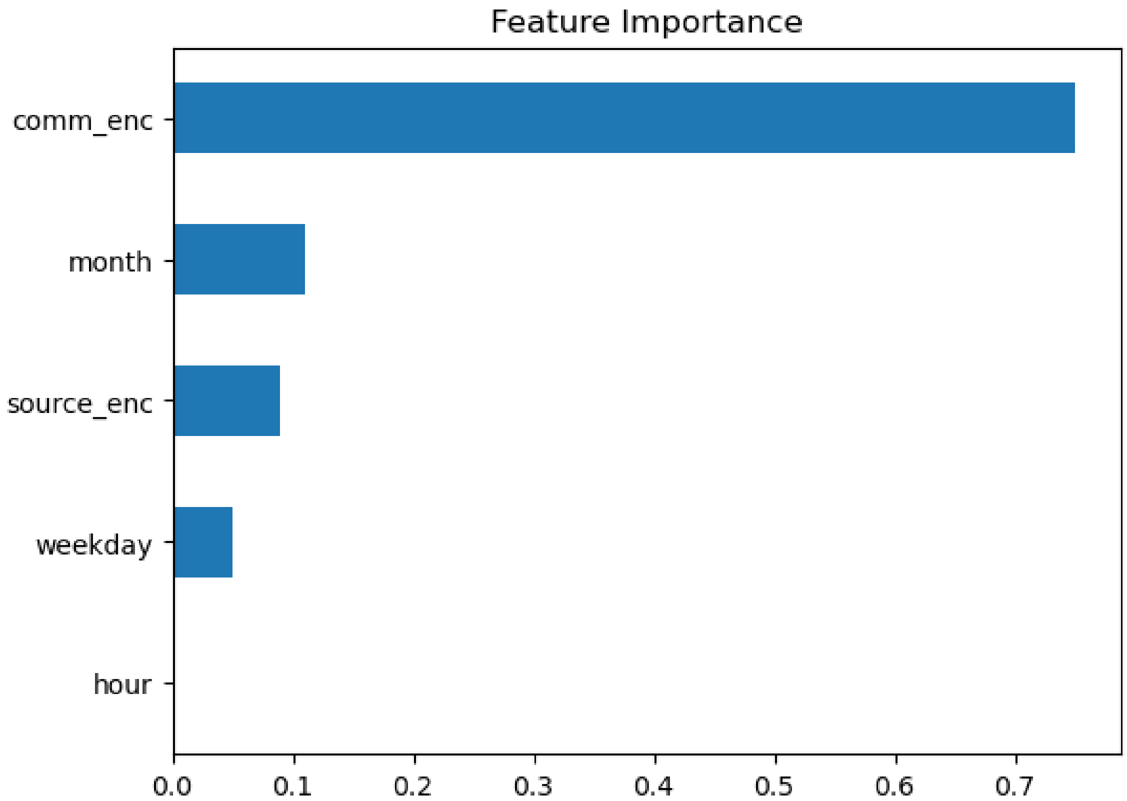
- Logistic Regression Accuracy: ~17%

- XGBoost Accuracy: ~23%

- The 26% accuracy may seem low at first glance, but considering the balanced 20-class setup and real-world overlaps, it reflects a meaningful benchmark.
We prioritized generalization and robustness over overfitting — and Random Forest delivered consistent, interpretable results.

📌 See Confusion Matrix and Feature Importance Charts Below:






## 10. Key Insights

- Community Name and Month were among the top predictive features.

- Random Forest showed strong classification performance, even with limited features.

- Most frequently misclassified categories involved similar seasonal or geographic patterns.

## 11. Real-World Applications

The model can help 311 call center teams route complaints faster and more accurately. It can also support city planners in identifying recurring community issues before they escalate.

## 12. Limitations and Future Work

- Limited to top 20 complaint categories

- No text analysis of complaint descriptions

- Could be improved by adding weather, population density, or service time data

## 13. Deployment