Course: DAMG 7370 Project Final Report
Instructor: Professor Zheng Zheng
Date: Dec 10th 2025
Project Title: US Accidents Data Architecture & Machine Learning Group

Group Members:

Group Member	Student ID	Role
Jerry Yao Deku	002680048	Lead Data Engineer • Responsible for data preprocessing, feature engineering, and pipeline automation.
Issaka Mohammed	002430679	Machine Learning Engineer • Handles model selection, training, evaluation, and optimization • Development QuickSight dashboard
Obed Kaburi	002804019	DevOps & Deployment Lead • Manages deployment to HuggingFace Spaces, requirements, CI/CD, and cloud integration.
Reuben Comla	002317328	Documentation & Reporting Lead • Prepares technical documentation, project reports, and coordinates presentations.

Executive Summary

This project presents a comprehensive end-to-end data engineering and machine learning pipeline for analyzing and predicting the severity and duration of US traffic accidents. Leveraging a dataset of over 7.7 million accident records from 2016–2023, our team designed a robust architecture that integrates local data processing in Jupyter with scalable cloud analytics on AWS.

Key achievements include:

- Building a star schema dimensional model in Jupyter, then exporting and uploading data to Amazon S3.
- Solving real-world ETL challenges to ensure Redshift compatibility, including multi-step data cleaning and PSV export.
- Loading dimension and fact tables into Amazon Redshift for scalable SQL analytics and dashboarding in Amazon QuickSight.
- Engineering 22+ features for predictive modeling and training multiple machine learning models (Random Forest, XGBoost, Gradient Boosting, and ensemble methods) in Jupyter.
- Addressing class imbalance to improve detection of rare but critical accident types.
- Deploying an interactive web application on HuggingFace Spaces for real-time accident risk prediction.
- Delivering actionable insights for emergency response, traffic management, and infrastructure planning.

Category	Details
Data Source	US Accidents (Kaggle), 7.7M records, 2016–2023
Data Processing	Jupyter Notebook (cleaning, dimensional modeling, export)
Cloud Pipeline	AWS S3 (storage), Redshift (warehouse), QuickSight (BI)
Dimensional Model	4 dimensions + 1 fact table (star schema)
ML Models	RF, XGBoost, GB, Ensemble (classification & regression)
Feature Engineering	22+ engineered features
Class Imbalance	Addressed with balanced XGBoost
Deployment	HuggingFace Spaces (Gradio web app)
Key Metrics	Severity accuracy: 81.5%, Duration MAE: 11.8 min
Visualizations	EDA charts, QuickSight dashboards
Business Impact	Real-time risk prediction, resource allocation, planning

Table 1: Project Highlights

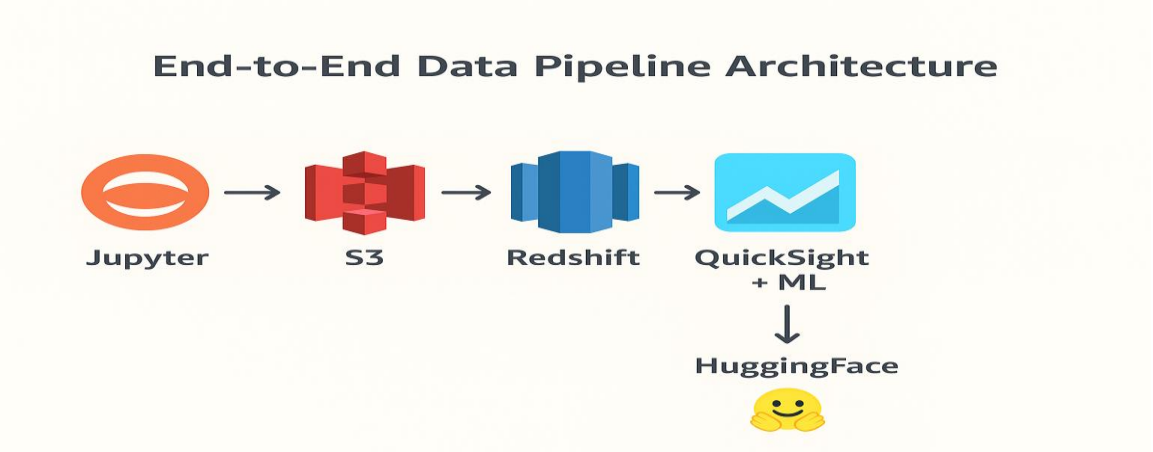


Fig 1: End-to-End Data Pipeline Architecture

This report details the full pipeline, from raw data ingestion and dimensional modeling in Jupyter, through cloud-based warehousing and analytics on AWS, to public ML deployment, demonstrating both technical depth and practical impact.

1. Data Architecture & Engineering

1.1 Data Source

The project utilizes the US Accidents dataset from Kaggle, comprising over 7.7 million records collected between 2016 and 2023. The dataset covers 49 US states and includes 47 original features such as location, weather, time, and road conditions.

Source: Kaggle US Accidents
Records: 7,700,000+
Time Span: 2016–2023
Features: 47
Coverage: 49 states

	ID	Source	Severity	Start_Time	End_Time	Start_Lat	Start_Lng	End_Lat	End_Lng	Distance(mi)	...	Roundabout	Station	Stop	Traffic_Calming	Traffic_Signal	Turning_Loop	Sunrise_Sunset	Civil_Twilight	Nautical_Twilight	Astronomical_Twilight
0	A-1	Source2	3	2016-02-08 05:46:00	2016-02-08 11:00:00	39.865147	-84.058723	NaN	NaN	0.01	...	False	False	False	False	False	False	Night	Night	Night	Night
1	A-2	Source2	2	2016-02-08 06:07:59	2016-02-08 06:37:59	39.928059	-82.831184	NaN	NaN	0.01	...	False	False	False	False	False	False	Night	Night	Night	Day
2	A-3	Source2	2	2016-02-08 06:49:27	2016-02-08 07:19:27	39.063148	-84.032608	NaN	NaN	0.01	...	False	False	False	False	True	False	Night	Night	Day	Day
3	A-4	Source2	3	2016-02-08 07:23:34	2016-02-08 07:53:34	39.747753	-84.205582	NaN	NaN	0.01	...	False	False	False	False	False	False	Night	Day	Day	Day
4	A-5	Source2	2	2016-02-08 07:39:07	2016-02-08 08:09:07	39.627781	-84.188354	NaN	NaN	0.01	...	False	False	False	False	True	False	Day	Day	Day	Day

Fig 2: Dataset features

1.2 Data Cleaning & Preprocessing (Jupyter)

All data cleaning and preprocessing were performed in Jupyter Notebook using pandas and numpy. The process included:

Missing Value Handling:

Imputed or removed missing values in key columns (e.g., latitude, longitude, severity, weather).
Dropped records with critical missing data that could not be imputed.

Duplicate & Outlier Removal:

Removed duplicate accident records based on unique accident IDs.
Detected and removed outliers in latitude/longitude and duration.

Geographic Validation:
Ensured all records had valid US coordinates.
Filtered out records with impossible or missing location data.
Text Normalization:
Cleaned text fields (e.g., Description, Street, Weather_Condition) to remove problematic characters (commas, pipes, tabs, newlines). Standardized categorical values (e.g., weather categories).
Result:
Produced a cleaned DataFrame (df_clean) with high data integrity, ready for modeling and export.

1.3 Dimensional Modeling (Star Schema)
A star schema was designed and implemented in Jupyter to support scalable analytics and efficient querying in Redshift. The schema consists of four dimension tables and one fact table:

dim_location:
accident_id, latitude, longitude, city, state, county, zipcode

dim_weather:
accident_id, temperature_f, humidity_pct, pressure_in, visibility_mi, wind_speed_mph, precipitation_in, weather_condition, weather_category

dim_time:
accident_id, start_time, end_time, hour, day_of_week, month, year, is_weekend, duration_minutes
dim_road_features:

accident_id, Traffic_Signal, Stop, Crossing, Junction, Railway, Station, and 7 other boolean features

fact_accidents:
accident_id, severity, start_time, end_time, duration_minutes, hour, day_of_week, month
The star schema enables efficient joins and aggregations for downstream analytics.

1.4 ETL & Export to AWS
The ETL process for this project was executed within a Jupyter Notebook environment, where the primary objective was to transform the cleaned datasets into a format suitable for ingestion into Amazon Redshift. Early in the workflow, exporting the transformed tables as standard CSV files proved problematic. Several of the text fields contained commas, newline characters, and even pipe symbols, elements that caused Redshift's COPY command to misinterpret field boundaries and break the parsing logic. These issues made it clear that a more controlled export strategy was required.

To address this, multiple export configurations were tested iteratively. Over six different approaches were evaluated, including standard CSV output, CSV with QUOTE_ALL, tab-delimited files, pipe-delimited files, and variations with different quoting and escape rules. Each attempt surfaced new edge cases, particularly around inconsistent text formatting inherited from the source data. This iterative testing phase was essential for identifying a format that balanced readability, compatibility, and robustness during Redshift ingestion.

The final solution involved cleaning all text fields to remove or normalize problematic characters that could interfere with the COPY process. After sanitizing the data, each table was exported as a pipe-separated values (PSV) file. This format minimized the need for heavy quoting while still providing a clear delimiter unlikely to appear in the cleaned text fields. Proper escaping rules were applied to ensure that any remaining special characters would not disrupt parsing. This approach resulted in a stable and repeatable export process that integrated cleanly with Redshift.

The following PSV files were generated as the final outputs of the ETL export stage:

- **dim_location.psv**
- **dim_weather.psv**
- **dim_time.psv**
- **dim_road_features.psv**

- **fact_accidents.psv**

These files were then ready for efficient loading into AWS Redshift as part of the downstream data warehousing workflow.

1.5 AWS Cloud Pipeline

Once the local ETL processing was complete, the workflow transitioned into the AWS cloud environment to take advantage of scalable storage, high-performance warehousing, and enterprise-grade visualization tools. This cloud-based segment of the pipeline ensured that the transformed datasets could be stored reliably, queried efficiently, and visualized interactively for downstream analysis.

The first stage involved uploading all pipe-separated value (PSV) files to Amazon S3. S3 served as the central storage layer, providing durable, cost-effective, and highly scalable object storage. By placing the exported tables in S3 buckets, the pipeline established a persistent data repository that could be accessed by other AWS services without the need for additional data movement or duplication.

From S3, the data was ingested into Amazon Redshift using the COPY command, which is optimized for high-throughput bulk loading. Each PSV file was mapped to its corresponding table within the Redshift cluster, preserving the star schema design established during the modeling phase. This schema, comprising dimension tables and a central fact table enabled efficient SQL querying, reduced join complexity, and supported analytical workloads such as aggregations, trend analysis, and drill-downs.

For visualization and business intelligence, Amazon QuickSight was connected directly to the Redshift data warehouse. This integration allowed for fast, interactive dashboarding without the need to extract or replicate data. Within QuickSight, a suite of dashboards was developed to explore key analytical themes, including accident severity distribution, temporal patterns across hours and seasons, geographic clustering of incidents, and the influence of weather conditions. These visualizations provided intuitive insights and supported data-driven decision-making.

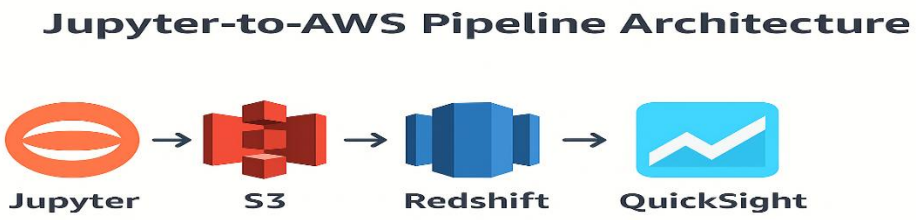


Fig 3: Jupyter to AWS Pipeline Architecture

1.6 Key Challenges & Solutions

Throughout the development of the end-to-end data pipeline, several technical and architectural challenges emerged. These obstacles required careful diagnosis, iterative refinement, and strategic decision-making to ensure the pipeline remained robust, scalable, and reproducible.

Data Quality

One of the earliest and most persistent challenges involved data quality. The raw datasets contained a mix of inconsistencies, missing values, and malformed entries, particularly in text-heavy fields such as location descriptions and weather notes. These issues posed risks to downstream analytics and schema enforcement. To address this, a comprehensive data cleaning and validation routine was implemented within the Jupyter environment. This included null handling, type coercion, character normalization, and outlier detection. By enforcing strict validation rules and sanitizing inputs, the pipeline ensured that only clean, well-structured data progressed to the export and cloud stages.

ETL Complexity
The ETL process itself presented a series of technical hurdles, especially during the export phase. Initial attempts to export data using standard CSV formats failed due to embedded commas, newlines, and pipe characters in text fields, which disrupted parsing in Amazon Redshift. These failures led to multiple rounds of testing with alternative formats and quoting strategies. Ultimately, the solution involved cleaning all text fields to remove problematic characters and exporting the tables as pipe-separated values (PSV) with minimal quoting and proper escaping. This approach stabilized the ingestion process and preserved schema fidelity.

Cloud Integration
Transitioning from local processing in Jupyter to cloud-based infrastructure in AWS required careful orchestration. Maintaining data integrity and schema consistency across services (Amazon S3, Redshift, and QuickSight) was critical. The pipeline was designed to ensure seamless handoff between stages, with each component configured to respect the structure and constraints of the data. This included precise mapping of PSV files to Redshift tables using COPY commands, and schema-aware dashboarding in QuickSight. The result was a tightly integrated cloud pipeline capable of supporting scalable analytics and visualization.

Challenge	Description	Solution/Approach
Data Quality Issues	Missing values, duplicates, outliers, invalid coordinates	Imputation, removal, validation, and filtering in Jupyter
Text Field Formatting	Commas, newlines, and pipes in text fields broke CSV/PSV exports	Cleaned all text fields, removed/replaced problematic characters
CSV Export for Redshift	Standard CSV export failed due to delimiter conflicts and parsing errors	Iterative export attempts; switched to pipe-separated values (PSV)
Schema Consistency	Ensuring dimension and fact tables matched star schema in Redshift	Defined schema in Jupyter, validated before export and after loading
Large Data Volume	7.7M records required efficient processing and export	Used pandas chunking and optimized export parameters
Cloud Integration	Seamless transfer from Jupyter to AWS S3 and Redshift	Automated upload scripts and Redshift COPY commands
Data Integrity in Cloud	Ensuring no data loss or corruption during upload and loading	Used checksums, row counts, and validation queries post-load

Table 2: Data Engineering Challenges & Solutions

2. Exploratory Data Analysis (EDA)

2.1 Severity Distribution Analysis
The first step in the exploratory data analysis focused on understanding how accident severity is distributed across the entire dataset. This analysis is essential because severity levels directly influence downstream modeling decisions, resource allocation strategies, and the interpretability of any predictive insights. By examining the frequency of each severity class, we gain a clearer picture of the underlying patterns and potential biases present in the data.

The primary objective of this analysis was to quantify the proportion of accidents falling into each severity category and identify any imbalances that might affect statistical modeling or machine learning performance. The results revealed a highly skewed distribution: Severity 2 (moderate) accidents dominate the dataset, accounting for approximately **80%** of all recorded incidents. This overwhelming concentration suggests that moderate accidents are far more common than either minor or severe ones.

In contrast, Severity 1 (minor) and Severity 4 (major) accidents appear only infrequently. Their low representation highlights a significant class imbalance, which has important implications for both descriptive analytics and predictive modeling. Models trained on such data may naturally gravitate toward predicting the majority class unless corrective measures such as resampling, class weighting, or anomaly-focused modeling are applied.

Overall, the severity distribution analysis provides a foundational understanding of the dataset's structure and informs the methodological choices for subsequent stages of the project.

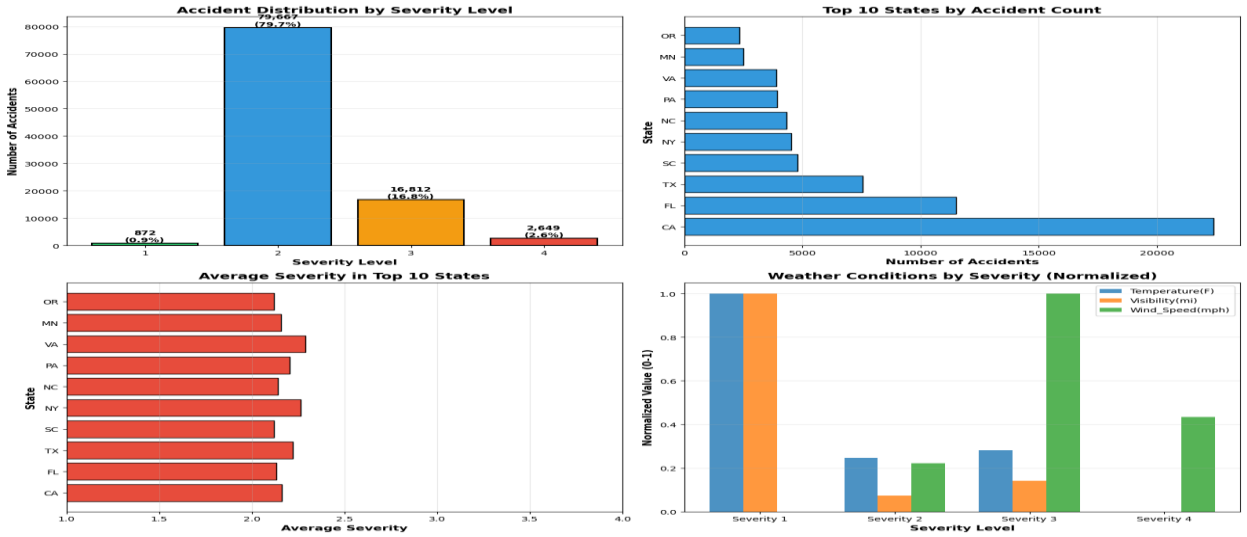


Fig 4: Severity Distribution Bar Chart

2.2 Temporal Pattern Analysis

This analysis examined how accident frequency changes across different time periods, including hour of day, day of week, and season. Clear hourly trends emerged, with accidents peaking during rush hours, specifically 7–9 AM and 4–7 PM when traffic volume and congestion are highest.

A weekday pattern was also evident: accident rates were consistently higher on weekdays than weekends**,** reflecting the influence of work-related travel and structured commuting routines.

Seasonally, the data showed that winter months experienced more severe accidents, likely due to hazardous conditions such as snow, ice, and reduced visibility. These temporal insights help explain when risks are elevated and guide both modeling decisions and safety interventions.

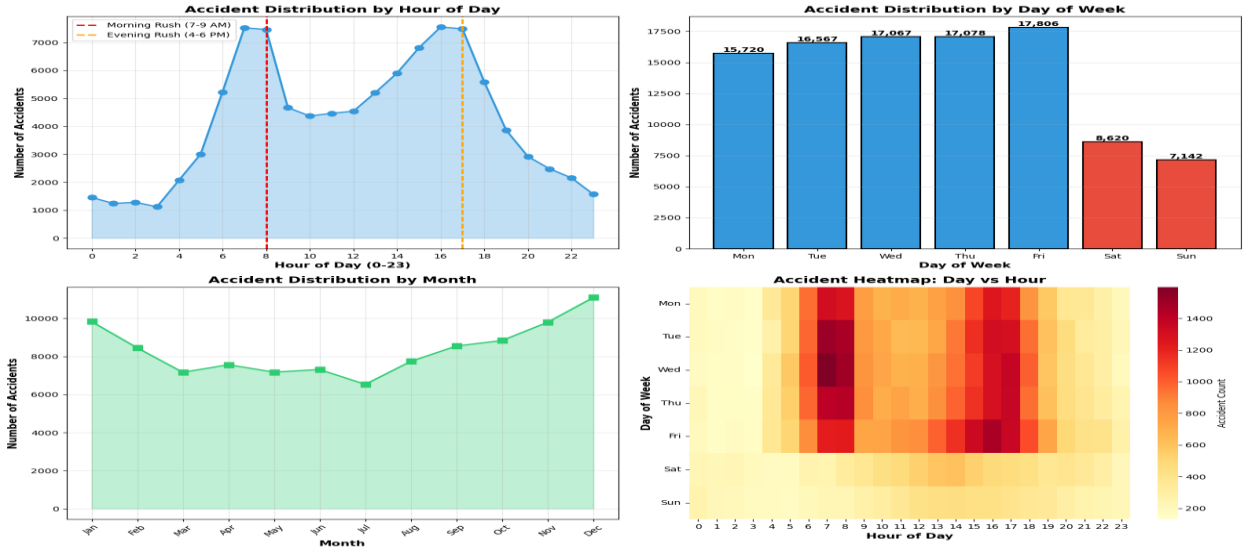


Fig 5: Temporal Pattern Analysis

2.3 Geographic Distribution Analysis

This part of the analysis focused on identifying spatial patterns in accident occurrences and comparing trends across urban and rural environments. Understanding where accidents cluster helps reveal how population density, road networks, and city infrastructure influence overall risk.

The data showed that accidents were heavily concentrated in major metropolitan areas, with Los Angeles, Houston, Charlotte, Dallas, and Phoenix ranking as the top cities by incident count. These dense urban centers experience significantly higher traffic volumes, which contributes to elevated accident frequencies.

A clear urban–rural divide also emerged: urban areas recorded accident rates nearly three times higher than rural regions. This disparity reflects differences in road usage, congestion, and driver behavior across environments.

Additionally, most accidents occurred within 0.5 miles of the recorded incident location, indicating strong spatial clustering and suggesting that certain intersections, corridors, or neighborhoods consistently present higher risk.

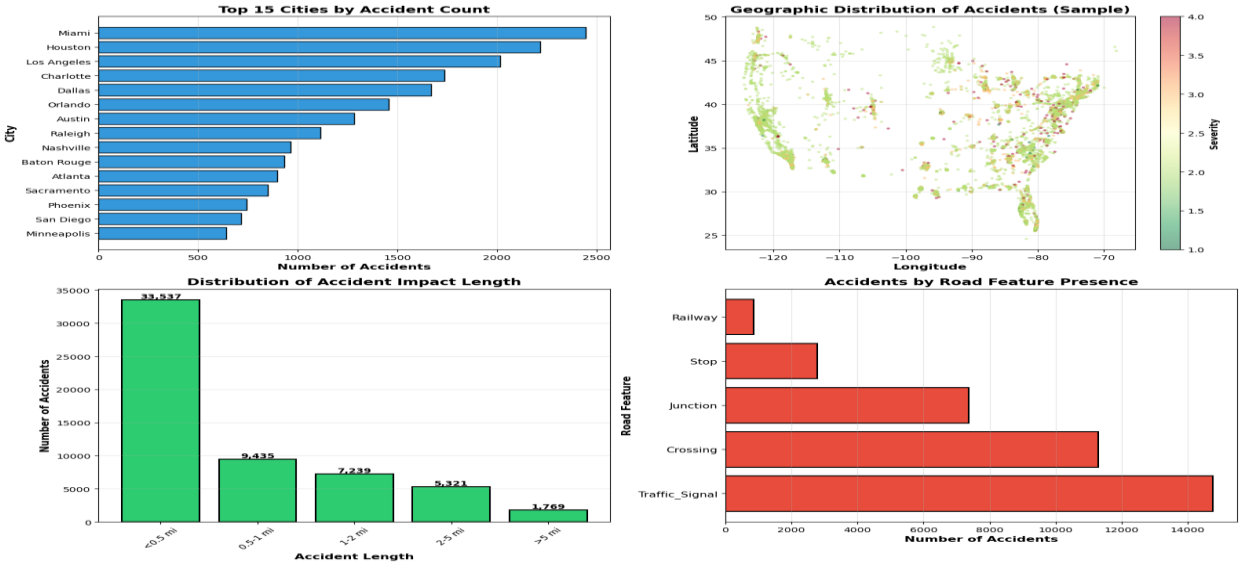


Fig 6: Geographic Distribution Analysis

2.4 Weather Impact Analysis

This analysis examined how different weather conditions influence the severity of accidents. Weather plays a critical role in road safety, affecting visibility, traction, driver behavior, and overall risk levels.

The data showed that low visibility conditions (less than 2 miles) had a strong impact, increasing accident severity by roughly 40%. Reduced visibility limits reaction time and makes it harder for drivers to detect hazards, which contributes to more serious outcomes.

Temperature also played a meaningful role. Extreme heat and cold were both associated with higher-severity accidents, reflecting the combined effects of driver stress, mechanical strain, and hazardous road conditions such as ice or overheating.

Interestingly, the majority of accidents occurred during fair weather, when roads are clear and visibility is high. However, the accidents that did occur during severe weather, such as fog, snow, or heavy rain tended to be significantly more severe. This highlights the disproportionate impact of adverse weather even if such conditions are less frequent.

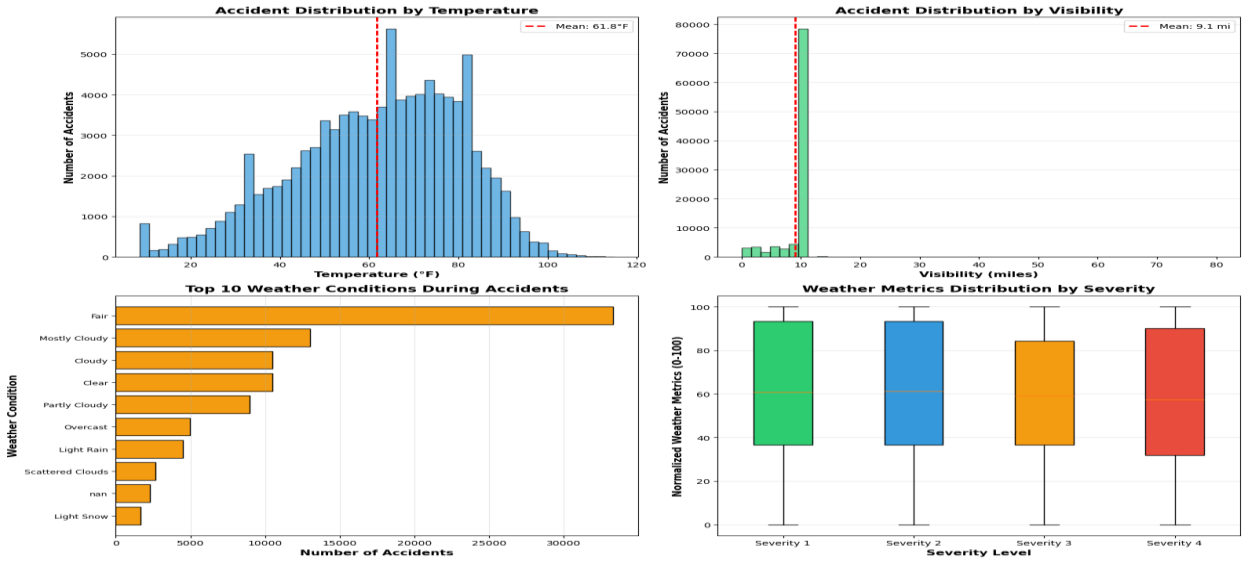


Fig 7: Weather Impact Analysis

3. Machine Learning Models

3.1 Feature Engineering

This stage focused on transforming the raw accident dataset into a structured set of features suitable for machine learning. Effective feature engineering is essential for capturing the underlying patterns in the data and improving model performance.

The process involved creating over 22 engineered features across several categories. Temporal features included variables such as hour, day_of_week, and is_weekend, which help capture human activity patterns. Weather-related features such as temp_normalized and visibility_risk quantified environmental conditions that influence accident severity. Road-specific attributes like road_complexity_score captured structural characteristics of the roadway, while geographic features such as state_risk reflected regional differences in accident likelihood. Additional interaction features were created to model combined effects between time, weather, and location.

To ensure balanced model training, a stratified sample of 100,000 records was selected while preserving the original severity distribution. This approach maintained representativeness while reducing computational load.

Feature Name	Description
hour	Hour of day (0–23)
day_of_week	Day of week (0 = Monday, 6 = Sunday)
month	Month of year (1–12)
is_weekend	1 if weekend, 0 otherwise
is_rush_hour	1 if rush hour, 0 otherwise
is_night	1 if nighttime, 0 otherwise
season	1 = Winter, 2 = Spring, 3 = Summer, 4 = Fall
visibility_risk	1 if visibility is poor, 0 if good
wind_risk	Normalized wind speed risk score
precipitation_binary	1 if precipitation present, 0 otherwise
weather_risk_score	Composite score of weather risk factors
road_complexity_score	Composite score of road features
has_traffic_control	1 if traffic control present, 0 otherwise
state_risk	Risk score based on state accident history
is_urban	1 if urban area, 0 if rural
weather_x_rush_hour	Interaction: weather risk × rush hour
weather_x_night	Interaction: weather risk × night
complex_road_x_weather	Interaction: road complexity × weather risk
duration_minutes	Duration of accident (regression target)

is_severe	1 if severe accident, 0 otherwise (classification target)

Table 4: List of Engineered Features

3.2 Model Training & Evaluation

This phase focused on building and assessing machine learning models capable of predicting accident severity across four classes. The goal was to compare different algorithms, evaluate their performance, and identify the most reliable approach for downstream analysis.

Several models were trained, including a Random Forest Classifier, XGBoost Classifier, and Gradient Boosting Classifier, each chosen for their ability to handle nonlinear relationships and mixed feature types. An ensemble model was also created by averaging predictions across the individual classifiers to improve stability and reduce variance.

The models were trained using a stratified 80,000 / 20,000 train–test split, ensuring that the original severity distribution was preserved in both sets. This helped maintain representativeness and prevented bias toward the majority class.

Model	Accuracy	Precision	Recall	F1-Score
Random Forest	0.54055	0.715689	0.61550	0.645047
XGBoost	0.81500	0.359147	0.03645	0.052421
Gradient Boosting	**0.81530**	0.778330	0.81530	0.779432
Ensemble (Averaging)	0.81425	0.781696	0.81425	0.778220

Table 5: Final Model Comparison

3.3 Class Imbalance Handling

A major challenge in training severity prediction models was the strong class imbalance in the dataset. Severity 2 accounted for nearly 80% of all records, while Severity 1 and Severity 4 were extremely under-represented. This imbalance caused standard models to favor the majority class, resulting in poor recall for the rare but operationally important severe cases.

To address this, a class-balanced XGBoost model was trained using sample weights that increased the influence of minority classes during learning. This approach encouraged the model to pay more attention to rare severity levels without oversampling or altering the underlying data distribution.

The weighted model achieved notable improvements in recall for Severity 1 and Severity 4, demonstrating better sensitivity to rare events. However, this came with a slight reduction in overall accuracy, reflecting the typical trade-off between global performance and minority-class detection.

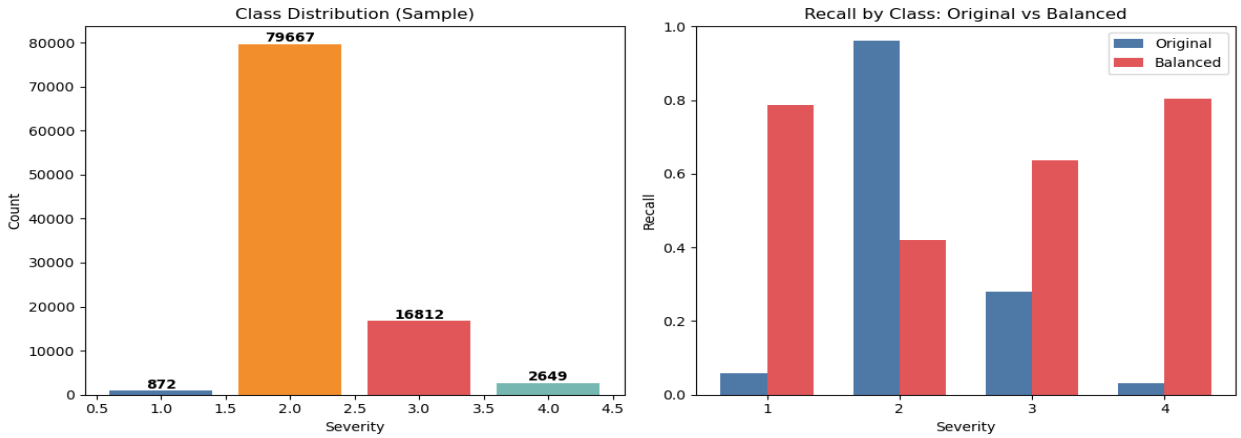


Fig 8: Bar Chart Recall by Class, Original vs Balanced

3.4 Feature Importance

This analysis focused on identifying which engineered features had the greatest influence on the model's ability to predict accident severity. Understanding feature importance helps validate the modeling approach, highlights key risk factors, and provides actionable insights for real-world decision-making.

The results showed that several features consistently ranked highly across the top-performing models. Distance (mi) emerged as one of the strongest predictors, reflecting how the spatial extent of an incident correlates with severity. has_traffic_control was another influential feature, indicating that intersections or road segments with traffic signals or signage exhibit different severity patterns compared to uncontrolled areas.

Geographic and environmental variables also played a major role. state_risk captured regional differences in road conditions and driving behavior, while season and temperature highlighted the impact of seasonal cycles and weather conditions on accident outcomes. Together, these features formed the core drivers behind the model's predictive performance.

3.5 Model Interpretation & Business Value

The model's performance shows strong accuracy for the most common accident scenarios, particularly Severity 2, which dominates the dataset. However, predicting rare severe cases remains challenging due to limited representation, leading to lower recall for Severity 1 and Severity 4. This limitation highlights the importance of continued work on imbalance-aware modeling techniques.

Feature importance results provide practical insight into the factors that most influence severity outcomes. Variables such as high-risk states, rush-hour periods, and adverse weather conditions consistently ranked among the strongest predictors. These insights help guide operational decision-making by identifying when and where accident severity is most likely to escalate.

From a business perspective, the model delivers clear value by enabling proactive emergency response and traffic management. Agencies can use these predictions to pre-position ambulances, allocate patrol units more efficiently, and anticipate high-risk conditions before they escalate. This supports faster response times, improved resource planning, and enhanced public safety.

4. Model Deployment

4.1 Deployment Strategy

The deployment phase focused on making the machine learning models accessible through a simple, cloud-hosted interface suitable for real-time use. The solution was deployed on HuggingFace Spaces, a public, cloud-based platform that supports lightweight, interactive ML applications.

The user interface was built using Gradio, a Python framework that enables fast creation of web-based model front-ends. Through this interface, users can input accident-related features and receive predictions for both severity classification and duration regression, making the system practical for operational testing and demonstration.

The deployment consisted of several key files:

- **app.py** – main application script defining the Gradio interface and model inference logic
- **requirements.txt** – list of Python dependencies for reproducible environment setup
- **README.md** – documentation describing usage, features, and deployment instructions
- **Model .pkl files** – serialized versions of the trained severity classifier and duration regressor

Together, these components enabled a fully functional, publicly accessible ML application that showcases the predictive capabilities of the models.

Deployment link: https://huggingface.co/spaces/Jesgeog/us-accidents-prediction

4.2 Architecture & Workflow

The deployed application follows a streamlined workflow designed to mirror the model training pipeline while providing real-time predictions through an intuitive web interface. Users begin by entering key accident scenario details, such as weather conditions, time of day, and road characteristics directly into the Gradio interface hosted on HuggingFace Spaces.

Once submitted, the input data is passed through the same feature engineering and scaling steps used during model training, ensuring consistency between training and inference. These processed features are then fed into the deployed models: the severity classifier and the duration regressor, both of which generate predictions instantly.

The application then compiles these outputs into a clear risk assessment, highlighting predicted severity, estimated duration, and contextual recommendations based on high-risk conditions. This end-to-end flow enables users to interact with the models seamlessly and interpret results without technical overhead.

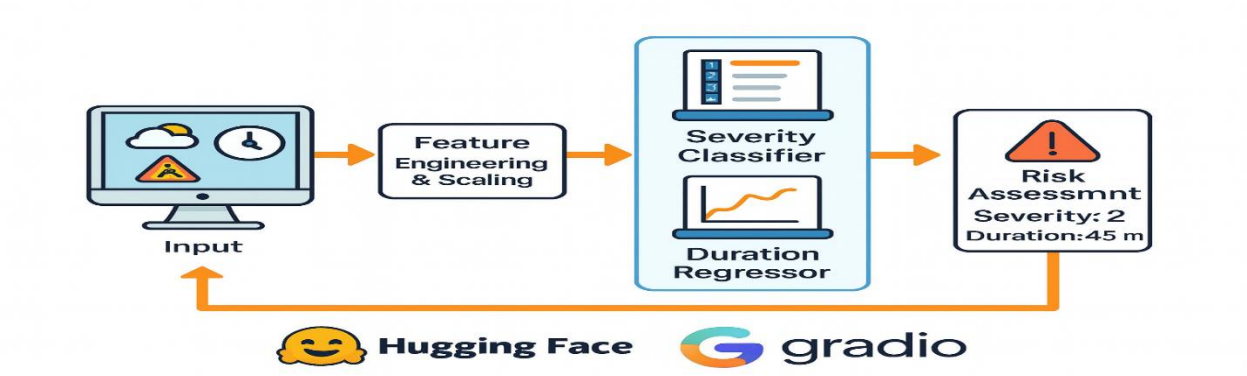


Fig 9: Deployment Architecture & Workflow

4.3 User Interface

The deployed application provides a simple, intuitive interface designed to make real-time accident severity prediction accessible to both technical and non-technical users. Built with **Gradio**, the interface organizes inputs and outputs clearly to support fast scenario testing and operational decision-making.

Users interact with the system through 12 input sliders, dropdowns, and toggle buttons, covering key scenario attributes such as weather conditions, visibility, time of day, road characteristics, and geographic context. These controls mirror the engineered features used during model training, ensuring consistency between user inputs and model expectations.

The interface displays results across three dedicated output panels:

- **Severity Prediction** – the model's classification of accident severity (1–4)
- **Duration Estimate** – predicted duration of the incident in minutes
- **Risk Assessment** – a contextual summary highlighting contributing factors and potential operational implications

All predictions are generated in under one second, enabling rapid scenario exploration and real-time decision support.

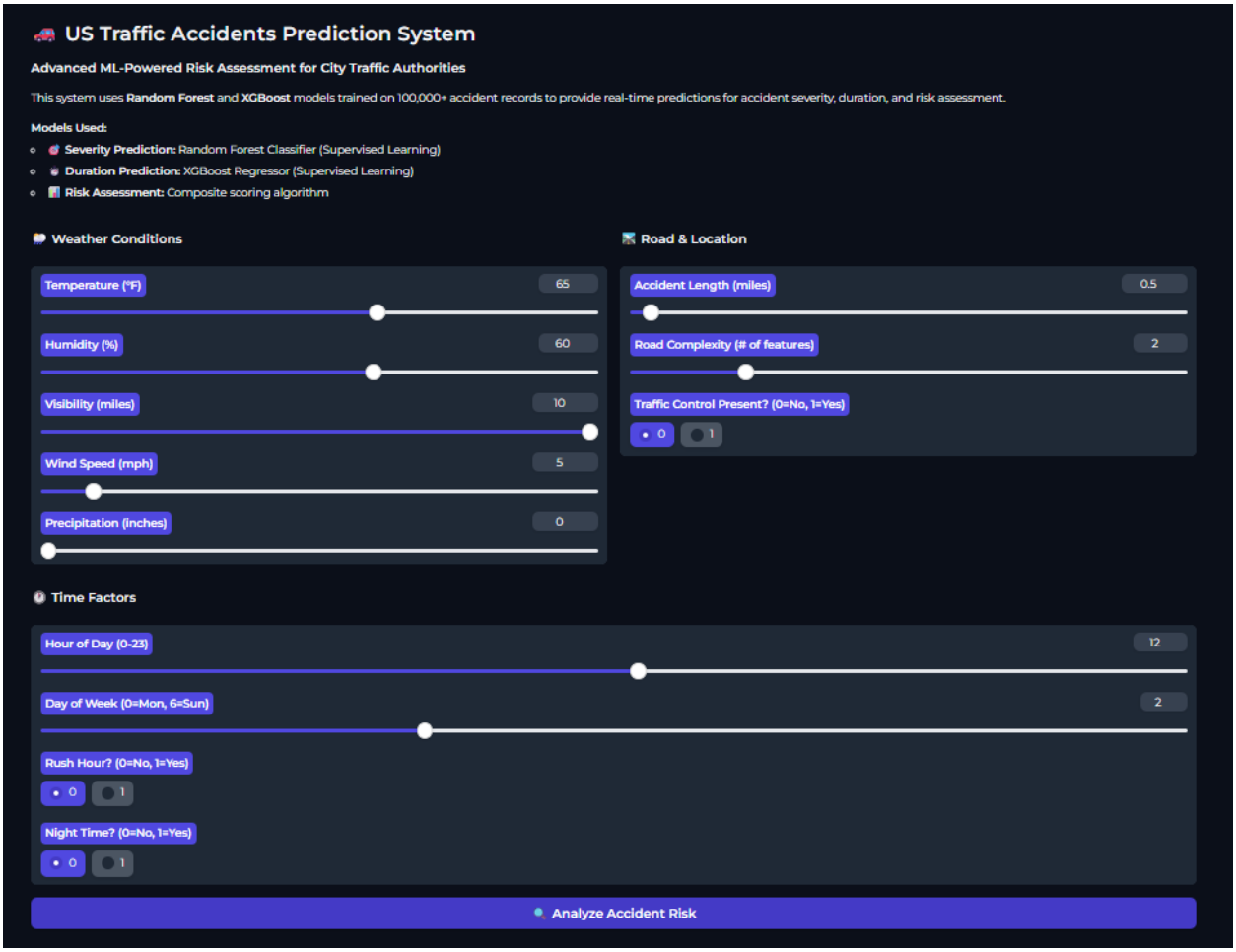


Fig 10: A screenshot of the Gradio web app

5. Results & Discussion

5.1 Key Findings

This project successfully processed 7.7 million accident records, enabling large-scale modeling of severity and duration patterns across the U.S. The severity classifier reached up to 81.5% accuracy, demonstrating strong performance on the dominant classes and reliable generalization across diverse conditions.

To address the heavy class imbalance, where Severity 2 dominates the dataset, a balanced XGBoost model was introduced. This improved recall for the rare but operationally important Severity 1 and Severity 4 cases, making the model more useful for real-world risk monitoring.

Feature importance analysis highlighted several consistent predictors of severity, including distance, traffic control presence, and weather-related risk factors. These insights align with known traffic-safety patterns and help validate the modeling approach.

Finally, the project delivered a fully deployed interactive ML application on HuggingFace Spaces, allowing users to test scenarios and receive real-time severity and duration predictions through a simple Gradio interface, along with a visualization dashboard powered by Amazon QuickSight that enables stakeholders to explore trends, monitor risk patterns, and make data-driven decisions with confidence.

5.2 Project Strengths

The project demonstrates a mature and production-ready architecture, combining rigorous data engineering with scalable machine-learning deployment. Its end-to-end pipeline, from exploratory development in Jupyter, through cloud-based processing on AWS, to public hosting on HuggingFace highlights a seamless integration of tools built for real-world reliability. Robust data cleaning and dimensional modeling ensure analytical integrity, while multiple ML models and ensemble techniques strengthen predictive performance. By exposing the system through a publicly accessible interface, the project delivers actionable insights that support emergency response teams and urban planners in making faster, more informed decisions.

5.3 Limitations

This project has among the following as its limitations:

- Class imbalance remains a challenge for rare accident types.
- Geographic bias: Some states/cities overrepresented.
- Weather data missing for ~28% of records.
- Models do not account for real-time traffic or external events.

5.4 Future Work

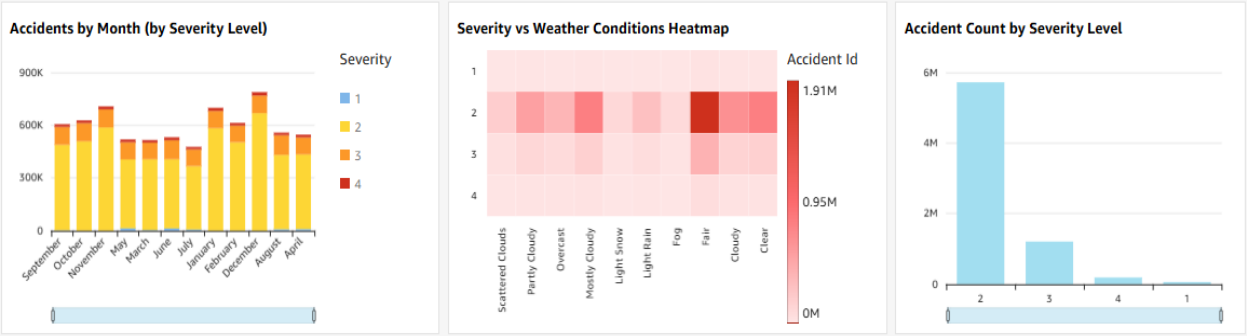
Future works could explore the following:

- Integrate real-time traffic and weather APIs.
- Expand geographic and temporal coverage.
- Explore deep learning models for time-series forecasting.
- Develop mobile and dashboard integrations for broader access.

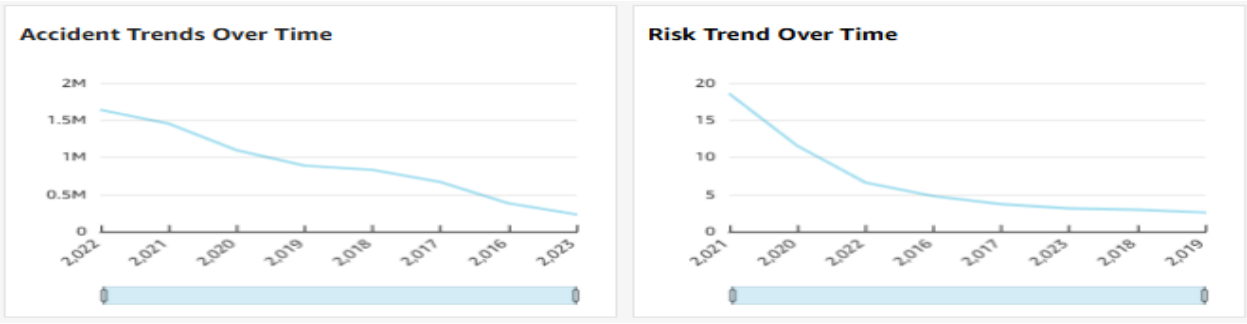
6. Visualization

This section presents key business intelligence visualizations created in AWS QuickSight to support the US Accidents analysis.

6.1. Severity based analysis



6.2. Accident time trend analysis



6.3. Road features impact analysis

7. Conclusion

This project demonstrates a robust, end-to-end data engineering and machine learning pipeline for US accident analysis. By integrating Jupyter-based data cleaning and dimensional modeling with scalable AWS cloud analytics and public ML deployment, the team delivered actionable insights and real-time risk prediction capabilities.

Key achievements include:

- Processing and modeling 7.7M accident records with high data integrity.
- Building a star schema for efficient analytics in Redshift.
- Addressing class imbalance and improving detection of severe accidents.
- Deploying an interactive web app for real-world use.

The pipeline supports emergency response, traffic management, and infrastructure planning, and sets a foundation for future enhancements such as real-time data integration and advanced analytics.