USC Marshall
School of Business

DSO 530

Insurance Loss Analytics

For Professor **Paromita Dubey**

Group 25

Wanyi (Eve) Miao - Contact Person
(Email: wanyimia@usc.edu & #ID: 6705304004)

Yini Li (#ID: 8458454270)
Yuxuan Li (#ID: 9412115514)
Ziyu (Cathy) Fang (#ID: 5749501519)
Chih-Chi (Gina) Liao (#ID: 4418770321)

Date of Report: 5/7/2025

Executive Summary

This project aimed to develop predictive models for three critical insurance metrics: Claim Status (CS), Loss Cost (LC), and Historically Adjusted Loss Cost (HALC). These models support underwriting, pricing, and risk segmentation by identifying high-risk policyholders and accurately estimating expected financial losses.

Task 1: Predicting LC and HALC

We engineered meaningful features such as license age, vehicle age, and policy tenure. Recognizing the highly skewed, zero-inflated nature of LC and HALC, we capped extreme values to improve model robustness. Using LASSO regression, we selected 12 key predictors covering behavioral patterns (e.g., cancellations, tenure), vehicle risk, and pricing signals.

Multiple regression models were evaluated, including Tweedie GLM, Neural Networks, LightGBM, and XGBoost. Tweedie GLM offered strong interpretability, while XGBoost captured complex non-linear patterns. Our final solution was an ensemble model combining Tweedie GLM (95%) and XGBoost (5%), achieving the lowest RMSEs: 441.70 for LC and 824.42 for HALC. SHAP analysis confirmed that the model prioritized business-relevant variables, including policy duration, net premium, and cancellation history.

Task 2: Predicting Claim Status (CS)

For CS, we prioritized recall to minimize false negatives, reducing the risk of underpricing. After evaluating XGBoost, Random Forest, Gradient Boosting, and Logistic Regression, XGBoost emerged as the top performer (ROC-AUC = 0.705, Recall = 0.651). Post-tuning, recall improved to 0.777 and ROC-AUC to 0.716.

To address class imbalance, we integrated SMOTE into a cross-validation framework. SHAP values validated that the model relied on logical predictors such as cancellation history, payment frequency, vehicle type, and tenure.

Business Insights & Innovations

- **Accuracy vs. Interpretability**: For CS, recall was emphasized; for LC/HALC, model interpretability supported pricing transparency.
- **Challenges Overcome**: Addressed severe class imbalance, skewed targets, and complex feature engineering.
- **Innovations**: Used model ensembling to combine interpretability and predictive power; explored risk-type segmentation for future improvements.

Conclusion

Our final models successfully balanced predictive performance with business relevance and interpretability. The solutions provide actionable insights to enhance underwriting, pricing strategies, and risk management, while maintaining alignment with industry best practices and regulatory expectations.

Task 1: Predicting LC and HALC

Objective

Our objective is to build a model to predict Loss Cost (LC) and Historically Adjusted Loss Cost (HALC). For LC and HALC, we aimed to estimate the expected financial loss associated with each policy, helping guide premium pricing, risk segmentation, and reserve planning.By accurately predicting these outcomes, insurers can better align pricing with underlying risk and maintain financial stability.

Data Preprocessing & Feature Engineering

We began by transforming several raw columns into more interpretable features. For example, we created license_age, vehicle_age, and duration using the date fields provided. We also performed one-hot encoding on categorical features, such as: vehicle type and fuel source, to capture more nuanced vehicle risk signals.

Upon exploring the target distributions for LC and HALC, we observed that over 33,000 entries had a value of 0, and a small portion of entries had extremely high values. Hence, we capped LC at $20,000 and HALC at $30,000 to mitigate the influence of extreme values and make the model more robust. For other numerical features, we identified strong skewness and chose not to transform them further. From a business perspective, this skewness is meaningful: in insurance, a small group of policyholders often drives a majority of the claims. Preserving this pattern allowed us to better capture risk.

Variable Selection

We applied Lasso regression (L1 regularization) to identify the most impactful features while removing redundant or noisy variables. Lasso shrinks less useful coefficients to zero. It helps to reduce multicollinearity and improve generalizability. This method was particularly aligned with our use of Tweedie GLM, which is sensitive to multicollinearity. Ultimately, we selected 12 variables that reflected behavioral patterns (e.g., policy duration, cancellations), vehicle risks (e.g., weight, power, age), and pricing-related signals (e.g., premium paid, urban vs rural). SHAP values were also used later in the process to validate our feature choices. *(*See appendix: Task 1 Lasso Variable Selection)*

Model Comparison & Evaluation

We tested several regression models, including Support Vector Regression (SVR), ElasticNet, Random Forest, Tweedie GLM, Neural Network, LightGBM, and XGBoost. SVR was excluded due to its inefficiency on large datasets. ElasticNet and Random Forest yielded high RMSE scores and were ultimately outperformed by other methods.

Tweedie GLM performed well with interpretable coefficients. Neural Networks showed slightly better RMSEs but sacrificed interpretability. LightGBM and XGBoost both captured non-linear interactions effectively. However, LightGBM was found to be highly correlated with XGBoost and slightly less accurate.

All models were tuned using k-fold cross-validation with GridSearchCV. Performance was primarily evaluated using RMSE for both LC and HALC.

Model	LC MSE	LC RMSE	HALC MSE	HALC RMSE
Tweedie GLM	197155.60	444.01	682975.21	826.42
Neural Network	196525.88	443.31	681166.44	825.33
LightGBM	198675.23	445.73	689198.83	830.18
XGBoost	198007.20	444.98	687738.49	829.3

Final Model Selection & Ensembling

We finalized on two models for ensembling: Tweedie GLM and XGBoost. We tested multiple weight combinations on the validation set. The best result came from a 95% weight on Tweedie GLM and 5% on XGBoost. This ensemble leveraged Tweedie's strength with zero-inflated continuous targets while allowing XGBoost to refine non-linear patterns. It achieved the lowest RMSE overall — 441.70 for Loss Cost (LC) and 824.42 for HALC — outperforming any individual model.

Model Interpretation

We used SHAP values to interpret the final ensemble. Policy duration was the most influential driver. Longer policies tended to be associated with higher claims. Other important features included net premium, number of cancelled policies, and payment frequency. These variables reflect real-world business risk: policyholders with longer tenure, higher payments, or churn history pose different risk profiles. The SHAP analysis confirmed that the model prioritized interpretable, business-aligned predictors, which helps build trust in its deployment. *(*See appendix: Task 1 SHAP Summary)*

Prediction Output

For the final prediction, we employed ensembled model (95% weighted Tweedie GLM and 5% weighted XGBoost) to generate prediction outputs for Loss Cost (LC) and Historically Adjusted Loss Cost (HALC). Both LC and HALC have skewed distributions, which aligns with typical insurance loss data where most policies have small or no losses and a few have very large losses. HALC is not only higher on average than LC but also shows greater variability and skew, likely reflecting adjustments for inflation or historical severity patterns.

Final Summary

Our final LC and HALC models leveraged a weighted ensemble of Tweedie GLM and XGBoost. This combination was selected after observing that Neural Networks compromised interpretability and LightGBM was highly correlated with XGBoost but less accurate. The ensemble achieved the lowest RMSE across all tested models. SHAP analysis confirmed that top features aligned with business logic, including policy duration, net premium, and cancellation history.

Ensembling Model

**See the innovation section*

Task 2: Predicting Claim Status (CS)

Objective

Our objective was to develop predictive models for Claim Status (CS), Loss Cost (LC), and Historically Adjusted Loss Cost (HALC). For CS, we aimed to identify policyholders likely to file claims, guiding underwriting and premium pricing by flagging high-risk customers.

Review of Approaches Considered

We initially evaluated four classifiers for CS: XGBoost, Random Forest, Gradient Boosting, and Logistic Regression. Simpler models like Decision Trees and SVMs were ruled out due to scalability issues and weaker recall on imbalanced datasets. For LC and HALC, we considered linear models (LASSO, Ridge, Tweedie regression) and tree-based regressors (Random Forest, Gradient Boosting, XGBoost). While Tweedie regression is suitable for zero-inflated data, tree-based models offered better performance and flexibility.

Class imbalance was a key challenge for CS. We addressed this using SMOTE in a train-test split and cross-validation setting. We prioritized recall to avoid false negatives, which are costly in insurance. Tree-based models were favored for LC and HALC due to skewed target distributions and their ability to model non-linear interactions.

Model	ROC-AUC	Recall	Precision	F1-Score
XGBoost	0.704792	0.651166	0.183101	0.285825
Random Forest	0.699125	0.190075	0.231551	0.208621
Gradient Boosting	0.702719	0.258970	0.219695	0.237618
Logistic Regression	0.652084	0.341362	0.188409	0.242781

Final Model Selection and Tuning

XGBoost achieved the best performance in initial CS tests (ROC-AUC = 0.705, Recall = 0.651), outperforming other classifiers. We tuned it further using RandomizedSearchCV over a parameter grid (e.g., n_estimators, max_depth, learning_rate, subsample, colsample_bytree, gamma, scale_pos_weight) with 5-fold Stratified CV. This improved recall to 0.777 and ROC-AUC to 0.716, confirming XGBoost as the best classifier.

For LC and HALC, we finalized on Gradient Boosting and XGBoost regressors based on their lower Mean Squared Errors. Linear models underperformed due to the target distribution's skewness and inability to model complex interactions.

Model Interpretation

We used SHAP values to interpret our tuned XGBoost CS model. The top predictors included: X.12 (canceled policies): more cancellations strongly increased claim probability; X.13 (payment frequency): semi-annual payers had higher risk; Vehicle characteristics and tenure features: newer policyholders and certain vehicle types showed higher claim risk.

SHAP confirmed the model relied on business-relevant variables, ensuring interpretability for stakeholders. *(*See appendix: Task 2 SHAP Summary)*

Prediction Output

For the final model deployment, we applied the tuned XGBoost classifier pipeline to the test dataset. The predicted probabilities ranged from near zero up to approximately 0.81, with most values concentrated below 0.27 and an average probability of 0.157. This reflects a reasonable alignment with the expected class imbalance, identifying a small subset of policyholders with elevated claim risk.

Final Summary

Our final CS model used a tuned XGBoost pipeline with SMOTE, selected for its high recall and ROC-AUC. LC and HALC predictions used tree-based regressors to capture skewed distributions. Our approach balanced predictive strength with model interpretability using SHAP.

Innovations

- **Business understanding:** For claim status, accuracy, particularly recall, was prioritized to avoid underpricing by minimizing false negatives. For LC/HALC, interpretability was emphasized to support transparent and justifiable premium setting. Key business-driven variables included cancellation history, policy tenure, payment frequency, and vehicle attributes like horsepower and fuel type, which reflect risk factors commonly used in underwriting. The model is auto-insurance specific and cannot be applied to life insurance without retraining on domain-relevant features such as health, lifestyle, and demographic factors.
- **Challenges:** We addressed severe class imbalance for CS and heavily right-skewed targets for LC/HALC, both of which complicated model training and validation. Feature engineering was essential to transform raw dates into meaningful features like age, policy tenure, and driving years, enabling the models to learn relevant patterns. Balancing accuracy and interpretability was a continuous challenge, requiring the use of complex tree-based models supported by SHAP for transparency.
- **Model ensembling:** We chose to ensemble Tweedie GLM and XGBoost to leverage the strengths of both linear and non-linear modeling approaches. Tweedie GLM is well-suited for insurance data due to its ability to handle zero-inflated, continuous target variables and provide interpretable coefficients. It aligns closely with business logic and traditional actuarial models. However, it may struggle to capture complex interactions or non-linear relationships in the data. On the other hand, XGBoost excels at modeling non-linear patterns and interactions between features, offering strong predictive power, but at the cost of reduced interpretability. By combining the two, we maintained the interpretability and robustness of Tweedie while enhancing predictive accuracy with XGBoost's non-linear learning.
- **Risk-type segmentation models:** To enhance model performance and better capture risk patterns, we considered segmenting the data by risk type, specifically distinguishing between passenger cars and non-passenger vehicles. These groups exhibit significantly different behaviors in terms of claim frequency, severity, and cost. Training a single model on the combined data risks masking these distinctions, leading to less precise predictions. By building separate models for each segment, we aim to improve accuracy, reduce variance, and align the modeling approach more closely with real-world underwriting practices.

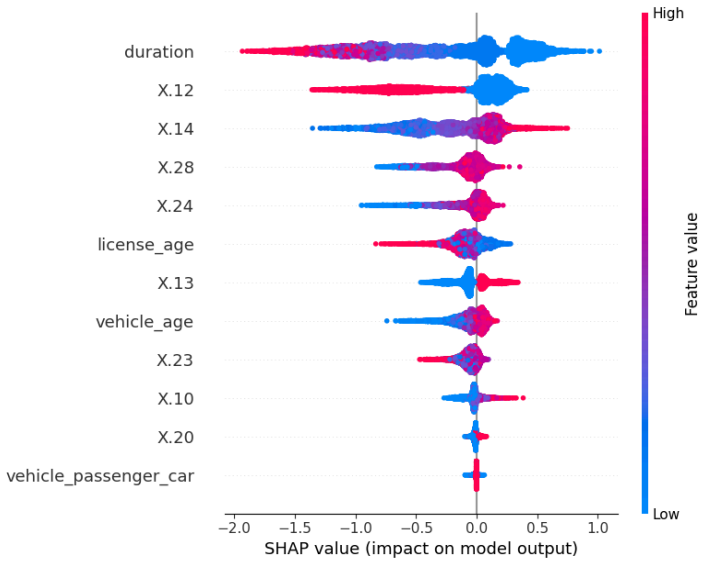
Appendix

Task 1 Lasso Variable Selection:

```
from sklearn.linear_model import LassoCV
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split

# Drop target-leaking columns
features =  ['X.8', 'X.9', 'X.10','X.12', 'X.13', 'X.14', 'X.20','X.23', 'X.24', 'X.25', 'X.28',
            'vehicle_van', 'vehicle_passenger_car', 'vehicle_agri',
            'vehicle_age', 'driver_age', 'license_age', 'duration']

# Ensure X and y are derived from the same DataFrame (after dropping missing values)
df_model = df_train[features + ['LC']]
X = df_model[features]
y = df_model['LC']  # or use 'HALC' if needed

# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Fit LASSO with cross-validation
lasso = LassoCV(cv=5, random_state=42)
lasso.fit(X_scaled, y)

# Select features with non-zero coefficients
selected_features = X.columns[lasso.coef_ != 0]
print("Selected Features by LASSO:")
print(selected_features)
```

```
Selected Features by LASSO:
Index(['X.10', 'X.12', 'X.13', 'X.14', 'X.20', 'X.23', 'X.24', 'X.28',
       'vehicle_passenger_car', 'vehicle_age', 'license_age', 'duration'],
      dtype='object')
```

Task 1 SHAP Summary:



Task 2 SHAP Summary: