Optimizing Insurance Processing of Insurance Claims Through Automation and Machine Learning

- By Jerry Akom
15th December 2024

CIS 731

Problem Statement

Challenges in Insurance Claims Processing:

- Manual evaluations are time-consuming.
- Human errors lead to inconsistencies.
- Increasing fraud detection needs.

Proposed Solution:

- Automate car damage severity detection using AI/ML models.
- Streamline claims processing workflows.

Objectives

- Automate car damage severity detection.
- Improve accuracy and efficiency in claims evaluation.
- Reduce manual effort and operational costs.
- Enhance customer satisfaction with quicker claims processing.

Methodology Overview

- Fig 1 shows a representation of the Smart Claims service which forms the backbone of this project. Based on this this I deduced three major phases for this project: (1) data collection and transformation, (2) image classification (car damage severity detection), (3 accident scoring, claims determination and visualization. The project was conducted on the databricks platform and with processing cluster and storages built on Amazon Web Services (AWS) according to requirements specified in Smart Claims Demo.

1. Data Collection and Transformation

- All project datasets come from Smart Claims Datasets smart-claims project (Smart Claims Demo). It is part of the Smart Claims Datasets demo portfolio used to showcase their platform and its capabilities. The data usage is restricted to the Smart Claims Datasets platform only (Smart Claims Datasets).
- The main datasets used are: a) Images (.png) - contains images of 56 cars. It contains data car in three damage severity categories (major - 18, minor -21, ok -17). This is the major dataset used in model training/finetuning and validation. Some examples are shown in in Fig 2, b) Claims (.json) – contains insurance claims data, incident (accident) information and some customer data,. C) Policy (.csv) - contains insurance policy information including policy type, customer and vehicle information. d) Telematics (.parquet) - contains data of car location and driving speed data, which represents mileage and driving habits. e) Accidents (.jpg) – Accident images (together with their metadata in csv format) submitted by customers when they make claims claims, used for claims scoring. While the claims and policy data are synthetic datasets, the images are of real cars scraped from the internet while the telematics data was generate using the open NYC Taxi dataset but reveal information regarding real people.

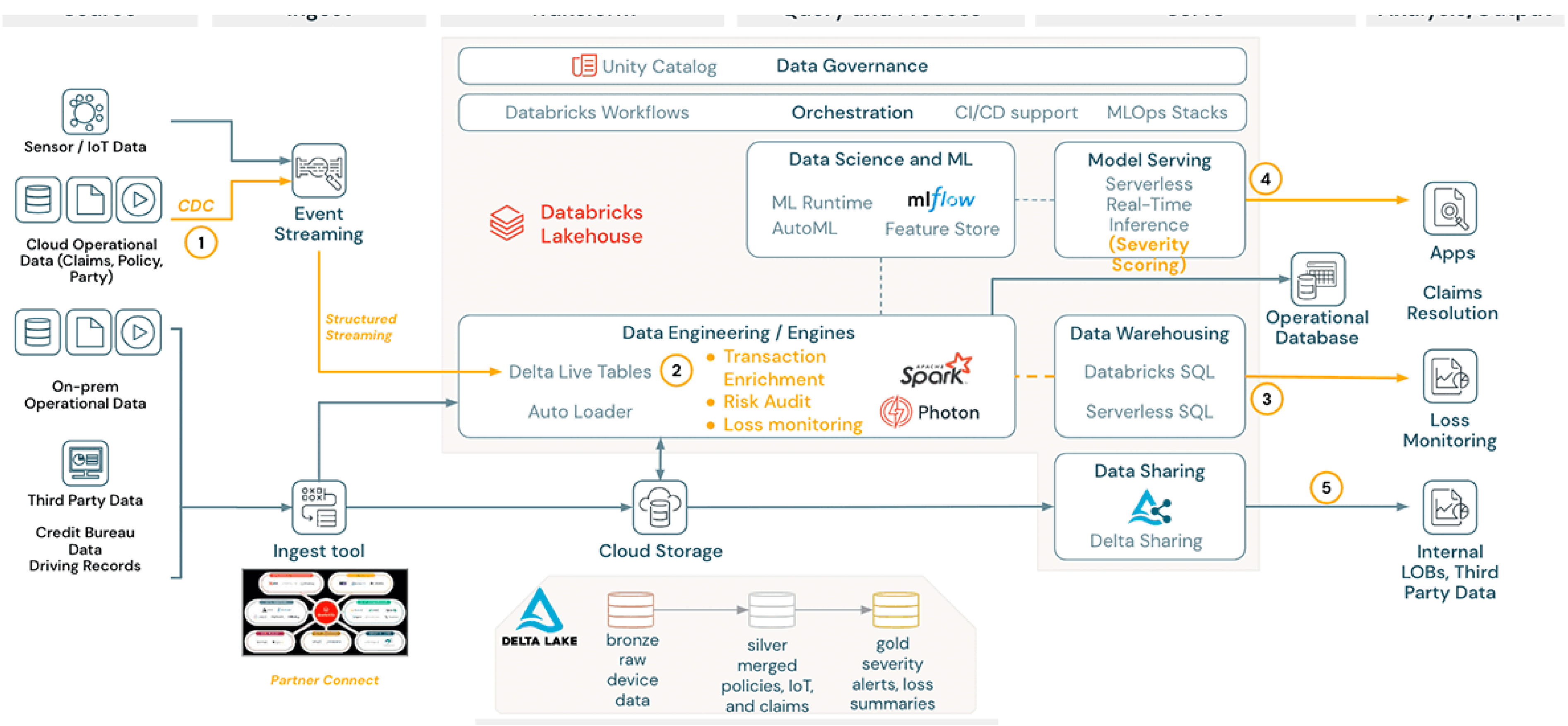
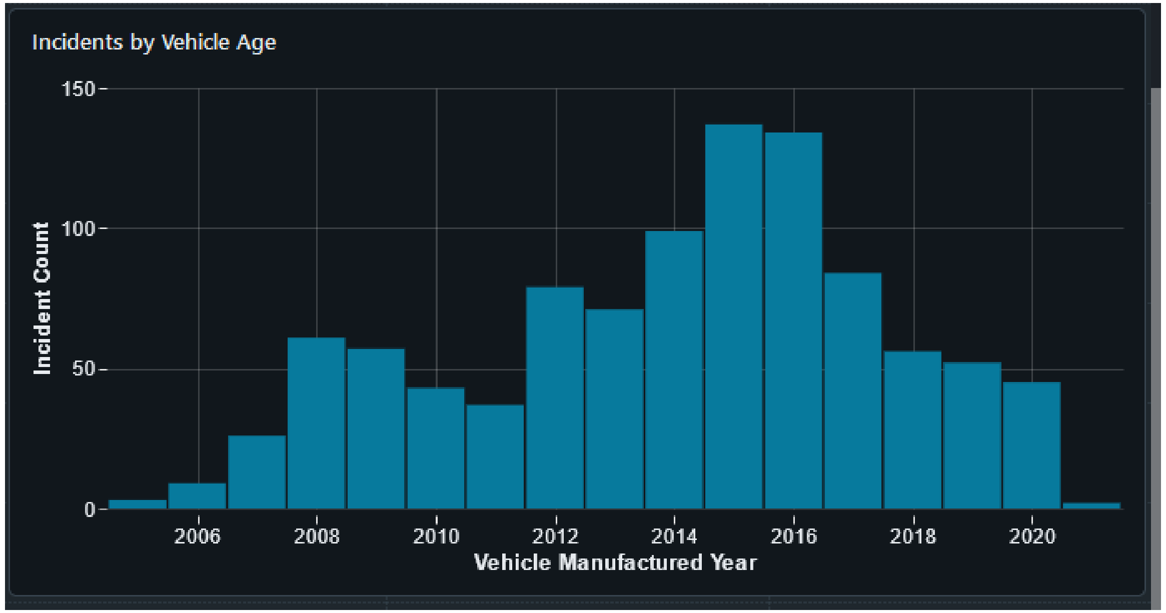


Fig 1. Workflow of Smart Claims Datasets smart-claim platform.

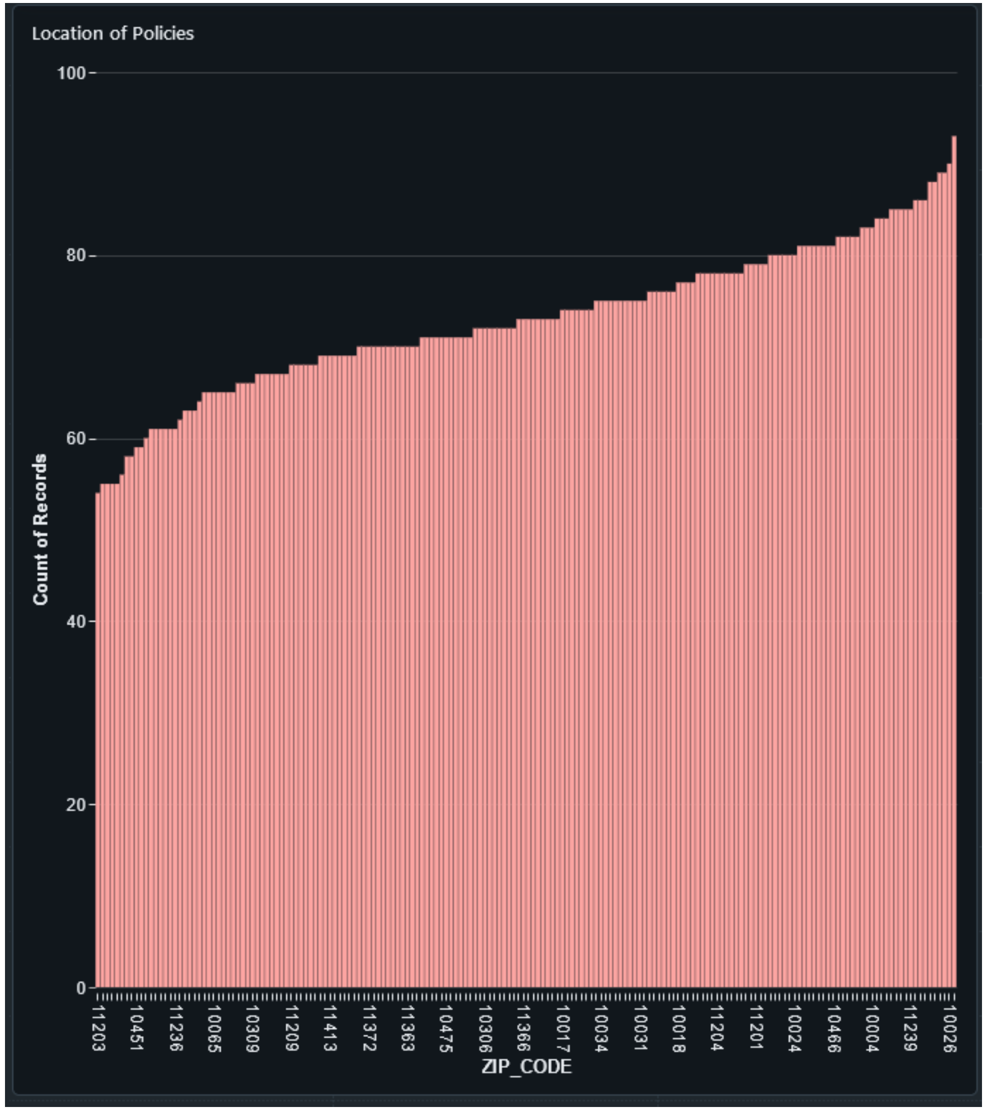
Fig 2. Examples of car damage severity levels

Data Insights

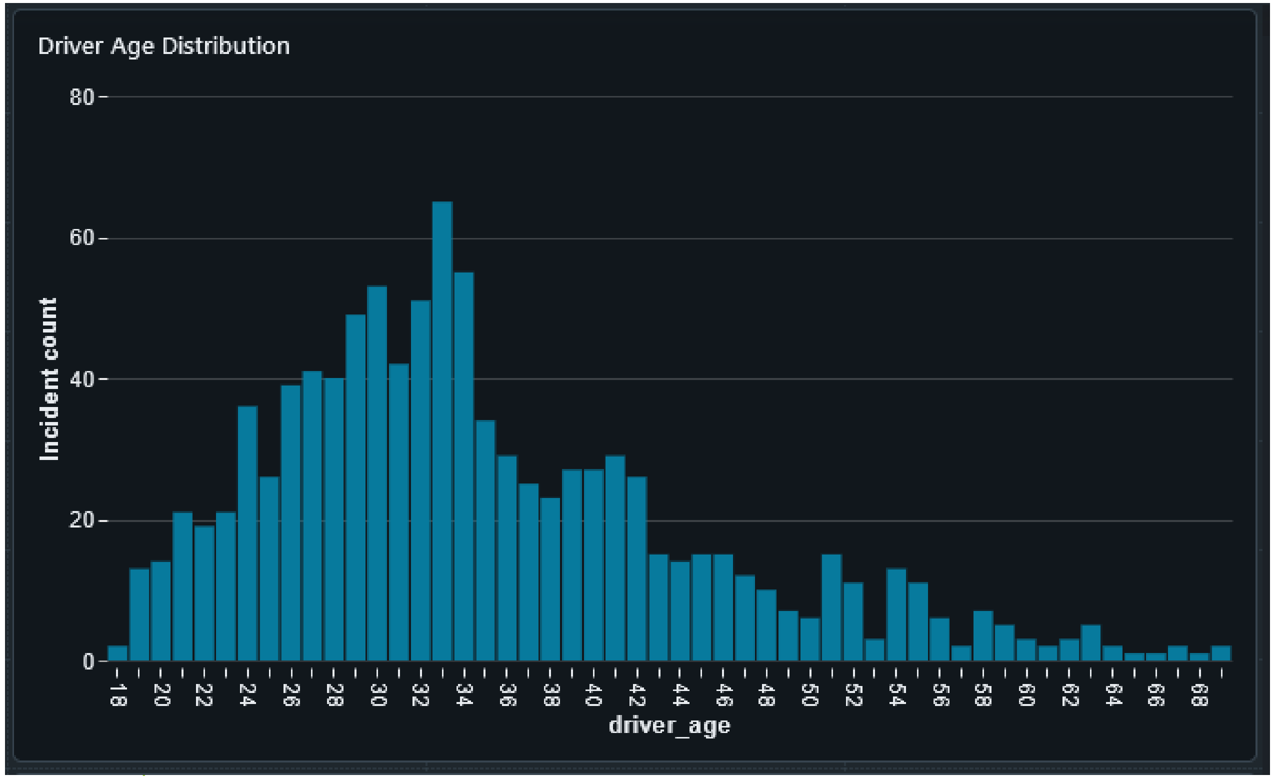


Incident count by year of vehicle manufacture.

- **Key Features**:
  - Accident Type
  - Speed (e.g., high-speed cases)
  - Vehicle Telematics
- **Data Challenges**:
  - Synthetic nature of datasets.
  - Missing values and feature imbalances.



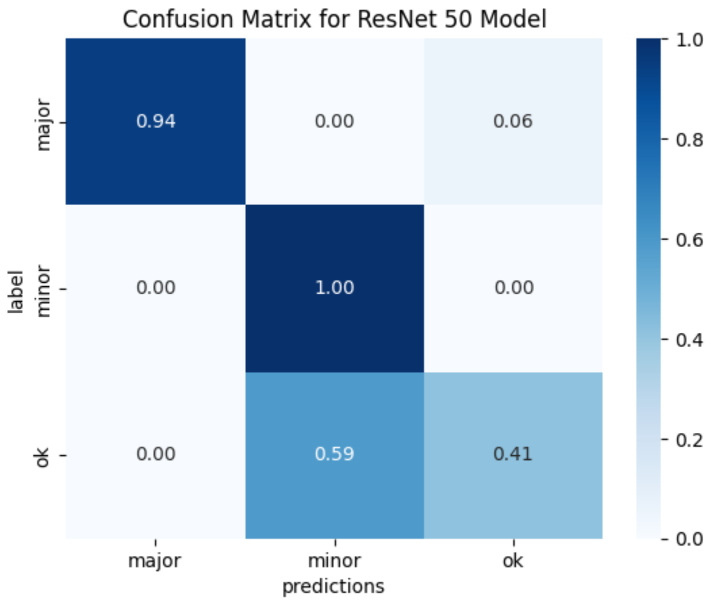
Number of accidents by zipcode in NYC

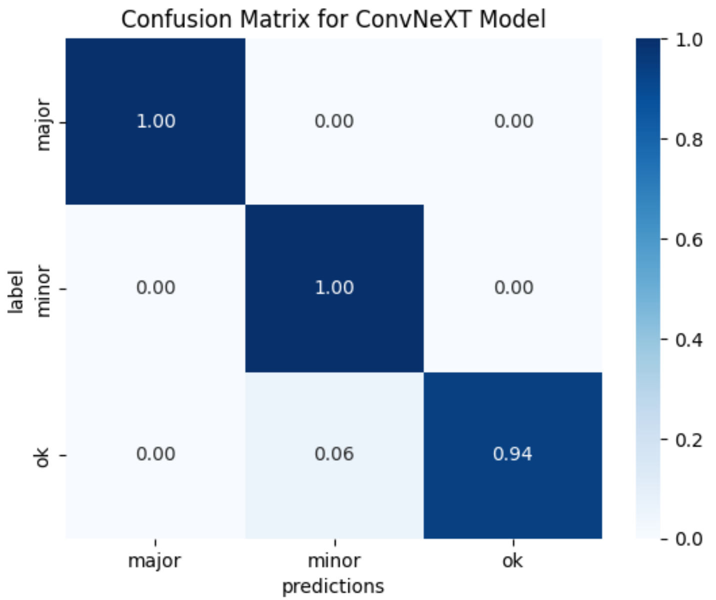


Incident count by age.

Model Development

- **Models Used**:
  - ResNet: Deep residual learning for image classification.
  - ConvNeXT: Modernized convolutional network.
- **Training Details**:
  - Training/validation split.
  - Data augmentation for generalization.
- **Performance Metrics**:
  - Accuracy, Precision, Recall, F1-Score.





Model Performance Comparison

ResNet:

Accuracy: 82%

F1-Score: Moderate.

ConvNeXT:

Accuracy: 91%

F1-Score: High.

Key Insight:

ConvNeXT outperformed ResNet due to architecture improvements.

Challenges and Limitations

- Synthetic datasets may not generalize well.
- Model biases due to unbalanced data.
- Integration challenges with existing systems.
- Need for large-scale real-world validation.

Recommendations for Future Work

- Incorporate real-world datasets for better generalization.
- Address model biases with balanced training data.
- Expand telematics integration for richer insights.
- Validate models on live claims data.

Q&A

Questions or feedback?

- Thank you