Experiment Notebook

Setup Environment

In [1]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
!pip install -q utstd

from utstd.folders import *
from utstd.ipyrenders import *

at = AtFolder(
    course_code=36106,
    assignment="AT1",
)
at.run()

import warnings
warnings.simplefilter(action='ignore')
```

```
ERROR: Could not install packages due to an OSError: [WinError 5] 拒绝访问。: 'C:\\Users\\brohao\\AppData\\Local\\Programs\\Python\\Python311\\Lib\\site-packages\\~2learn\\.libs\\msvcp140.dll'
Consider using the `--user` option or check the permissions.


[notice] A new release of pip available: 22.3.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
You can now save your data files in: c:\Users\brohao\Desktop\UTS\36106\AT1\36106\assignment\AT1\data
```

Student Information

In [2]:
```
student_name = "Jiayu Hao"
student_id = "25948860"
```

In [3]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h1", key='student_name', value=student_name)
```

student_name

Jiayu Hao

In [4]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h1", key='student_id', value=student_id)
```

student_id

25948860

---

0. Python Packages

0.a Install Additional Packages

> If you are using additional packages, you need to install them here using the command: `! pip install <package_name>`

In [5]:
```
!pip install numpy
!pip install scikit-learn
```

```
Requirement already satisfied: numpy in c:\users\brohao\appdata\local\programs\python\python31
1\lib\site-packages (2.3.2)
[notice] A new release of pip available: 22.3.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
Requirement already satisfied: scikit-learn in c:\users\brohao\appdata\local\programs\python\p
ython311\lib\site-packages (1.6.1)
Requirement already satisfied: numpy>=1.19.5 in c:\users\brohao\appdata\local\programs\python
\python311\lib\site-packages (from scikit-learn) (2.3.2)
Requirement already satisfied: scipy>=1.6.0 in c:\users\brohao\appdata\local\programs\python\p
ython311\lib\site-packages (from scikit-learn) (1.16.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\brohao\appdata\local\programs\python
\python311\lib\site-packages (from scikit-learn) (1.5.1)
Requirement already satisfied: threadpoolctl>=3.1.0 in c:\users\brohao\appdata\local\programs
\python\python311\lib\site-packages (from scikit-learn) (3.6.0)
[notice] A new release of pip available: 22.3.1 -> 25.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

0.b Import Packages

In [6]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
import pandas as pd
import altair as alt
```

In [7]:
```
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_absolute_error
import numpy as np
```

A. Experiment Description

In [8]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
experiment_id = "2"
print_tile(size="h1", key='experiment_id', value=experiment_id)
```

experiment_id

2

In [9]:
```
experiment_hypothesis = """
The hypothesis is that Lasso with L1 regularization can keep accuracy while performing automat
The question is whether different alpha values can lower validation MAE and at the same time r
It is worthwhile because insurance data often has many correlated or low-value features, and
This creates a simpler and more interpretable model, reduces noise, improves stability.
"""
```

In [10]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h3", key='experiment_hypothesis', value=experiment_hypothesis)
```

experiment_hypothesis

The hypothesis is that Lasso with L1 regularization can keep accuracy while performing automatic feature selection, making it more robust than Ridge when redundant or correlated features exist. The question is whether different alpha values can lower validation MAE and at the same time reduce the number of active features, improving interpretability and reducing noise. It is worthwhile because insurance data often has many correlated or low-value features, and Lasso can shrink or remove them. This creates a simpler and more interpretable model, reduces noise, improves stability.

In [11]:
```
# Detail what will be the expected outcome of the experiment. If possible, estimate the goal y
# List the possible scenarios resulting from this experiment.
experiment_expectations = """
The expected outcome is that Lasso will match or improve validation MAE compared to Ridge (~12
This should give a simpler and more stable model, lower overfitting risk, and provide a clear
If MAE improves, Lasso can be adopted; if similar, it still adds value by improving interpreta
"""
```

In [12]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h3", key='experiment_expectations', value=experiment_expectations)
```

experiment_expectations

The expected outcome is that Lasso will match or improve validation MAE compared to Ridge (~127) while reducing the number of active features. This should give a simpler and more stable model, lower overfitting risk, and provide a clear list of key drivers for premium pricing. If MAE improves, Lasso can be adopted; if similar, it still adds value by improving interpretability; if worse, alpha can be retuned or other models considered.

B. Feature Selection

In [13]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
# Load data
try:
  X_train = pd.read_csv(at.folder_path / 'X_train.csv')
  y_train = pd.read_csv(at.folder_path / 'y_train.csv')

  X_val = pd.read_csv(at.folder_path / 'X_val.csv')
  y_val = pd.read_csv(at.folder_path / 'y_val.csv')

  X_test = pd.read_csv(at.folder_path / 'X_test.csv')
  y_test = pd.read_csv(at.folder_path / 'y_test.csv')
except Exception as e:
  print(e)
```

In [14]:
```
features_list = list(X_train.columns)
print("Number of features:", len(features_list))
```

```
Number of features: 34
```

In [15]:
```
feature_selection_explanations = """
The selected features include numerical variables such as customer seniority, vehicle attribut
Identifiers such as ID, name, address, phone, and email were removed because they do not contr
Only features with direct or indirect impact on premiums were kept.
"""
```

In [16]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h3", key='feature_selection_explanations', value=feature_selection_explanati
```

feature_selection_explanations

The selected features include numerical variables such as customer seniority, vehicle attributes, and claim history, as well as one-hot encoded categorical variables like gender, policy type, and channel. Identifiers such as ID, name, address, phone, and email were removed because they do not contribute to prediction. Only features with direct or indirect impact on premiums were kept.

C. Train Machine Learning Model

C.1 Import Algorithm

In [17]:
```
algorithm_selection_explanations = """
Lasso is a good fit because L1 regularization can shrink unimportant feature coefficients to z
It also handles multicollinearity by selecting only part of a correlated group, making results
"""
```

In [18]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h3", key='algorithm_selection_explanations', value=algorithm_selection_expla
```

algorithm_selection_explanations

Lasso is a good fit because L1 regularization can shrink unimportant feature coefficients to zero, reducing dimensionality and improving interpretability. It also handles multicollinearity by selecting only part of a correlated group, making results easier to explain.

C.2 Set Hyperparameters

In [19]:
```
# Set Hyperparameters
alphas = [0.0001, 0.001, 0.01, 0.1, 1, 10]
# Set Hyperparameters through the Result
alphas_new = [0.07, 0.08, 0.09, 0.1, 0.11]
```

In [20]:
```
hyperparameters_selection_explanations = """
We tune alpha because it controls the strength of regularization.
A larger alpha makes the model sparser but may underfit, while a smaller alpha is closer to li
We test a range of values (0.0001 to 10) and select the best on the validation set.
"""
```

In [21]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h3", key='hyperparameters_selection_explanations', value=hyperparameters_sel
```

We tune alpha because it controls the strength of regularization. A larger alpha makes the model sparser but may underfit, while a smaller alpha is closer to linear regression and may overfit. We test a range of values (0.0001 to 10) and select the best on the validation set.

C.3 Fit Model

In [22]:
```
lasso_results = []
coef_cards = []  # Record Non-zero coefs

for a in alphas:
    lasso = Lasso(alpha=a, random_state=42, max_iter=10000)
    lasso.fit(X_train, y_train)

    y_tr_pred = lasso.predict(X_train)
    y_va_pred = lasso.predict(X_val)

    tr_mae = mean_absolute_error(y_train, y_tr_pred)
    va_mae = mean_absolute_error(y_val,   y_va_pred)
    nonzero = int(np.sum(lasso.coef_ != 0))

    lasso_results.append((a, tr_mae, va_mae, nonzero))
    coef_cards.append((a, nonzero))

print("Lasso Regression Results:")
print("alpha | Train MAE | Validation MAE | #Nonzero Coefs")
for a, tr, va, nnz in lasso_results:
    print(f"{a:5} | {tr:9.2f} | {va:14.2f} | {nnz:13d}")

# Choose best alpha by Validation MAE
best_alpha, best_train_mae, best_val_mae, best_nnz = sorted(lasso_results, key=lambda x: x[2]
print("\nBest alpha by Validation MAE:", best_alpha)
print(f"Best MAE (Train/Val): {best_train_mae:.2f} / {best_val_mae:.2f}")
print("Nonzero Coefs:", best_nnz)
```

```
Lasso Regression Results:
alpha | Train MAE | Validation MAE | #Nonzero Coefs
0.0001 |     34.67 |         127.56 |            30
0.001 |     34.68 |         127.54 |            30
 0.01 |     34.68 |         127.36 |            26
  0.1 |     34.76 |         126.54 |            22
    1 |     35.34 |         130.14 |            15
   10 |     38.09 |         147.72 |             0

Best alpha by Validation MAE: 0.1
Best MAE (Train/Val): 34.76 / 126.54
Nonzero Coefs: 22
```

In [23]:
```
lasso_results = []
coef_cards = []  # Record Non-zero coefs

for a in alphas_new:
    lasso = Lasso(alpha=a, random_state=42, max_iter=10000)
    lasso.fit(X_train, y_train)

    y_tr_pred = lasso.predict(X_train)
    y_va_pred = lasso.predict(X_val)

    tr_mae = mean_absolute_error(y_train, y_tr_pred)
    va_mae = mean_absolute_error(y_val,   y_va_pred)
```

```
    nonzero = int(np.sum(lasso.coef_ != 0))

    lasso_results.append((a, tr_mae, va_mae, nonzero))
    coef_cards.append((a, nonzero))

print("=== Lasso Regression Results ===")
print("alpha | Train MAE | Validation MAE | Nonzero Coefs")
for a, tr, va, nnz in lasso_results:
    print(f"{a:5} | {tr:9.2f} | {va:14.2f} | {nnz:13d}")

# Choose best alpha by Validation MAE
best_alpha, best_train_mae, best_val_mae, best_nnz = sorted(lasso_results, key=lambda x: x[2]
print("\nBest alpha by Validation MAE:", best_alpha)
print(f"Best MAE (Train/Val): {best_train_mae:.2f} / {best_val_mae:.2f}")
print("Nonzero Coefs:", best_nnz)
```

```
=== Lasso Regression Results ===
alpha | Train MAE | Validation MAE | Nonzero Coefs
 0.07 |     34.74 |         126.51 |            23
 0.08 |     34.75 |         126.46 |            23
 0.09 |     34.76 |         126.50 |            23
  0.1 |     34.76 |         126.54 |            22
 0.11 |     34.77 |         126.58 |            22

Best alpha by Validation MAE: 0.08
Best MAE (Train/Val): 34.75 / 126.46
Nonzero Coefs: 23
```

In [24]:

```
# Fit
best = Lasso(alpha=0.08, random_state=42, max_iter=5000).fit(X_train, y_train)

# Check shape
print("coef shape:", best.coef_.shape)
print("X_train shape:", X_train.shape)

# Use Column name of X_train
coef_series = pd.Series(best.coef_, index=X_train.columns)

# Print 23 Non-zero features
print(coef_series[coef_series != 0].sort_values(key=abs, ascending=False).head(23))
```

```
coef shape: (34,)
X_train shape: (32136, 34)
payment_method_0                        -1.610867e+01
second_driver_0                         -1.224851e+01
distribution_channel_1                   7.092442e+00
vehicle_value                            6.371612e+00
policy_type_3                           -5.709206e+00
total_claims_number_ratio                3.935519e+00
car_age                                 -3.595484e+00
lapsed_policies                          3.290664e+00
driving_experience                      -3.145398e+00
total_claims_number_in_history           2.896067e+00
seniority                               -2.518055e+00
vehicle_weight                           2.167951e+00
current_policies_held                   -2.141609e+00
vehicle_length                           1.823253e+00
total_claims_number_in_current_year     -1.638586e+00
vehicle_horsepower                       1.235289e+00
gender_m                                 8.560614e-01
vehicle_fuel_type_D                     -8.033245e-01
vehicle_cylinder                        -6.283746e-01
max_policies_held                       -4.702714e-01
total_claims_cost_in_current_year       -2.830757e-02
payment_method_1                         6.965024e-14
second_driver_1                          1.210726e-15
dtype: float64
```

D. Model Evaluation

D.1 Model Technical Performance

In [25]:
```
# Provide some explanations on model performance
model_performance_explanations = """
The model performance is read by tracking validation MAE across alpha values.
Small alpha gives low train MAE but high validation MAE, showing overfitting.
As alpha increases, validation MAE may drop to a minimum, while the number of nonzero coeffic
The best Lasso model is compared with Ridge and the baseline:
if validation MAE is lower, L1 feature selection adds value; if not, more nonlinear or intera
"""
```

In [26]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h3", key='model_performance_explanations', value=model_performance_explanati
```

model_performance_explanations

The model performance is read by tracking validation MAE across alpha values. Small alpha gives low train MAE but high validation MAE, showing overfitting. As alpha increases, validation MAE may drop to a minimum, while the number of nonzero coefficients falls, showing feature sparsity. The best Lasso model is compared with Ridge and the baseline: if validation MAE is lower, L1 feature selection adds value; if not, more nonlinear or interaction features may be needed.

D.2 Business Impact from Current Model Performance

In [27]:
```
# Interpret the results of the experiments related to the business objective set earlier. Est
business_impacts_explanations = """
```

```
Lower MAE means smaller pricing errors, which supports fairer premiums and reduces both custor
Lasso also provides a clear list of key factors through nonzero coefficients, helping the pric
If some features are dropped, it shows they may add noise or instability, and the business car
"""
```

In [28]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h3", key='business_impacts_explanations', value=business_impacts_explanation
```

business_impacts_explanations

Lower MAE means smaller pricing errors, which supports fairer premiums and reduces both customer loss from overpricing and claim risk from underpricing. Lasso also provides a clear list of key factors through nonzero coefficients, helping the pricing team understand which variables matter most for premiums and guiding risk control and product strategy. If some features are dropped, it shows they may add noise or instability, and the business can review whether to improve or stop collecting them.

E. Conclusion

In [29]:
```
# <Student to fill this section and then remove this comment>
experiment_outcome = "Hypothesis Partially Confirmed"
```

In [30]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h2", key='experiment_outcomes_explanations', value=experiment_outcome)
```

experiment_outcomes_explanations

Hypothesis Partially Confirmed

In [31]:
```
# <Student to fill this section and then remove this comment>
experiment_results_explanations = """
The Lasso experiment shows that L1 regularization can both lower validation MAE (126.5 vs. 148
The result improves interpretability, which still can be improved, so further experiments are
The next steps are:
(1) feature engineering with interactions and nonlinear terms,;
(2) ElasticNet to balance sparsity and stability;
(3) alternative models such as tree-based methods;
(4) data quality improvements;
(5) segmented modeling by policy or fuel type.
Experiments should continue following the ranked priorities.
"""
```

In [32]:
```
# DO NOT MODIFY THE CODE IN THIS CELL
print_tile(size="h2", key='experiment_results_explanations', value=experiment_results_explana
```

The Lasso experiment shows that L1 regularization can both lower validation MAE (126.5 vs. 148 baseline) and perform automatic feature selection, leaving 22 nonzero coefficients that give a clear list of drivers for premiums. The result improves interpretability, which still can be improved, so further experiments are worthwhile. The next steps are: (1) feature engineering with interactions and nonlinear terms,; (2) ElasticNet to balance sparsity and stability; (3) alternative models such as tree-based methods; (4) data quality improvements; (5) segmented modeling by policy or fuel type. Experiments should continue following the ranked priorities.