🚀 Comprehensive Project Report: Car Insurance Claim Dashboard

---

🧭 1. Project Overview

Goal:
To build a complete, automated, and interactive **data analysis and visualization pipeline** for car insurance claim prediction and insights.

The project covers:

- Data loading and cleaning
- Logging and report generation
- Summary metrics and analysis
- Static and interactive visualization
- A fully interactive **Dash web dashboard**

This project helps explore **claim patterns** across customer demographics and vehicle attributes, revealing insights like:

- Claim likelihood by age, gender, and driving experience
- Relationships between vehicle factors and claim outcomes
- Key performance indicators (KPIs) for decision-making

🧱 2. Project Structure

Below is your final directory layout:

```
C:\Users\user\Desktop\my_dash_board_project

|

|  Car_Insurance_Claim.csv        # Raw dataset

|  cleaned_data.csv               # Final cleaned dataset

|  data_cleaning.py               # Script for data cleaning & preprocessing

|  metrics_summary.py             # Script for calculating summary statistics

|  requirements.txt               # Python dependencies
```

```
|
├──data
|     app.py                    # Main dashboard application
|     cleaned_data.csv           # Copy of cleaned data (used by app)
|
├──logs
|     cleaning_log_*.log          # Logs from data cleaning
|     dashboard.log              # Logs from the dashboard runtime
|
├──outputs
|   ├──logs
|   |     data_summary.log        # Log from metrics summary step
|   |     data_visualization.log   # Log from visualization script
|   |     data_visualization.py    # Visualization script
|   |
|   ├──metrics                    # CSV outputs with grouped summary stats
|   |     claims_by_age.csv
|   |     claims_by_gender.csv
|   |     claims_by_driving_experience.csv
|   |     claims_by_vehicle_ownership.csv
|   |     claims_by_vehicle_year.csv
|   |     summary_metrics.csv
|   |     violations_by_claim_status.csv
|   |     violations_by_outcome.csv
|   |
|   └──visualizations
|       ├──interactive          # HTML-based dynamic visualizations
|      └──static                # PNG plots (static graphs)
|
 ├──reports
|     data_cleaning_report.csv     # Cleaning summary report
|     data_integrity_summary.csv   # Validation & consistency report
|
└──visuals
      distribution_boxplots.png    # Outlier & spread visualization
      missing_values_heatmap.png   # Missing data map
```

⚙️ 3. Step-by-Step Workflow

STEP 1 — Data Cleaning (data_cleaning.py)

Objective:
Prepare and sanitize the raw dataset Car_Insurance_Claim.csv to produce a clean, analysis-ready dataset.

Tasks Performed:

- Handled **missing values**
- Converted **data types** (e.g., categorical and numeric columns)
- Fixed **inconsistent labels**
- Removed **duplicates**
- Created new **derived columns** (like age bins or risk factors)
- Exported clean data as cleaned_data.csv

Outputs:

- ✅ cleaned_data.csv (in project root & /data/)
- 🧾 Logs: logs/cleaning_log_*.log

- 📊 Reports:
    - reports/data_cleaning_report.csv
    - reports/data_integrity_summary.csv

Issues Solved:

- **Type mismatch errors** (some columns read as object instead of int/float)
  → Solution: used pd.to_numeric(errors='coerce') and filled missing values.
- **Column naming inconsistencies** (e.g., Vehicle_Ownership vs vehicle_ownership)
  → Standardized using df.columns = df.columns.str.lower().

STEP 2 — Metrics & Summary (metrics_summary.py)

Objective:
Generate core descriptive statistics and aggregated summaries for further visualization.

Metrics Computed:

- Claims by **age**, **gender**, **driving experience**, **vehicle year**, **ownership**
- Summary metrics (averages, claim rate, totals)
- Violations grouped by **claim status**

Outputs:
All CSV summary files are stored under:
📁 outputs/metrics/

Example Files:

- claims_by_age.csv
- claims_by_gender.csv
- summary_metrics.csv
- violations_by_outcome.csv

Logs:
🪵 outputs/logs/data_summary.log

Common Issue:

- ValueError: No numeric types to aggregate
  → Fixed by ensuring grouping columns were categorical and numeric columns were properly converted before aggregation.

STEP 3 — Visualization (data_visualization.py)

Objective:
Create static (PNG) and interactive (HTML) plots to explore data patterns visually.

Tools Used:

- matplotlib & seaborn → static PNG plots
- plotly.express → interactive HTML visualizations

Key Visuals:

- Claim rate by **age**, **gender**, **vehicle year**
- Correlation heatmap
- Speeding violations vs claim rate
- Education vs claim likelihood

Outputs:

- 🖼️ **Static Plots:** outputs/visualizations/static/
- 🌐 **Interactive Plots:** outputs/visualizations/interactive/
- 🪵 **Logs:** outputs/logs/data_visualization.log

Issues Faced:

- ❌ *"agg function failed [how->mean,dtype->object]"*
  → Fixed by converting mixed-type columns to numeric before aggregation using:
- df[column] = pd.to_numeric(df[column], errors="coerce")
- ⚠️ *FutureWarning about seaborn palette usage*
  → Harmless; can safely be ignored or resolved by assigning x variable to hue.

STEP 4 — Interactive Dashboard (data/app.py)

Objective:
Create a live dashboard to explore and visualize filtered insurance claim data dynamically.

Framework Used:
Dash (Plotly) + dash-bootstrap-components for UI styling.

Core Components:

- **Filters:** Dropdowns for age, gender, driving experience, vehicle year
- **KPI Cards:** Show key metrics (Total Customers, Claim Rate, etc.)
- **Graphs:** Interactive bar and box plots for claim trends

Callback Mechanism:
When filters change, a callback function dynamically:

- Filters the dataset
- Recalculates KPIs
- Updates all graphs in real time

Metrics Displayed:

- Total Customers
- Total Claims
- Claim Rate (%)
- Average Credit Score
- Average Annual Mileage

Outputs:

- 🌐 Dashboard runs locally at:
  👉 http://127.0.0.1:8050
- 🪵 Logs dashboard events in:
  logs/dashboard.log

Common Issues Fixed:

- Data not updating on filter → corrected callback filtering logic
- Plotly errors for non-numeric columns → enforced numeric casting

- Duplicate cleaned data paths → standardized loading from root cleaned_data.csv

---

🧮 5. Data Flow Summary

Step	Script	Input	Process	Output	Output Location
1	data_cleaning.py	Car_Insurance_Claim.csv	Cleaning & validation	cleaned_data.csv	/ & /data/
2	metrics_summary.py	cleaned_data.csv	Summary stats computation	CSV summaries	/outputs/metrics/
3	data_visualization.py	cleaned_data.csv	Generate visualizations	PNG + HTML plots	/outputs/visualizations/
4	app.py	cleaned_data.csv	Interactive dashboard	Live web app	http://127.0.0.1:8050

🧰 6. Tools & Libraries Used

Category	Libraries
Data Processing	pandas, numpy
Visualization	matplotlib, seaborn, plotly
Dashboarding	dash, dash-bootstrap-components
Logging	logging, os, datetime
File Handling	csv, os, pathlib

🧾 7. Logs & Reporting System

Logging was integrated throughout all scripts for full transparency.

Log File	Purpose
logs/cleaning_log_*.log	Tracks data cleaning steps and errors
outputs/logs/data_summary.log	Records summary generation progress
outputs/logs/data_visualization.log	Records visualization creation
logs/dashboard.log	Monitors live dashboard events

Reports generated:

- reports/data_cleaning_report.csv — detailed cleaning actions
- reports/data_integrity_summary.csv — quality validation checks