Interim Report: AlphaCare Insurance Risk Analytics

Author: Eyoab Amare
Date: June 15, 2025
Repository: https://github.com/Eyoab11/Insurance-Risk-Analytics

Project Overview

This project analyzes historical car insurance data for AlphaCare Insurance Solutions (ACIS) to identify key risk drivers and uncover actionable insights. The primary objectives are to optimize marketing strategies by targeting low-risk customer segments and to lay the groundwork for a data-driven, predictive pricing model. This report covers the foundational work of data exploration and establishing a reproducible data pipeline.

Task 1: Project Setup & Exploratory Data Analysis (EDA)

1. Implementation Summary

- Project Scaffolding: Established a clean project structure with separate directories for data, notebooks, scripts, and plots.
- Data Conversion: Developed a Python script to parse the raw pipe-delimited (|) text file and convert it into a standard, analysis-ready CSV format.
- Initial Analysis: Performed a comprehensive EDA in a Jupyter Notebook, which included:
  - Calculating descriptive statistics for numerical features.
  - Assessing data quality and quantifying missing values.
  - Analyzing variable distributions (univariate analysis).
  - Investigating relationships between key features like premium, claims, and vehicle value (bivariate analysis).
- Visualization: Generated over 8 plots to capture key insights, including three main visualizations on geographic risk, claim severity, and temporal trends.

2. Key Metrics

Metric	Value
Total Records Analyzed	~290,000
Total Features	53
Visualizations Generated	8+
Highest Missing Data (Column)	CapitalOutstanding (over 90%)

3. Outputs

- **Data Conversion Script:** scripts/convert_to_csv.py
- **Cleaned Dataset:** data/insurance_data.csv
- **Analysis Notebook:** notebooks/task_1_eda.ipynb
- **Generated Plots:** All saved in the plots/ directory.

3. Challenges & Solutions

Challenge	Solution
Raw data was in a non-standard, pipe-delimited format.	Created scripts/convert_to_csv.py to convert it to a standard CSV.
Inconsistent column capitalization (e.g., 'make' vs. 'Make').	Used df.columns to inspect names and standardized usage in the code.
Unclear date format in 'TransactionMonth'.	Diagnosed the format using df['col'].unique() and applied the correct pd.to_datetime method.

Task 2: Data Version Control (DVC)

1. Implementation Summary

- DVC Initialization: Successfully initialized DVC in the project repository to enable data versioning separate from code.
- Remote Storage Setup: Configured a local remote storage directory outside the project folder to act as the central repository for data files.
- Data Tracking: Used dvc add to place the cleaned insurance_data.csv under DVC's control, replacing the large data file with a small pointer file in Git.
- Configuration: Modified the .gitignore file to correctly ignore large data files while ensuring the small .dvc pointer files are tracked by Git.
- Data Push: Pushed the versioned dataset to the configured local remote, completing the versioning cycle.

2. Key Metrics

Metric	Value
Datasets Tracked	1 - insurance_data.csv
DVC Remote Type	Local File System
Data File Size	~85 MB
.dvc Pointer File Size	<1kb

3. Outputs

- DVC Pointer File: data/insurance_data.csv.dvc
- DVC Configuration: .dvc/config
- Updated Git Ignore: .gitignore with rules for DVC.

3. Challenges & Solutions

Challenge	Solution
dvc add failed due to .gitignore ignoring all files in the data/ directory.	Modified .gitignore to use !/data/*.dvc to explicitly un-ignore the DVC pointer files.

Next Steps

- Hypothesis Testing: Proceed with Task 3 to statistically validate or reject the initial hypotheses formed during the EDA (e.g., risk differences across provinces and genders).
- Feature Engineering: Begin creating new features based on EDA insights to prepare the data for modeling.
- Modeling: Start building the first predictive models to estimate claim severity.

Tools Used\

- Python 3.13
- Pandas & NumPy
- Matplotlib & Seaborn
- Git & GitHub
- DVC (Data Version Control)