INSNOVA AUTO CLAIM CASE STUDY

Happy Penguins Presents

Introduction

Background: Create a rating plan for InsNova Auto Insurance Company based on the historical auto claim data.

Goal: Provide a method for predicting the claim cost for each policy

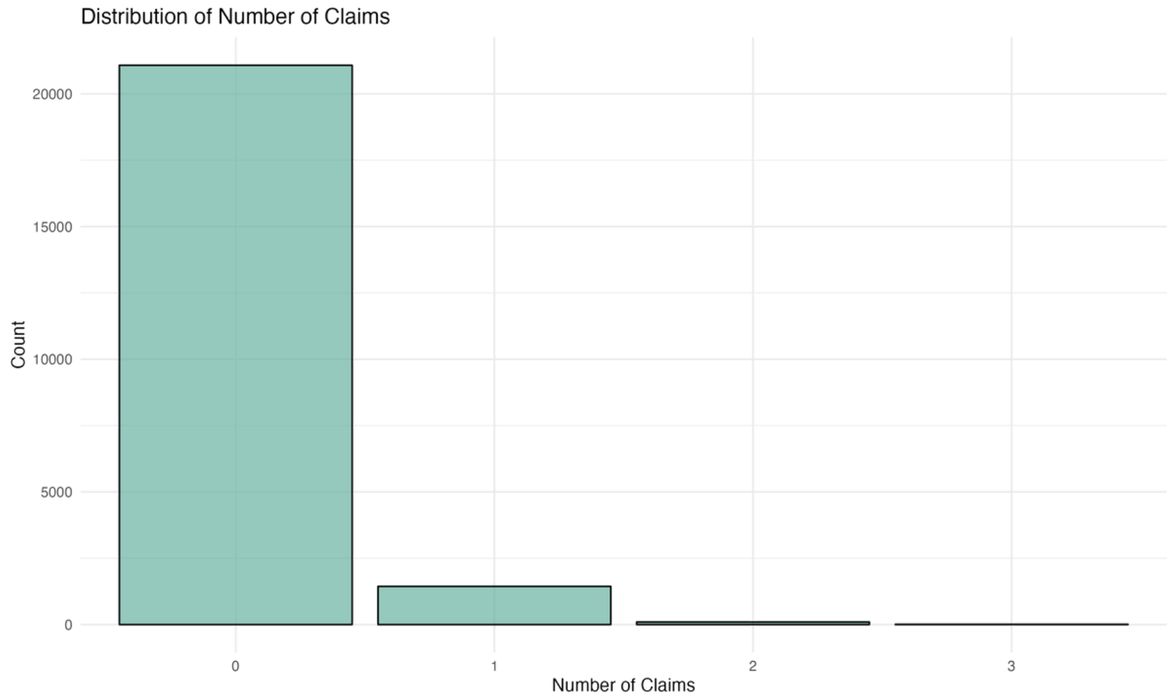
InsNova DataSet

- Based on one-year vehicle insurance policies from 2004 to 2005
- **45, 239** policies
- Around **6.8%** had at least one claim

Data Overview

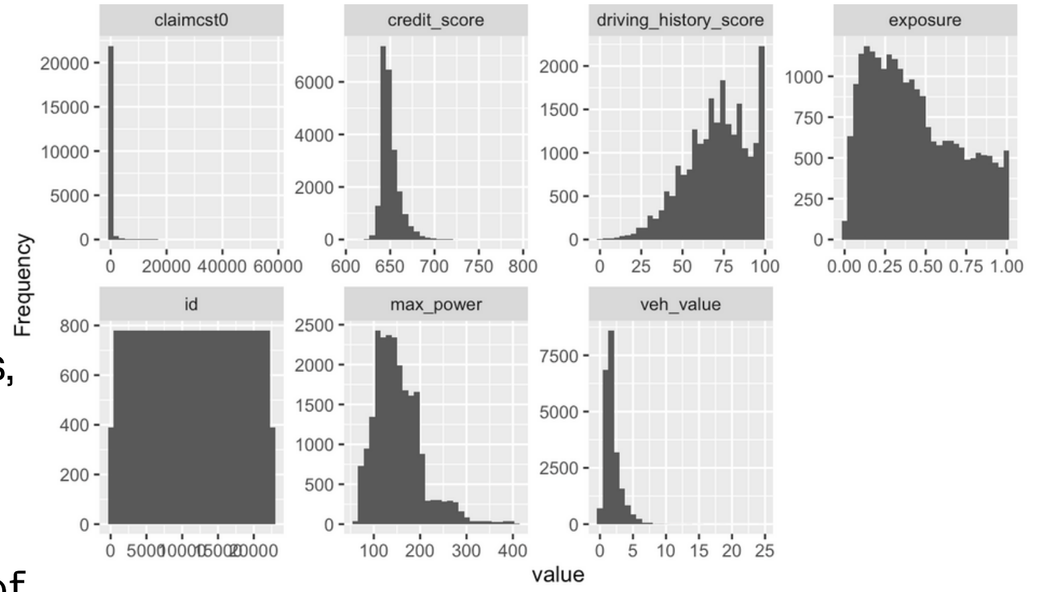
Training Dataset

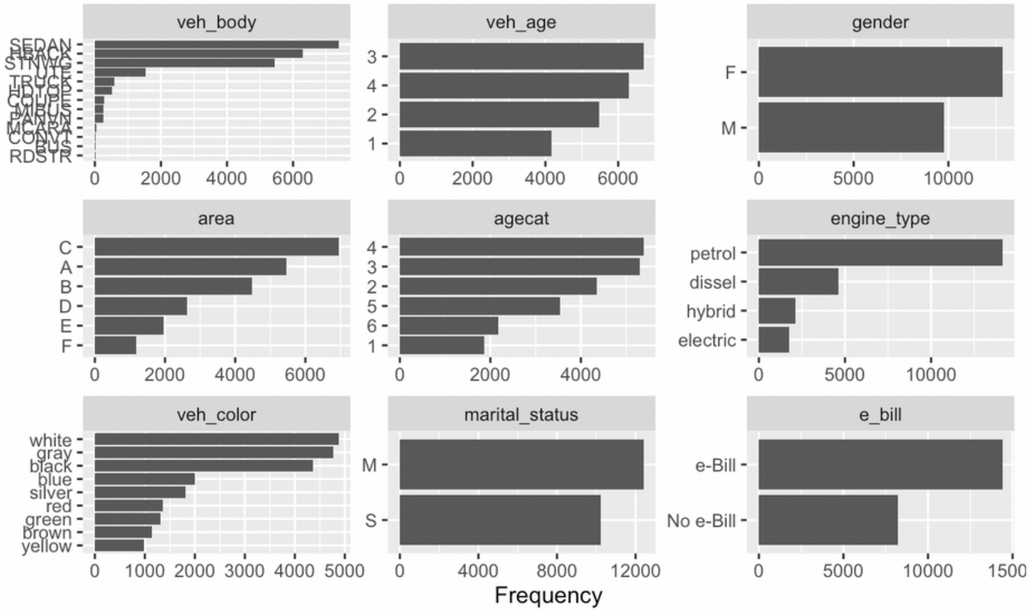
- Total number of policies: **22,619**
- Instances with no claims (claim number = 0 and claim cost = $0): **21,077**
- Instances with claims (claim number > 0): **1,542**



Variables

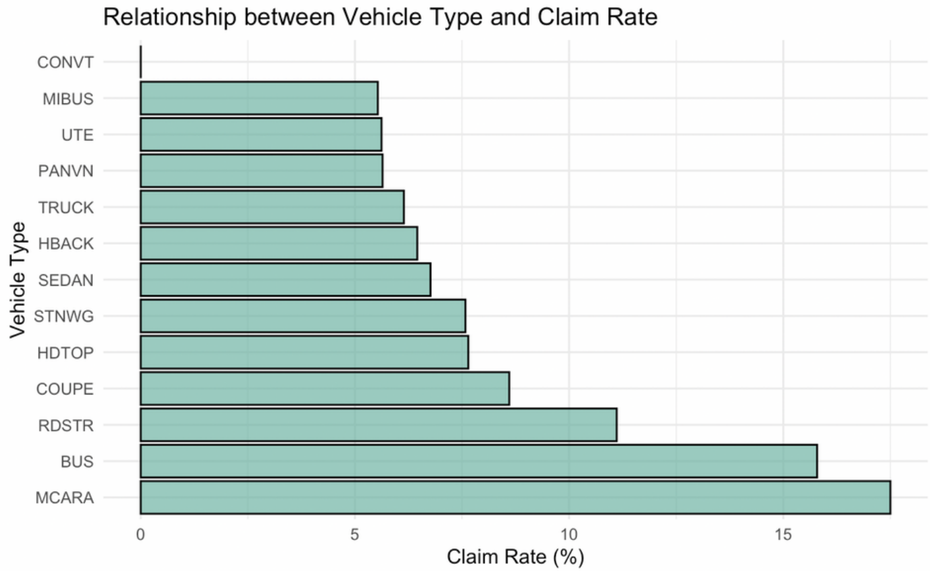
- **Outcome variable:** claim cost
- **Explanatory variables:**
  - **Total:** 18 variables
  - **Categorical:** Type of vehicles, age of vehicles, gender of driver, driving area of residence, driver's age, engine type of vehicles, color of vehicles, marital status of driver, indicator for paperless billing, most frequent driving date of the week, most frequent driving time of the day, term length, indicator for higher education
  - **Numerical:** Market value of the vehicle in $10,000's, exposure, max horsepower of vehicles, driving score based on past driving history, credit score
- **Omitted variable:** indicator of claim (0=no, 1=yes), the number of claims, id/policy key

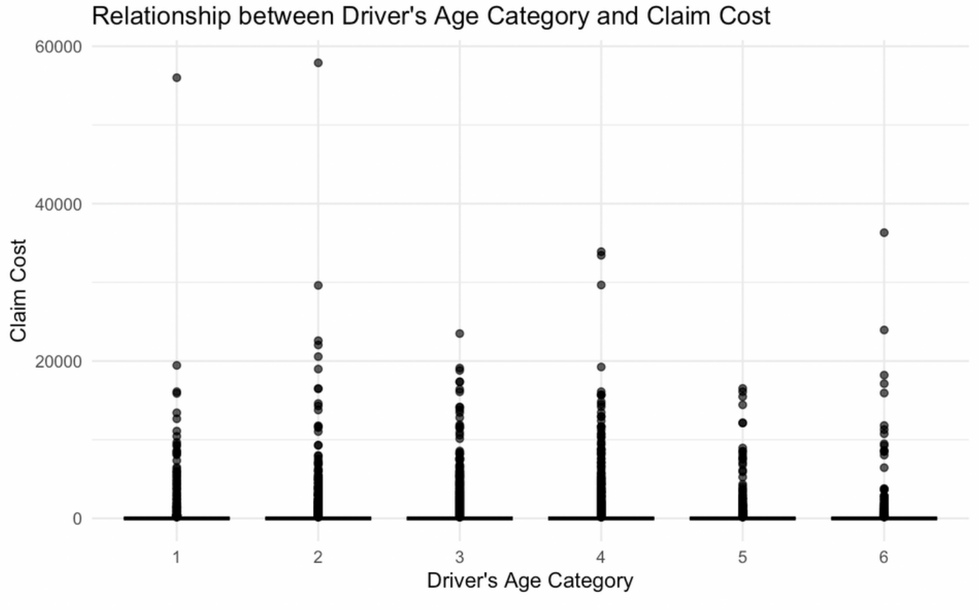




Variable Selection

- **Number of observation:** we used all the variables present in both the training and test data because we had plenty of observations. This allowed us to explore various patterns in the data more comprehensively
- **Post initial model building:** we assessed the feature importance to understand the impact of various variables on the model. Variables with lower importance were identified and subsequently removed. Following this refinement, the model was rerun to evaluate for potential improvements
- **Pure premium explained:** intuitively, the cost of the vehicle is a big factor in determining insurance premiums because pricier cars can lead to higher costs in case of an accident. Additionally, the age of the vehicle and the driver matter— younger and older drivers are more likely to have accidents, and older vehicles might have more issues, increasing the likelihood of accidents.



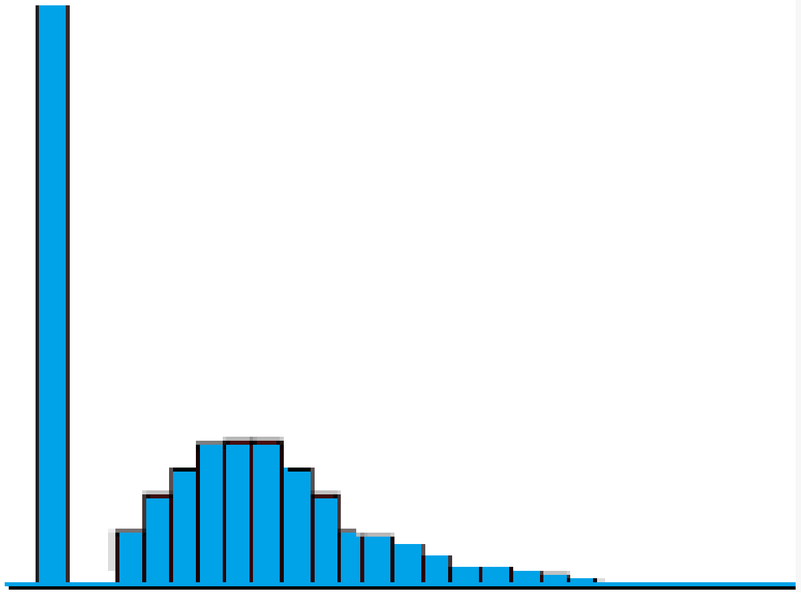


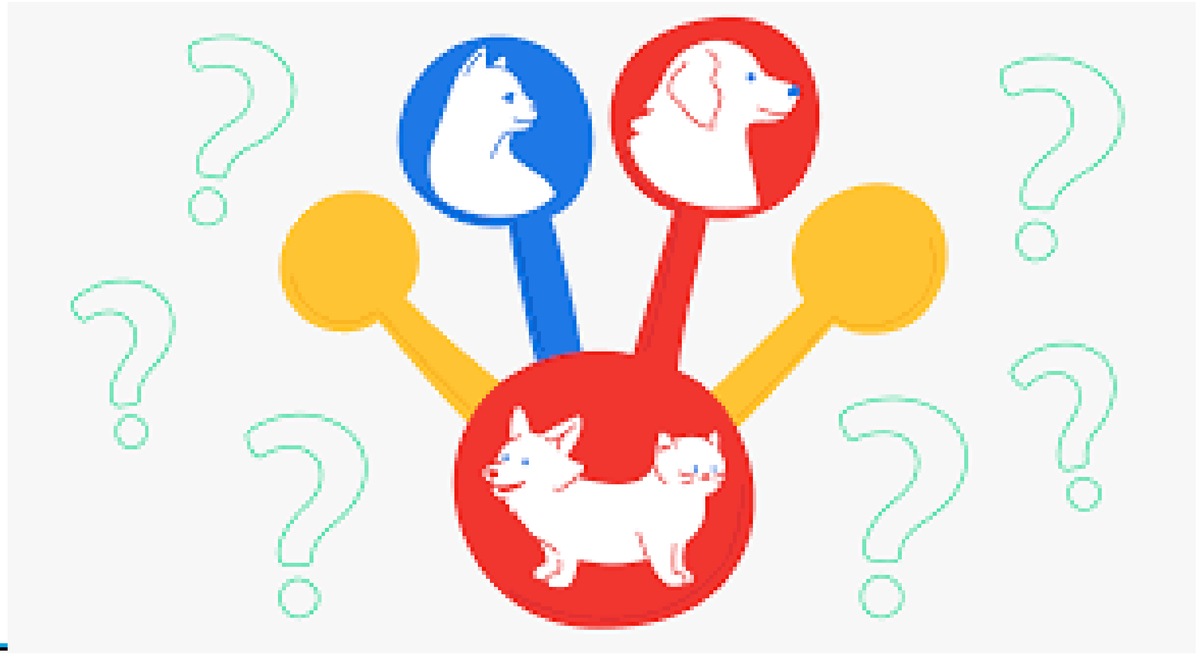
Methods

- **Random Forest:**
  - **Oversampling & Under-sampling:** Methods to balance the dataset.
  - **Two-Part Implementation:** With and without separate parts for model training.
- **Decision Tree**
- **Zero-Inflated Models**
  - Poisson Regression
  - Negative Binomial Regression
  - Zero-Inflated Normal Regression
- **Tobit Regression**
- **Hurdle Regression:** A two-component model combining:
  - Logistic Regression for non-zero cost occurrence.
  - Gamma Regression for positive responses (cost amount among non-zero observations).
- **LightGBM:** A gradient boosting framework that uses tree-based learning algorithms.
- **XGBoost:** Another efficient and scalable implementation of gradient boosting.

Final Models

- Tweedie with Random Search
- CatBoost with Grid Search
- CatBoost with Tweedie with Grid Search





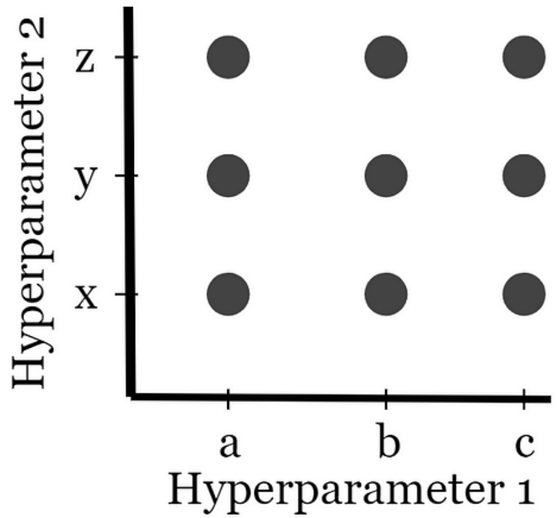
Grid Search & Random Search

- Definition: Methods that explore a variety of parameter values within the model.
- Purpose: Identifying the best parameters based on training data with **cross validation** (model assessment).

Grid Search

Pseudocode

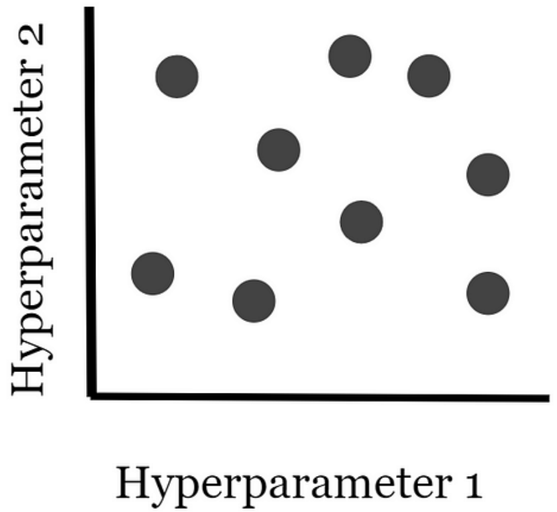
```
Hyperparameter_One = [a, b, c]
Hyperparameter_Two = [x, y, z]
```



Random Search

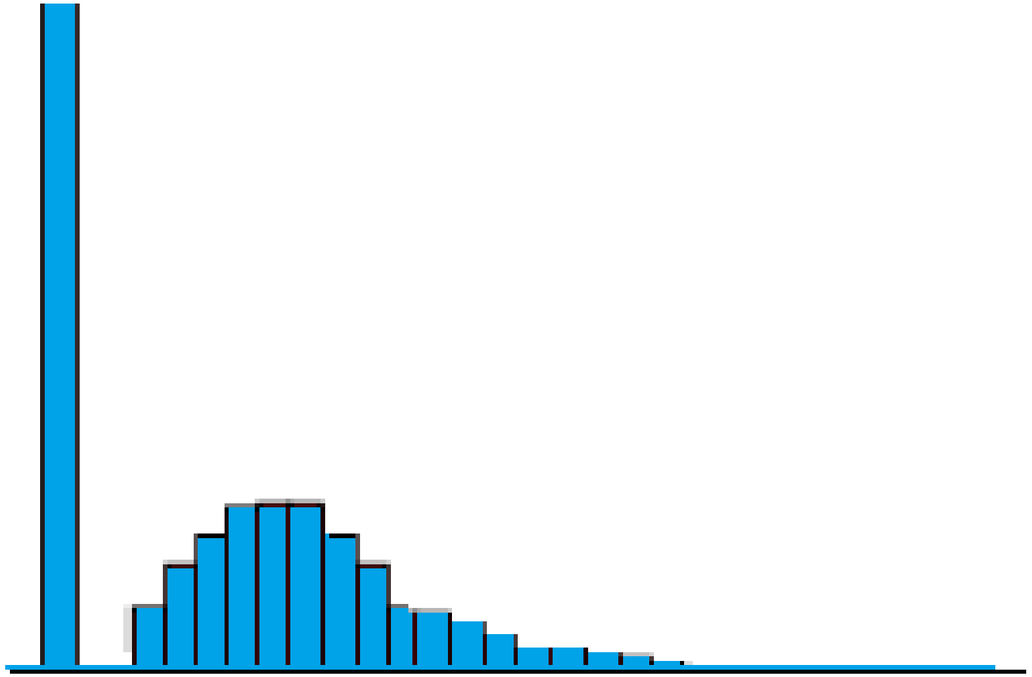
Pseudocode

```
Hyperparameter_One = random.num(range)
Hyperparameter_Two = random.num(range)
```



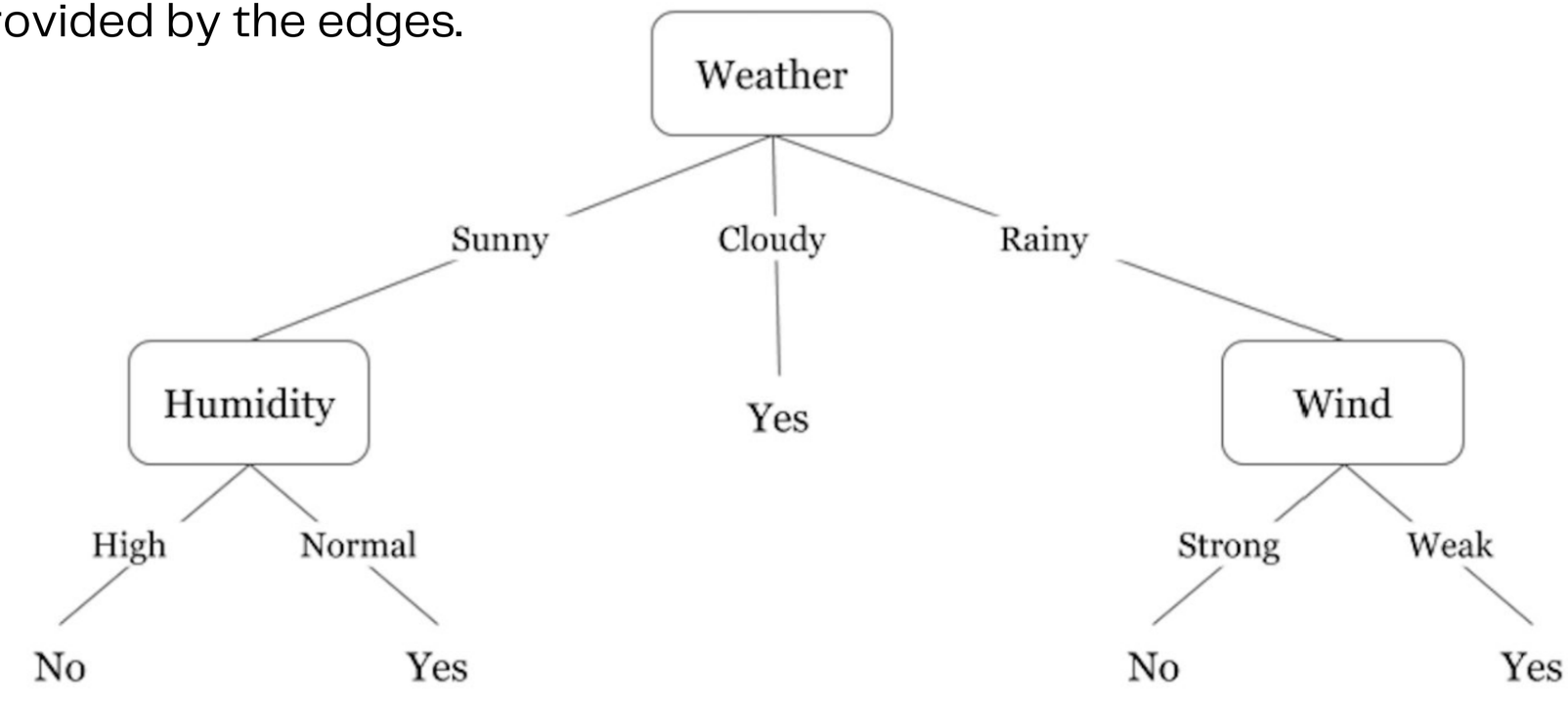
Tweedie Model with Random Search

- Generalized linear model with Tweedie distribution
  - Description: The outcome variable encompasses lots of zero values and is continuous.
  - Application: Ideal for predicting claim costs.
- Utilized the sklearn library in Python for model implementation.
- Result: The predictions seemed reasonable and did well in prediction



Gradient Boosting

- Sequentially adding decision trees, each tree corrects the errors of the previous ones.
- Decision trees:
  - A tree-like model of decisions and potential outcomes by splitting the dataset based on criteria.
  - Trees are built by recursively evaluating different explanatory variables and using the one at each node that best splits the data and maximizes the R squared.
  - To read a tree, we start from the root node and go to the next node based on the options provided by the edges.



Catboost Model with Grid Search

CatBoost Model with grid search stands out for its ability to handle unbalanced data, efficiently process mixed data types, and optimize for high performance, making it a powerful tool for predicting insurance claim costs.

Advantages:

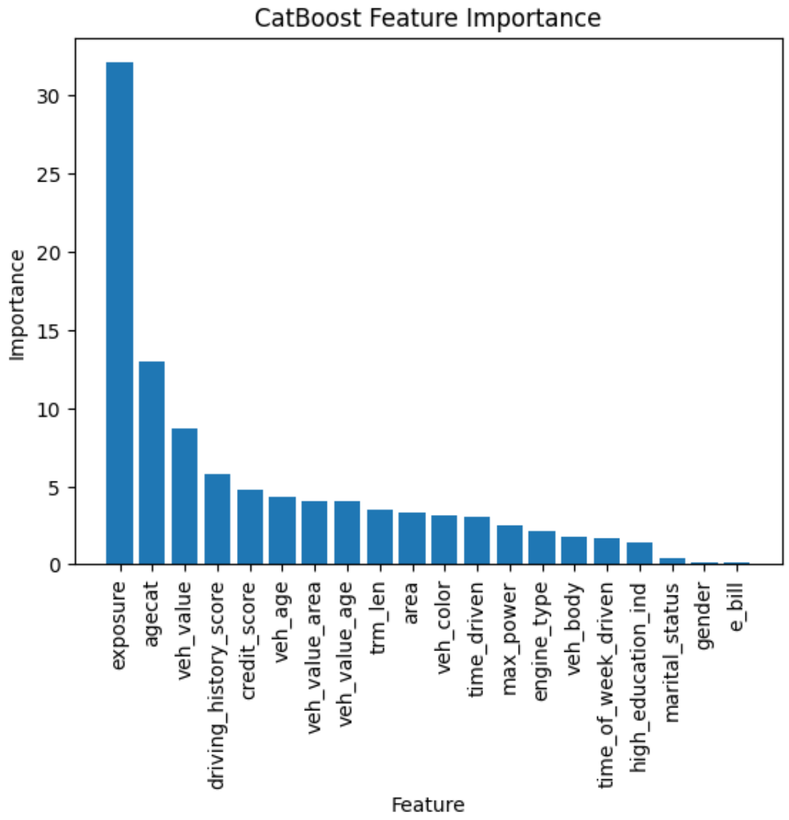
- Unbalanced Data Handling:
  - **Zero-Inflation Management:** high number of zero claims.
  - **Overfitting Prevention:** avoid overfitting in unbalanced datasets.
- Categorical Data Efficiency:
  - Effectively handles categorical data (e.g., vehicle type, driving area) without extensive pre-processing.
- Feature Selection:
  - Provides feature importance scores.
- Hyperparameter Optimization via Grid Search:
  - Explores hyperparameters to find the optimal model setup using **negative mean square errors** as the metric.
- **Gradient boosting**: Sequentially adding decision trees, each tree corrects the errors of the previous ones.

Results:

- The predictions seemed reasonable and did well in prediction.

Catboost with Tweedie Loss Function

- After looking at the performance from Catboost and Tweedie, we decided it would be a good approach to combine them as they both had individual strengths
- Grid Search
  - We used the same approach to find the best parameters for the final ensemble trees
- Feature importance
  - After building the initial model, we looked at feature importance to find the top 10 features to simplify our model.
  - Ultimately, it performed worse when removing the features so we kept the original variables
- Results
  - Did worse than just using Catboost



Strengths & Limitations

- Strengths
  - Categorical variable handling with Catboost
  - Tweedie helps capture outliers where extreme values occur in the data (right-skewed)
- Limitations
  - Imbalanced dataset
    - A dataset with a large number of claim costs equals zero
  - Simplification of predicting Claim Cost vs Frequency and Severity
    - Modeling for just Claim Cost made it harder to understand how specific variables influenced the model
    - Splitting the model into two, one for predicting frequency of claims, and another for predicting severity of claims may have been a better approach to interpret significance of different variables

THANK YOU

For Your Attention and Time