B5W3: End-to-End Insurance Risk Analytics & Predictive Modeling

Dive into real insurance data to uncover low-risk segments and build smart models that optimize premiums.

BUSINESS OBJECTIVE

Your employer **AlphaCare Insurance Solutions (ACIS)** is committed to developing cutting-edge risk and predictive analytics in the area of car insurance planning and marketing in South Africa. You have recently joined the data analytics team as marketing analytics engineer, and your first project is to analyse historical insurance claim data. The objective of your analyses is to help optimise the marketing strategy as well as discover “low-risk” targets for which the premium could be reduced, hence an opportunity to attract new clients.

In order to deliver the business objectives, you would need to brush up your knowledge and perform analysis in the following areas:

- **Insurance Terminologies**
  - Read on how insurance works. Check out the key insurance glossary 50 Common Insurance Terms and What They Mean — Cornerstone Insurance Brokers
- **A/B Hypothesis Testing**
  - Read on the benefits of A/B hypothesis testing
  - Accept or reject the following null hypothesis
    - There are no risk differences across provinces
    - There are no risk differences between zipcodes
    - There are no significant margin (profit) difference between zip codes
    - There are not significant risk difference between Women and men
- **Machine Learning & Statistical Modeling**
  - For each zipcode, fit a linear regression model that predicts the total claims
  - Develop a machine learning model that predicts optimal premium values given
    - Sets of features about the car to be insured
    - Sets of features about the owner
    - Sets of features about the location of the owner
    - Other features you find relevant
  - Report on the explaining power of the important features that influence your model

Your final report should detail the methodologies used, present the findings from your analysis, and make recommendations on plan features that could be modified or enhanced based on the test results. This will help AlphaCare Insurance Solutions to tailor their insurance products more effectively to meet consumer needs and preferences.

MOTIVATION

The challenge will sharpen your skills in Data Engineering (DE), Predictive Analytics (PA), and Machine Learning Engineering (MLE).

The tasks are designed to improve your ability to manage complex datasets, adapt to challenges, and think creatively, skills that are essential in insurance analytics. This analysis will help you understand more about how hypothesis testing and predictive analytics can be applied in insurance analysis.

Engage with as many tasks as possible. The volume and complexity of the tasks are designed to simulate the pressures and deadlines typical in the financial analytics field.

DATA

The historical data is from Feb 2014 to Aug 2015, and it can be found here

The structure of the data is as follows

- Columns about the insurance policy

UnderwrittenCoverID
PolicyID

- The transaction date
  TransactionMonth
- Columns about the client

IsVATRegistered
Citizenship
LegalType
Title
Language
Bank
AccountType
MaritalStatus
Gender

- Columns about the client location
  Country
  Province
  PostalCode
  MainCrestaZone
  SubCrestaZone
- Columns about the car insured
  ItemType
  Mmcode
  VehicleType
  RegistrationYear
  Make
  Model
  Cylinders
  Cubiccapacity
  Kilowatts
  Bodytype
  NumberOfDoors
  VehicleIntroDate
  CustomValueEstimate
  AlarmImmobiliser
  TrackingDevice
  CapitalOutstanding
  NewVehicle

WrittenOff
Rebuilt
Converted
CrossBorder
NumberOfVehiclesInFleet

- Columns about the plan
  SumInsured
  TermFrequency
  CalculatedPremiumPerTerm
  ExcessSelected
  CoverCategory
  CoverType
  CoverGroup
  Section
  Product
  StatutoryClass
  StatutoryRiskType

Columns about the payment & claim
TotalPremium
TotalClaims

LEARNING OUTCOMES

- Understanding the data provided and extracting insight. You will have to explore different techniques, algorithms, statistical distributions, sampling, and visualization techniques to gain insight.
- Understand the data structure and algorithms used in EDA and machine learning pipelines.
- Modular and object-oriented Python code writing. Python package building.
- Statistical Modeling and Analysis. You will have to use statistical models to predict and analyze the outcomes of A/B tests, applying techniques such as logistic regression, or chi-squared tests, as appropriate to the hypotheses being tested.
- A/B Testing Design and Implementation. You will design robust A/B tests that can yield clear, actionable results. This includes determining the sample size, selecting control and test groups, and defining success metrics.
- Data Versioning. You will manage and document versions of datasets and analysis results.

TEAM

Facilitator:

- Mahlet
- Kerod
- Rediet
- Rehmet

KEY DATES

- **Challenge Introduction** - 9:30 AM UTC time on Wednesday 11 June 2025.
- **Interim Submission - 8:00 PM UTC time on Friday 13 June 2025.**
- **Final Submission** - 8:00 PM UTC time on Tuesday 17 June 2025.

DELIVERABLES AND TASKS TO BE DONE

Task 1:

1.1 Git and GitHub

- Tasks:
  - Create a git repository for the week with a good Readme
  - Git version control
  - CI/CD with Github Actions
- Key Performance Indicators (KPIs):
  - Dev Environment Setup.
  - Relevant skill in the area demonstrated.

1.2 Project Planning - EDA & Stats

- Develop a foundational understanding of the data, assess its quality, and uncover initial patterns in risk and profitability
- **Tasks**:
  - Data Understanding
  - Exploratory Data Analysis (EDA)
  - Guiding Questions:
    - What is the overall Loss Ratio (TotalClaims / TotalPremium) for the portfolio? How does it vary by Province, VehicleType, and Gender?
    - What are the distributions of key financial variables? Are there outliers in TotalClaims or CustomValueEstimate that could skew our analysis?
    - Are there temporal trends? Did the claim frequency or severity change over the 18-month period?
    - Which vehicle makes/models are associated with the highest and lowest claim amounts?
  - Statistical thinking
- **KPIs**:
  - Proactivity to self-learn - sharing references.
  - EDA techniques to understand data and discover insights,
  - Demonstrating Stats understanding by using suitable statistical distributions and plots to provide evidence for actionable insights gained from EDA.

Minimum Essential To Do

- Create a github repository that you will be using to host all the code for this week.
- Create at least one new branch called ”task-1” for your analysis of day 1
- Commit your work at least three times a day with a descriptive commit message
- Perform Exploratory Data Analysis (EDA) analysis on the following:
  - **Data Summarization**:
    - Descriptive Statistics: Calculate the variability for numerical features such as TotalPremium, TotalClaim, etc.
    - Data Structure: Review the dtype of each column to confirm if categorical variables, dates, etc. are properly formatted.
  - **Data Quality Assessment**:
    - Check for missing values.
  - **Univariate Analysis**:
    - Distribution of Variables: Plot histograms for numerical columns and bar charts for categorical columns to understand distributions..
  - **Bivariate or Multivariate Analysis:**
    - Correlations and Associations: Explore relationships between the monthly changes TotalPremium and TotalClaims as a function of ZipCode, using scatter plots and correlation matrices.
  - **Data Comparison**
    - **Trends Over Geography: Compare the change in insurance cover type, premium, auto make, etc.**

- **Outlier Detection:**
  - Use box plots to detect outliers in numerical data
- **Visualization**
  - Produce 3 creative and beautiful plots that capture the key insight you gained from your EDA

Task 2:

Establish a reproducible and auditable data pipeline using Data Version Control (DVC), a standard practice in regulated industries.

In finance and insurance, we must be able to reproduce any analysis or model result at any time for auditing, regulatory compliance, or debugging. DVC ensures our data inputs are as rigorously version-controlled as our code.

Data Version Control (DVC)

- Tasks:
  - Install DVC
    - pip install dvc
  - Initialize DVC: In your project directory, initialize DVC
    - dvc init
  - Set Up Local Remote Storage
    - Create a Storage Directory
      - mkdir /path/to/your/local/storage
    - Add the Storage as a DVC Remote
      - dvc remote add -d localstorage /path/to/your/local/storage
  - Add Your Data:
    - Place your datasets into your project directory and use DVC to track them
      - dvc add <data.csv>
  - Commit Changes to Version Control
    - Create different versions of the data.
      -
    - Commit the .dvc files (which include information about your data files and their versions) to your Git repository
  - Push Data to Local Remote
    - dvc push

Minimum Essential To Do:

- Merge the necessary branches from task-1 into the main branch using a Pull Request (PR)
- Create at least one new branch called "task-2"
- Commit your work with a descriptive commit message.
- Install DVC
- Configure local remote storage
- Add your data
- Commit Changes to Version Control
- Push Data to Local Remote

Task 3:

Statistically validate or reject key hypotheses about risk drivers, which will form the basis of our new segmentation strategy.

A/B Hypothesis Testing

For this analysis, "risk" will be quantified by two metrics: Claim Frequency (proportion of policies with at least one claim) and Claim Severity (the average amount of a claim, given a claim occurred). "Margin" is defined as (TotalPremium - TotalClaims).

- Accept or reject the following **Null Hypotheses:**

1. **H_0:**There are no risk differences across provinces
2. **H_0:**There are no risk differences between zip codes
3. **H_0:**There are no significant margin (profit) difference between zip codes
4. **H_0:**There are not significant risk difference between Women and Men

- Tasks:
    - Select Metrics
        - Choose the key performance indicator (KPI) that will measure the impact of the features being tested.
    - Data Segmentation
        - **Group A (Control Group)**: Plans without the feature
        - **Group B (Test Group)**: Plans with the feature.
        - For features with more than two classes, you may need to select two categories to split the data as Group A and Group B. You must ensure, however, that the two groups you selected do not have significant statistical differences on anything other than the feature you are testing. For example, the client attributes, the auto property, and insurance plan type are statistically equivalent.
    - Statistical Testing
        - Conduct appropriate tests such as chi-squared for categorical data or t-tests or z-test for numerical data to evaluate the impact of these features.
        - Analyze the p-value from the statistical test:
            - If p_value < 0.05 (typical threshold for significance), reject the null hypothesis. This suggests that the feature tested does have a statistically significant effect on the KPI.
            - If p_value >= 0.05, fail to reject the null hypothesis, suggesting that the feature does not have a significant impact on the KPI.
    - Analyze and Report
        - Analyze the statistical outcomes to determine if there's evidence to reject the null hypotheses. Document all findings and interpret the results within the context of their impact on business strategy and customer experience.

Minimum Essential To Do:

- Merge the necessary branches from task-2 into the main branch using a Pull Request (PR)
- Create at least one new branch called "task-3"
- Commit your work with a descriptive commit message.
- Select Metrics
- Data Segmentation
- Statistical Testing
- Analyze and Report
- Execution of the tests.
- **Interpretation & Business Recommendation:** For each rejected hypothesis, provide a clear interpretation of the result in business terms. E.g., We reject the null hypothesis for provinces (p < 0.01). Specifically, Gauteng exhibits a 15% higher loss ratio than the Western Cape, suggesting a regional risk adjustment to our premiums may be warranted.

Task 4:

Build and evaluate predictive models that form the core of a dynamic, risk-based pricing system.

Modeling Goals:

1. Claim Severity Prediction (Risk Model): For policies that have a claim, build a model to predict the TotalClaims amount. This model estimates the financial liability associated with a policy.

Target Variable: TotalClaims (on the subset of data where claims > 0).

Evaluation Metric: Root Mean Squared Error (RMSE) to penalize large prediction errors, and R-squared.

1. Premium Optimization (Pricing Framework): Develop a machine learning model to predict an appropriate premium. A naive approach is to predict CalculatedPremiumPerTerm, but a more sophisticated, business-driven approach is required.

Advanced Task: Build a model to predict the probability of a claim occurring (a binary classification problem). The Risk-Based Premium can then be conceptually framed as: Premium = (Predicted Probability of Claim * Predicted Claim Severity) + Expense Loading + Profit Margin.

Statistical Modeling

- Tasks:
    - Data Preparation:
        - Handling Missing Data: Impute or remove missing values based on their nature and the quantity missing.
        - Feature Engineering: Create new features that might be relevant to TotalPremium and TotalClaims.
        - Encoding Categorical Data: Convert categorical data into a numeric format using one-hot encoding or label encoding to make it suitable for modeling.
        - Train-Test Split: Divide the data into a training set (for building the model) and a test set (for validating the model), typically using a 70:30 or 80:20 ratio.
    - Modeling Techniques
        - **Linear Regression**
        - Decision Trees
        - Random Forests
        - **Gradient Boosting Machines (GBMs):**
            - **XGBoost**
    - Model Building
        - Implement Linear Regression, Random Forests, and XGBoost models
    - Model Evaluation
        - Evaluate each model using appropriate metrics like accuracy, precision, recall, and F1-score.
    - Feature Importance Analysis
        - Analyze which features are most influential in predicting retention.
    - Use SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) to interpret the model's predictions and understand how individual features influence the outcomes.
    - Report comparison between each model performance.

Minimum Essential To Do:

- Merge the necessary branches from task-3 into the main branch using a Pull Request (PR)
- Create at least one new branch called "task-4".
- Commit your work with a descriptive commit message.
- Data preparation
- Model building
- Model evaluation
- A rigorous model evaluation section comparing models on appropriate metrics.
- Model Interpretability: Use SHAP or LIME to identify the top 5-10 most influential features for your best-performing model. Your report must explain how these features impact the prediction and what that means for the business. E.g.,

"SHAP analysis reveals that for every year older a vehicle is, the predicted claim amount increases by X Rand, holding other factors constant. This provides quantitative evidence to refine our age-based premium adjustments."

DUE DATE (SUBMISSION)

Interim Submission Sunday (15 June 2025): 8:00 PM (UTC)

- GitHub link to your main branch, showing merged work from task-1 and task-2.
- Interim report - Covering task-1 and task-2 summarizing your EDA findings and DVC setup.

Final Submission Tuesday (17 June, 2025): 8:00 PM (UTC)

- GitHub Link to your main branch
- A polished **final report** in the format of a Medium blog post. This should be a self-contained, professional artifact that includes:
  - A brief, non-technical overview of the project for leadership.
  - A clear description of your analytical approach.
  - The most important insights from your EDA, hypothesis testing, and modeling.
  - Concrete, data-backed suggestions for ACIS's marketing and pricing strategy.
  - Acknowledgment of data/model limitations and suggestions for future work.

Feedback

You may not receive detailed comments on your interim submission but will receive a grade.

Other Considerations:

- **Documentation:** Encourage detailed documentation in code and report writing.
- **Collaboration:** Emphasise collaboration through Github issues and projects.
- **Communication**: Regular check-ins, Q&A sessions, and a supportive community atmosphere.
- **Flexibility:** Acknowledge potential challenges and encourage proactive communication.
- **Professionalism:** Emphasise work ethics and professional behavior.

Time Management: Stress the importance of punctuality and managing time effectively.

TUTORIALS SCHEDULE

In the following, the **Bold** indicates morning sessions, and *Italic* indicates afternoon sessions.

- Day 1 Wednesday (11 June 2025): `
  - **Introduction to the Challenge (Mahlet)**
  - Introduction to Insurance Analytics (Rediet)
- Day 2 Thursday (12 June 2025 :
  - **Statistical distributions, hypothesis testing, and creating actionable insights (Kerod)**
  - Data Version Control (DVC) (Rehmet)
- Day 3 Friday (13 June 2025) :
  - **Statistical Modeling and Evaluation (Rediet)**
  - Introduction to model interpretability (Kerod)
- Day 4 Monday (16 June 2025) :
  - Q&A (Rehmet and Mahlet)

REFERENCES

- **Insurance Analytics**
  - https://www.fsrao.ca/media/11501/download
  - https://www.xenonstack.com/blog/data-analytics-in-insurance

  - https://business.wisc.edu/wp-content/uploads/2021/07/ProjDescription_Web.pdf
  - https://www.swissre.com/risk-knowledge/driving-digital-insurance-solutions/connected-car-how-data-analytics-is-shaping-the-future-of-auto-insurance.html
- **A/B Hypothesis Testing**
  - https://www.engagys.com/insights/a-b-testing-the-key-to-effective-healthcare-communications
  - https://www.linkedin.com/pulse/abcs-ab-testing-healthcare-marketing-daniella-koren/
  - https://medium.com/tiket-com/a-b-testing-hypothesis-testing-f9624ea5580e
  - https://www.optimizely.com/insights/blog/why-an-experiment-without-a-hypothesis-is-dead-on-arrival/
- **Data Version Control(DVC)**
  - https://dvc.org/
  - https://dvc.org/doc/user-guide
- **Statistical Modeling:**
  - https://www.heavy.ai/technical-glossary/statistical-modeling
  - https://www.coursera.org/articles/statistical-modeling
  - https://www.statlect.com/glossary/statistical-model
  - https://builtin.com/data-science/random-forest-algorithm
  - https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/#:~:text=Logistic%20Regression%20is%20another%20statistical,pass%20this%20exam%20or%20not.
  - https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/
- **Version control – Git**
  - What is version control | Atlassian
  - Learn Git branching -- interactive way to learn Git
  - Git with large files
  - Which files to not track and how to not track them? | Atlassian
  - .gitignore docs
  - Conventional commits -- lightweight convention on top of commit messages.
- **CI/CD**
  - What is Continuous Integration | Atlassian
  - DevOps Pipeline | Atlassian
  - 7 Popular Open Source CI/CD Tools - DevOps.com
  - Setting up a CI/CD pipeline on Github