Predicting Customer Churn

Noah Armsworthy, October 31, 2024

# Introduction

Insurance provides security by covering unexpected expenses. Insurance companies rely on their customers for the capital to provide this support. Predicting customer churn can help companies implement strategies to retain customers at critical points. The dataset used for predicting customer churn consists of 6000 sample rows of the following information: ID, Age, Annual Premium, Tenure, Province, Marital Status, Policy Type, Claim Frequency, Customer Satisfaction, and Churn.

# Methodology

## Data Exploration

Several issues were immediately noticeable with the data:

* Unrealistic age values (13 to 125 years), indicating likely data entry errors.
* Significant outliers in Annual Premium, with an improbable minimum of 0.
* Approximately 10% missing values in Age, Annual Premium, and Tenure.
* A 2:1 class imbalance favoring the non-churn label.

### Outliers

* **Age:** Values below the legal adulthood age of 18 and above a generous likely old age of 105 were adjusted based on more realistic records associated with the same customer ID.
* **Annual Premium:** These extreme values did not seem to relate to a specific policy. There could be situations where people are given a great deal at $0 premium while others have factors influencing a much higher yearly rate. Some records had duplicates with more realistic values, but without more information from the data provider, the extreme values were retained.

### Continuous Feature Insights

#### A screenshot of a graph Description automatically generatedA group of blue bars with white text Description automatically generated

Figure . Feature Correlation

Figure . Feature Distributions

* **Normality:** The continuous features were not normally distributed, which informed later preprocessing decisions. (*Figure 2*)
* **Correlation:** No strong associations between features or the target variable were found. (*Figure 1*)

## Data Cleaning/Preprocessing

* **Missing Data:** Imputed with the most frequent value, as dropping records would result in too much data loss. A good rule of them is to only drop records with missing data if they make up less than 5% of the sample size.
* **Duplicate Data:** Exact duplicates were removed to strive for equal weighting for each customer.
* **Near Duplicates:** Records with the same customer ID but different features were kept, as removing them would result in a significant loss of samples and would be difficult to select without more information from the data provider.
* **Normalization:** Numerical features were scaled to a 0-1 range to prevent larger values from skewing model outcomes.
* **Feature Selection:** Customer ID was dropped as it is just the indexing value of the data.
* **Categorical Encoding:** One-hot encoding was used to transform categorical features into separate columns of Boolean variables. (*Figure 3*)

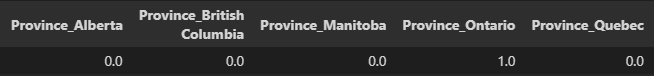


Figure . One-hot Encoded Provinces

## Model Selection

An **80/20** train-test split was used to evaluate several models on this binary classification task. Results are as follows:

Table . Model Performances with Top Two Performers Highlighted

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | | Recall | | F1-Score | | ROC  AUC |
| No Churn | Churn | No Churn | Churn | No Churn | Churn |
| Logistic Regression | 67% | 67% | 0% | 100% | 0% | 80% | 0% | 55% |
| SVM | 67% | 67% | 0% | 100% | 0% | 80% | 0% | 55% |
| Decision Tree Classifier | 90% | 91% | 87% | 94% | 82% | 93% | 84% | 87% |
| K-Nearest Neighbor | 64% | 69% | 43% | 84% | 25% | 76% | 31% | 57% |
| Naive Bayes | 66% | 67% | 40% | 98% | 3% | 80% | 5% | 53% |
| Random Forest Classifier | 93% | 90% | 100% | 100% | 78% | 95% | 88% | 98% |
| Gradient Boosting Classifier | 73% | 71% | 87% | 98% | 21% | 83% | 34% | 84% |

### The top performers were the Decision Tree and Random Forest models. Both tend to perform well on imbalanced datasets but may have overfit.

### Cross-Validation

To select the final model, stratified cross-validation was used to account for class imbalance. The Decision Tree model outperformed Random Forest, achieving an **83%** recall for the churn class, compared to **75%** for Random Forest.

### Hyper Parameter Tuning

GridSearchCV was employed to fine-tune the parameters of the Decision Tree Classifier to optimize the true labels **f1** score. The best parameters identified were: **class\_weight:** {0: 1, 1: 1.5}, **criterion:** 'gini', **max\_depth:** None, **max\_features:** None, **min\_samples\_leaf:** 1, and **min\_samples\_split:** 2. The best cross-validation score achieved during this tuning process was **0.894**, which did not demonstrate a substantial improvement the model with default parameters.

# Results

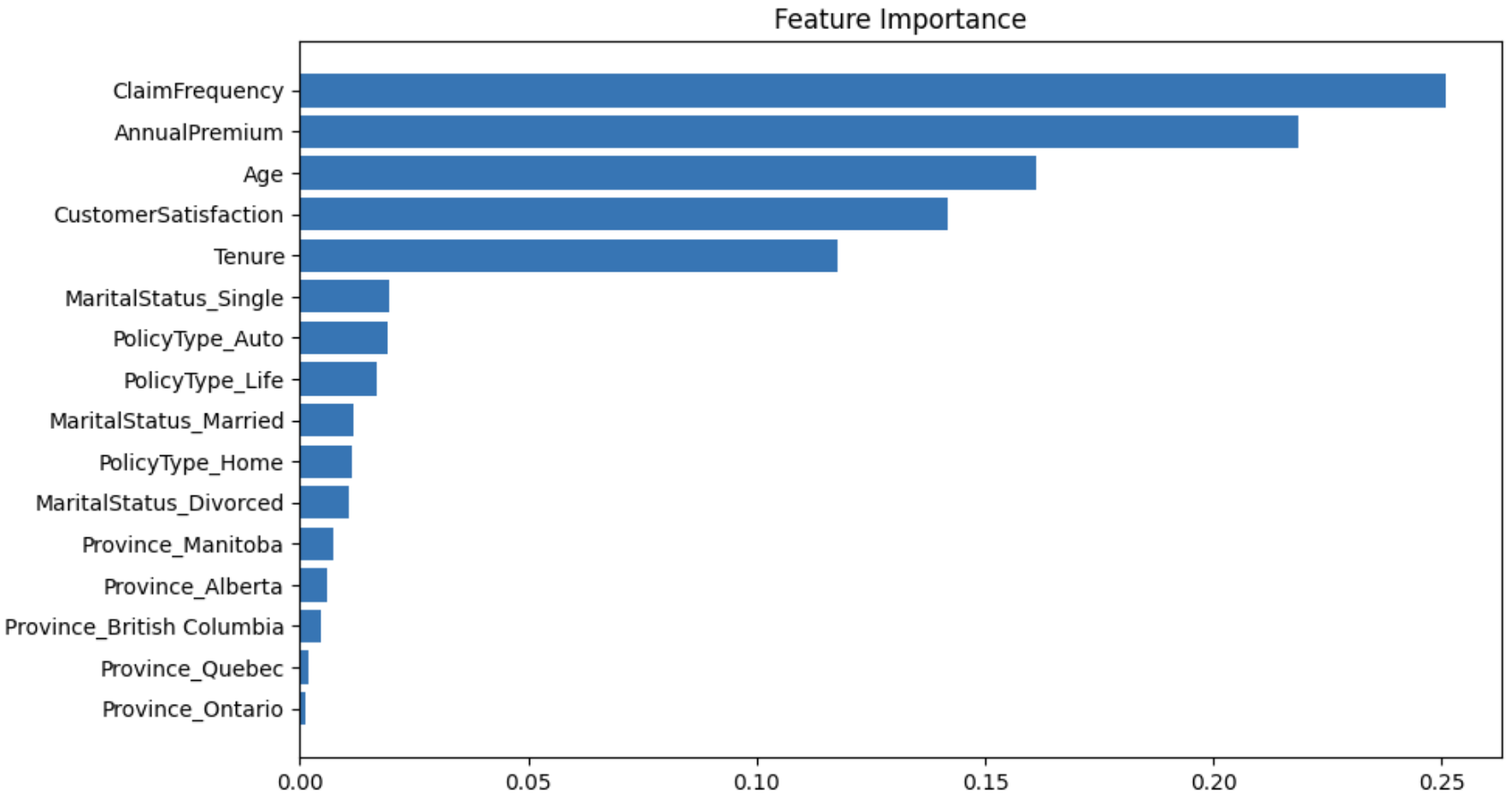


Figure . Decision Tree Feature Importances

* **Model Performance:** The Decision Tree was selected due to its higher recall on churn.
  + The final metrics were **92%** Accuracy, **88%** recall and **88%** f1 score on the test set.
  + The top features that influence predictions were revealed to be: ClaimFrequency, AnnualPremium, Age, CustomerSatisfaction, and Tenure. (*Figure 4*) Testing was done using only these features but it did not improve the performance of the model.
* **Challenges:** Imbalanced data, small amount of data, duplicate records, and issues with default cross-validation were notable obstacles.
* **Findings & Recommendations:**
  + Consult with the data provider about oddities in the data such as near duplicates records, and extreme annual premium values.
  + Consider over/under-sampling if future data has different class distributions.
  + Post-pruning for the Decision Tree may improve performance and interpretability.
  + Perform testing on a larger variety of hyper parameters.

# Conclusion

The Decision Tree Classifier demonstrated the best performance in predicting customer churn, especially in identifying potential churners. Future work could focus on addressing data imbalance and refining the model with additional tuning or pruning.

# References

1. CareerFoundry, "How to Find Outliers," *CareerFoundry*, [Online]. Available: <https://careerfoundry.com/en/blog/data-analytics/how-to-find-outliers/>. [Accessed: 29-Oct-2024].
2. R. A. Boyle, and D. M. S. Bowman, "Choosing the right statistical test: A practical guide for ecologists," *Methods in Ecology and Evolution*, vol. 1, no. 1, pp. 3–14, Jan. 2010. [Online]. Available: <https://besjournals.onlinelibrary.wiley.com/doi/10.1111/j.2041-210X.2009.00001.x>. [Accessed: 28-Oct-2024].
3. J. Brownlee, "How to Use One-Hot Encoding for Categorical Data," *Machine Learning Mastery*, [Online]. Available: <https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/>. [Accessed: 27-Oct-2024].
4. Scikit-learn, "Preprocessing Data," *Scikit-learn*, [Online]. Available: <https://scikit-learn.org/stable/modules/preprocessing.html>. [Accessed: 27-Oct-2024].
5. O. Nalcin, "StandardScaler vs MinMaxScaler vs RobustScaler," *Medium*, 2022. [Online]. Available: <https://medium.com/@onersarpnalcin/standardscaler-vs-minmaxscaler-vs-robustscaler-which-one-to-use-for-your-next-ml-project-ae5b44f571b9>. [Accessed: 28-Oct-2024].
6. J. Brownlee, "Cost-Sensitive Decision Trees for Imbalanced Classification," *Machine Learning Mastery*, [Online]. Available: <https://machinelearningmastery.com/cost-sensitive-decision-trees-for-imbalanced-classification/>. [Accessed: 29-Oct-2024].
7. Stack Exchange, "Training a Decision Tree Against Unbalanced Data," *Cross Validated*, [Online]. Available: <https://stats.stackexchange.com/questions/28029/training-a-decision-tree-against-unbalanced-data>. [Accessed: 29-Oct-2024].
8. Stack Exchange, "Why Is Accuracy Not the Best Measure for Assessing Classification Models?," *Cross Validated*, [Online]. Available: <https://stats.stackexchange.com/questions/312780/why-is-accuracy-not-the-best-measure-for-assessing-classification-models>. [Accessed: 29-Oct-2024].
9. Scikitlearn, "sklearn.tree.DecisionTreeClassifier," *Scikit-learn*, [Online]. Available: <https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html>. [Accessed: 30-Oct-2024].