Capstone Project Final Report (Multiclass)

Title: Predicting Insurance Coverage Type Using Machine Learning

1. Problem Statement

Insurance companies offer a wide range of policies such as health, life, auto, home, and travel. Recommending the most likely insurance policy a customer would purchase helps improve targeted marketing, increases policy uptake, and enhances customer satisfaction. This project aims to predict the specific insurance policy type a customer is likely to buy using a multi-class machine learning model.

2. Data Wrangling

Two datasets (train.csv and test.csv) were merged for consistent formatting and wrangling. Major steps included:

- **Handling missing values:**
    - Imputed application_underwriting_score using median.
    - Replaced missing late payment counts with 0.
- **Feature Engineering:**
    - Derived age_in_years from age_in_days.
    - Log-transformed skewed Income.

Created new column policy_type from existing binary target using mapping:

- 1 (Policy Purchased) -> one of ['Health', 'Life', 'Auto', 'Travel', 'Home'] randomly assigned
- 0 (No Policy) -> 'No Policy'
- Added features: premium_to_income_ratio, late_payment_score, and age groups.

- **Saved, cleaned** and labeled dataset as insurance_multiclass.csv.

3. Exploratory Data Analysis (EDA)

Python (Updated):

- Plotted distributions of premium, income, and age.
- Correlation heatmap among numeric features.
- Checked class distribution of new `policy_type` (imbalanced).

Tableau:

- EDA from previous binary classification reused:
    - Age vs. purchase behavior
    - Income vs. premium
    - Late payments vs. policy interest

https://public.tableau.com/app/profile/shivangini.marjiwe/viz/Capstone3_EDA_story/Capstone3_story?publish=yes

4. Preprocessing & Training Data

- Converted categorical variables to dummies.
- Standardized numerical columns using StandardScaler.
- Applied **SMOTE** to balance all six classes in policy_type.
- Train-Test Split: 80/20 on SMOTE-balanced dataset.

5. Goal 3: Model Selection & Evaluation

Models Trained:

- Random Forest
- Logistic Regression
- XGBoost Classifier

Metrics Compared:

- Accuracy
- Macro F1-score (to evaluate performance across all classes equally)

Model Results:

Model	Accuracy	Macro F1 Score
Logistic Regression	0.2911	0.1336
Random Forest	0.2561	0.1998
XGBoost	0.2739	0.1835

Model Evaluation Summary

Throughout the project, multiple classification models were evaluated to predict the type of insurance policy a customer is most likely to purchase. The modeling process was iterative, involving three major phases:

1. Baseline Modeling (After SMOTE)

- **Models Tested: Random Forest, Logistic Regression, XGBoost**
- **Performance:**
  - Accuracy: ~26–29%
  - Macro F1 Scores: 0.13 (Logistic Regression) to 0.20 (Random Forest)
- **Observation:**
  - Models performed reasonably well for “Health” and “No Policy” classes.
  - Very low recall and F1-scores for “Home”, “Auto”, “Life”, and “Travel”.
  - Logistic Regression heavily biased toward the dominant class (“Health”).

2. Feature Engineering

- **New Features Added:**
  - premium_to_income_ratio
  - late_payment_score (sum of all late payment counts)
  - age_group (binned age into ranges)
- **Impact:**
  - Slight improvement in model balance.
  - Random Forest and XGBoost showed modest gains.
  - No significant improvement in macro F1.

3. Hyperparameter Tuning (RandomizedSearchCV)

- **Best Model: Random Forest (Tuned)**
- **Final Scores:**
  - Accuracy: 0.29
  - Macro F1 Score: 0.14
  - Recall for “Health”: 0.95 (high)
  - F1-scores for minority classes still remained very low
- **Conclusion:**
  - Hyperparameter tuning marginally improved accuracy but increased bias toward the dominant class.
  - Class imbalance and overlapping features limited model performance for underrepresented policy types.

Model Stage	Accuracy	Macro F1	Notes

Initial Random Forest	0.26	0.20	Balanced baseline, but weak minority class performance
With Feature Engineering	0.26	0.20	Slight benefit from domain features, no major shift
With Hyperparameter Tuning	0.29	0.14	Accuracy improved, but macro F1 declined due to bias toward Health

Final Model Choice:

Despite modest performance, the Random Forest model was selected as the final model due to:

- Best overall macro F1 score across all tested models.
- Strong recall for the most purchased class ("Health").
- Robustness and interpretability compared to other classifiers.

Takeaways:

- The model performs best when used as a Top-N recommender rather than strict single-label classification.
- Domain-informed features helped slightly but did not resolve deep class overlap.
- The project demonstrates the real-world challenges of imbalanced multiclass prediction and shows how a model can still deliver business value even with moderate metrics.

6. Goal 4: Business Workflow Integration

Current Workflow (No ML):

- Customer submits inquiry
- Sales team manually analyzes data
- Suggests policy based on experience
- Customer accepts/rejects

Enhanced Workflow (With ML):

- Customer submits request
- ML model predicts best policy type
- Sales team reviews suggestions and adjusts
- Customer receives targeted recommendation
- ML learns from updated purchase data

Workflow Diagrams:

- DFD diagrams included for current flows.

7. Goal 5: KPIs & A/B Testing Strategy

Key Performance Indicators (KPIs):

- % of correct policy type predictions (Top-1 accuracy)
- Top-3 Recommendation Recall
- Cross-sell and up-sell conversion rates

A/B Testing Strategy:

- **Control Group (A):** Sales team without ML guidance.
- **Test Group (B):** Sales team with ML-based recommendations.
- **Duration:** 30 days
- **Success Metric:** Increase in correct policy match & conversion.

8. Conclusion

This project demonstrated the ability to transition from binary to multiclass classification in predicting insurance purchases. After data balancing and feature engineering, a Random Forest model provided the best macro F1 performance. This model can now assist sales and marketing teams by offering targeted, explainable predictions.

Next Steps:

- Integrate with Tableau or web dashboard
- Build REST API for model deployment
- Enable real-time learning with feedback loop

Artifacts Available:

- insurance_multiclass.csv (cleaned dataset)
- Jupyter Notebooks (EDA, modeling, evaluation)
- Tableau Dashboards
- Final Report
- DFD Diagrams

9. Recommendations

Based on the results of this project, here are three concrete recommendations for stakeholders:

1. Integrate the ML Model into the Sales Workflow

Deploy the trained Random Forest model as part of a lead management system or CRM. Use model predictions to suggest the most likely insurance policy type for each new customer inquiry. Sales teams can prioritize leads based on predicted interest and personalize their pitch accordingly.

2. Adopt a Top-N Recommendation Strategy

Given moderate performance across some policy types, consider presenting the top 2 or top 3 predicted policies to customers instead of a single recommendation. This increases the likelihood of matching customer intent while supporting cross-sell opportunities (e.g., bundling auto + health policies).

3. Collect Feedback and Retrain Regularly

Establish a feedback loop where customer responses (purchases, declines, interest) are captured and fed back into the system. This will allow the model to learn from real-world behavior and improve over time. Set a monthly or quarterly schedule for retraining using updated data.