

A PROJECT ON
“Health Insurance Cross Sell Prediction”

SUBMITTED IN
PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE COURSE OF
DIPLOMA IN BIG DATA ANALYSIS



SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY

Phase 2, Hinjewadi Rajiv Gandhi
Infotech Park, Hinjewadi, Pune,
Maharashtra 411057

SUBMITTED BY:

Harish Pawar (70190)

UNDER THE GUIDENCE OF:

Mrs. Manisha Hingne

Faculty Member

Sunbeam Institute of Information Technology, PUNE.



CERTIFICATE

This is to certify that the project work under the title 'Health Insurance Cross Sell Prediction' is done by Harish Pawar in partial fulfillment of the requirement for award of Diploma in Big Data Analytics Course.

Mrs. Manisha Hingne
(Project Guide)

Mrs. Pradnya Dindorkar
(Course Co-ordinator)

Date: 13/03/2023

ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards Mr. Nitin Kudale (Center Coordinator, SIIT, Pune) and Mrs. Pradnya Dindorkar (Course Coordinator, SIIT, Pune) and Project Guide Mrs. Manisha Hingne.

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

Harish Pawar

DBDA September 2022

SIIT, Pune

Table of contents:

1. Introduction

1.1. Introduction and Objectives

1.2. Dataset Information

1.3. Tools Used

2. Problem Definition and Algorithm

2.1 Problem Definition

2.2 Algorithm Definition

3. Data cleaning

4. Exploratory Data Analysis

5. Feature Engineering

5.1 Encoding Features

5.2 Feature Scaling

5.3 Balancing dataset

6. Results and Discussion

7. Deployment

8. Conclusion

9. References

1. Introduction

1.1 Introduction and Objectives

The amount of data we create has increased a lot due to smartphones and smart devices. This data can help financial companies understand their customers better and make better decisions. They can use machine learning to predict which customers might stop using their service or which customers they can offer new products to. This helps the company make more money and keep their customers happy.

The project aims to use machine learning to predict which health insurance policyholders are likely to be interested in buying vehicle insurance as a cross-selling strategy. The publicly available dataset with 12 attributes, including demographic and vehicle information. The study includes a literature review of cross-selling strategies in various sectors, particularly financial industries, and compares the performance of different machine learning algorithms before and after applying dimension reduction methods on an unbalanced target dataset. Overall, the study aims to provide insights into effective cross-selling techniques and demonstrate the impact of dimension reduction methods on model performance.

1.2 Dataset Information

1. id: Unique ID for the customer
2. Gender: Gender of the customer
3. Age: Age of the customer
4. Driving License: 0: Customer does not have DL, 1: Customer already has DL
5. Region Code: Unique code for the region of the customer
6. Previously Insured: 1 : Customer already has Vehicle Insurance, 0 : Customer doesn't have Vehicle Insurance
7. Vehicle Age: Age of the Vehicle
8. Vehicle Damage: 1: Customer got his/her vehicle damaged in the past. 0: Customer didn't get his/her vehicle damaged in the past.
9. Annual Premium: The amount customer needs to pay as premium in the year
10. Policy Sales Channel: Anonymized Code for the channel of outreaching to the customer i.e. Different Agents, Over Mail, Over Phone, In Person, etc.
11. Vintage: Number of Days, Customer has been associated with the company
12. Response: 1: Customer is interested, 0: Customer is not interested

1.3 Tools Used:

1. Programming Languages: Python is used for machine learning part. HTML and CSS were used for user interface.
2. Data Analysis and Visualization Tools: Tools such as Pandas, NumPy, and Matplotlib are used for data analysis and visualization.
3. Machine Learning Libraries: Popular machine learning libraries such as scikit-learn is used to develop and train the machine learning models for used car price prediction From scikit-learn we have used algorithms as Logistic Regression, SVM, Naïve Bayes, Decision Tree, Random Forrest, XGBoostClassifier.
4. Cloud Computing Platforms; Cloud computing platforms Amazon Web Services (AWS) is used to run the machine learning model on scalable infrastructure
5. Integrated Development Environments (IDES): IDEs such as PyCharm, Jupyter Notebook used to write, test, and debug the code for the used car price prediction machine learning project. These tools provide a range of features for efficient and effective coding.

2. Problem definition and algorithm

2.1 Problem Definition:

Our client, an insurance company, wants to predict if their customers who have health insurance policies with them would also be interested in purchasing vehicle insurance. Insurance policies provide compensation for loss, damage, illness, or death in return for regular premium payments. The concept of probabilities is used to calculate the risk of compensation pay-outs based on the number of customers paying premiums. Vehicle insurance is similar to health insurance in terms of premium payments and compensation for accidents. By predicting customer interest in vehicle insurance, the company can optimize their communication strategy and potentially increase revenue. We will be using the "uncleaned.csv" database, which contains around 500000 values and 12 columns, to analyse and process the data, balance the classes, and build machine learning classification models to achieve our goal.

2.2 Algorithm Definition / Modelling:

In the process of identifying target customers for a cross-selling campaign, various machine learning algorithms were utilized, including Naïve Bayes, Logistic Regression, K-Nearest Neighbors, Random Forest, XG Boost. Both parametric and non-parametric methods were employed to ensure the reliability and comprehensiveness of the classification results. The F1 score, which calculates metrics for each class attribute and weighs them by support, was used to evaluate the performance of the models. In addition, a 5-fold cross-validation score was used to measure the accuracy metrics.

Logistic Regression:

Logistic Regression is a statistical method used to model and analyse binary outcomes (0 or 1), or more generally, categorical outcomes with two or more possible values. It is a type of regression analysis that is commonly used in machine learning, data science, and statistics.

Logistic regression has several advantages over other classification algorithms, including simplicity, interpretability, and the ability to handle both categorical and continuous predictor variables. However, it also has some limitations, such as the assumption of linearity between the predictors and the logit function, and the potential for overfitting in high-dimensional datasets.

Naïve Bayes:

Naive Bayes is a probabilistic machine learning algorithm used for classification tasks. It's based on Bayes' theorem, which describes the probability of an event based on prior knowledge or

evidence. In the case of Naive Bayes, the algorithm assumes that the features of the data are independent of each other, which is a simplification known as the "naive" assumption.

There are several types of Naive Bayes classifiers, including:

1. Gaussian Naive Bayes: used for continuous data that follows a Gaussian distribution.
2. Multinomial Naive Bayes: used for discrete data with a limited number of possible values, such as text classification.
3. Bernoulli Naive Bayes: used for binary data where each feature can take on only two possible values.

Naive Bayes is a simple and fast algorithm that performs well on large datasets with high dimensions. However, its performance can be affected by the presence of correlated features or irrelevant features.

K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a popular machine learning algorithm used for classification and regression tasks. It's a non-parametric algorithm, which means it doesn't make any assumptions about the underlying distribution of the data.

In KNN, the algorithm assigns a label or predicts a target value for an instance by looking at the K nearest Neighbors in the training set. The value of K is a hyperparameter that can be tuned to improve the performance of the algorithm.

However, KNN has some limitations. It can be slow and memory-intensive when dealing with large datasets or high-dimensional data, and it's sensitive to the choice of distance metric used to measure the similarity between instances. Additionally, the performance of KNN can suffer when there are irrelevant or noisy features in the data.

Decision Tree:

A decision tree is a type of supervised learning algorithm used in data mining and machine learning. It is a graphical representation of all possible solutions to a decision based on certain conditions or events. The tree is made up of nodes and branches, where each node represents a decision or a test on a feature, and the branches represent the outcomes of that decision or test.

The root node is the first node in the tree, which represents the entire dataset. The tree then splits into smaller branches, each representing a subset of the data, based on the selected feature and its values. The splitting process continues until the terminal nodes, also known as the leaves, are reached. The leaves represent the final decision or outcome of the tree.

However, decision trees can also be prone to overfitting, especially when the tree is too complex and deep. To avoid overfitting, techniques such as pruning and regularization can be used.

GridSearch:

GridSearch is a hyperparameter tuning technique used in machine learning to find the optimal set of hyperparameters for a given model. Hyperparameters are parameters that cannot be learned during the training process but must be set before training begins. Examples of hyperparameters include the learning rate, regularization strength, number of hidden layers, and number of nodes in each layer.

GridSearch works by creating a grid of possible hyperparameter values and then exhaustively searching through that grid to find the best combination of hyperparameters. For example, if we are tuning the learning rate and regularization strength for a neural network, we might create a grid with three possible learning rates and three possible regularization strengths. We would then train a model with each combination of hyperparameters and select the combination that performs best on a validation set.

GridSearch can be computationally expensive because it requires training multiple models with different hyperparameters. However, it is a widely used technique because it can be very effective in finding the best hyperparameters for a given model.

Random Forest:

Random Forest is a machine learning algorithm that is commonly used for both classification and regression tasks. It is an ensemble learning technique that combines multiple decision trees to create a more accurate and robust model.

In a Random Forest, a set of decision trees are trained on random subsets of the data, and the final prediction is based on the average (for regression) or majority vote (for classification) of the predictions of individual trees. The randomness introduced in the algorithm reduces overfitting and increases the model's generalization ability.

Random Forest is a popular algorithm for various applications because of its ability to handle high-dimensional datasets with a large number of features, and it is also robust to missing values and noisy data. Additionally, it provides a measure of feature importance, which can be used to identify the most relevant features in the dataset.

XGBoost:

XGBoost (Extreme Gradient Boosting) is a popular open-source machine learning library used for building efficient and scalable machine learning models. It is a type of ensemble model that combines multiple weak learners (decision trees) to create a more powerful model. It was created by Tianqi Chen and initially maintained by the Distributed (Deep) Machine Learning Community (DMLC) group.

XGBoost is widely used in data science competitions and has become one of the most popular machine learning libraries due to its high performance, scalability, and accuracy. It is also relatively easy to use, making it accessible to both novice and expert machine learning practitioners.

Some key features of XGBoost include:

1. Regularization: XGBoost includes both L1 and L2 regularization to prevent overfitting.
2. Gradient boosting: XGBoost uses gradient boosting to iteratively train weak learners, with each iteration minimizing the loss function.
3. Cross-validation: XGBoost includes built-in cross-validation capabilities to evaluate model performance and prevent overfitting.
4. Parallel processing: XGBoost can be run in parallel on multiple cores, making it much faster than traditional machine learning models.

Overall, XGBoost is a powerful and versatile machine learning library that can be used for a wide range of tasks, including classification, regression, and ranking problems.

3.Data Cleaning

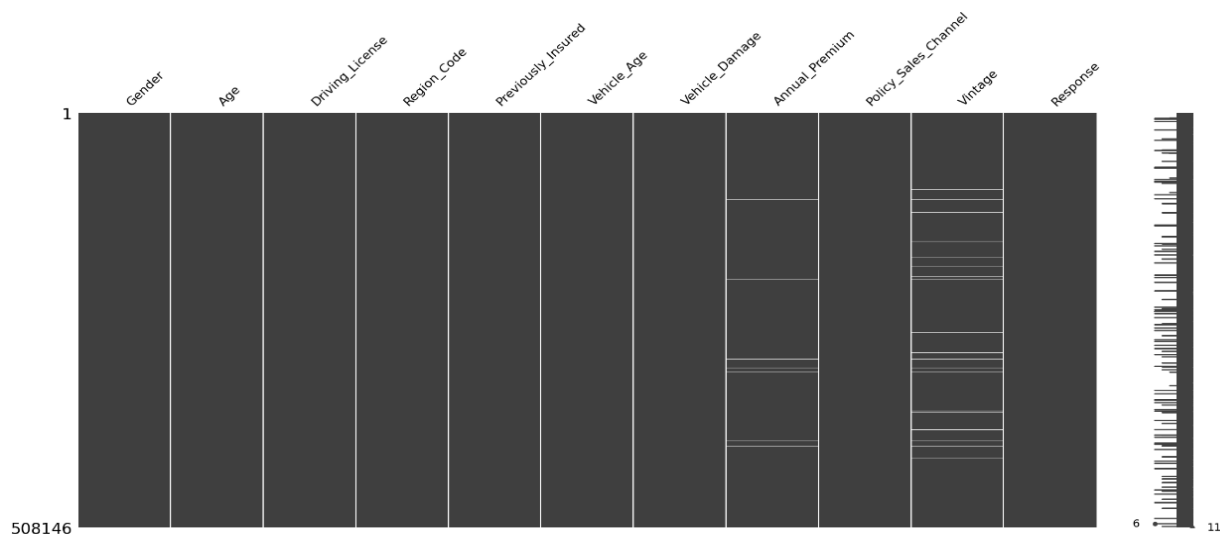


Fig 1. Missing values

After Checking NA values, there are missing values in several columns of our dataset. The columns with missing values are:

- Driving License: 63 missing values
- Region Code: 109 missing values
- Vehicle Age: 102 missing values
- Annual Premium: 5676 missing values
- Vintage: 12937 missing values

It's important to handle missing values appropriately before proceeding with any analysis or modelling tasks. Depending on the type and number of missing values, different techniques can be used to handle them, such as imputation or deletion of rows or columns. we can use panda's library in python to handle the missing values.

Handling Missing Values:

- **Mean**-It is preferred if data is numeric and not skewed.
- **Median**-It is preferred if data is numeric and skewed.
- **Mode**-It is preferred if the data is a string(object) or numeric

We also tried to impute the data by using a machine learning model but the accuracy of the model was not at all good thus we rejected it. And also, the missing value data was not more than 3% of total data hence we filled the missing values in annual premium and vintage column with their median and mean respectively.

4.Exploratory Data Analysis

The data set includes several numeric variables such as Age, Annual Premium, Region code, Policy sales channel, Vintage, and Response. To better understand the distribution and relationship among these variables, various visualization techniques have been used, including pair plots, heatmaps with Pearson correlation, and histograms.

To visualize the distribution of the continuous variables, bar plots have been created. These bar plots show the frequency of occurrence of each value or range of values for each variable. This helps to identify any outliers or unusual patterns in the data set.

The following Fig. 2 indicates the bar plots which shows the distribution of continuous variables:

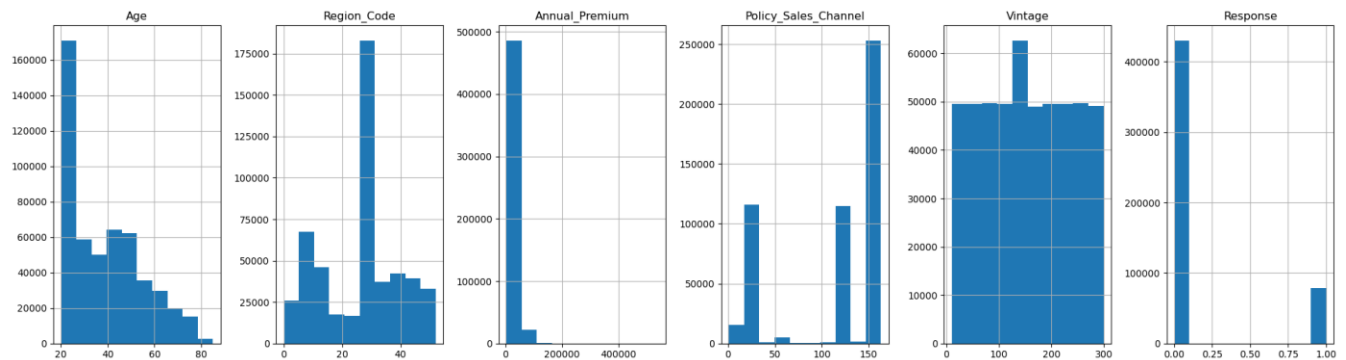


Fig 2. Numerical variables distribution

According to the bar plot figure, Age attribute of the policyholders demonstrate that younger customers are more likely to respond positively for buying vehicle insurance provided by the company so it can be clearly said that young customers of the insurance company are much more appropriate for applying cross-selling proposals. The health insurance policyholders who are roughly older than 20 are not interested in buying vehicle insurance, especially the customers who are older than 60. Histogram plot of age attribute in Figure shows the distribution and there is a right-skewed distribution that the frequency of young policyholders samples are significantly more than older policyholders in the sample of the dataset.

Based on the bar plot in the figure, it appears that younger policyholders are more likely to respond positively to the company's vehicle insurance offers, while older policyholders, particularly those over 40, are less interested. This suggests that the company may be more successful in applying cross-selling strategies to younger customers.

The histogram plot in the figure shows the distribution of the Age attribute, which is skewed towards the right. This means that there are more samples of younger policyholders in the dataset than older ones. Therefore, it can be inferred that the majority of policyholders in the dataset are younger individuals, and there are relatively fewer older individuals.

Based on the distribution of region codes, it appears that the most common region codes fall within the range of 25-35. However, the distribution of region codes is not normal, as it exhibits

negative skewness and kurtosis. It is also worth noting that the minimum region code is 0, which may be an invalid code.

The high variation in the region codes, indicated by a coefficient of variation of 0.50, suggests that there is a wide range of region codes and that they are not evenly distributed. This may indicate that certain regions are overrepresented or underrepresented in the data.

The amount customer needs to pay as premium in the year attribute that are defined as Annual Premium demonstrates that the policyholders pay around 2500 to 50000 are willing to buy more ensured insurance services by the company through looking at the pair plot. Annual Premium attribute shows also right-skewed distribution as well. Most of the randomly selected samples pay around 30000 to 50000 regularly to the company

The Vintage attribute represents the number of days that customers have been associated with the insurance company. The maximum number of days observed is 299, and the minimum is 10. The skewness of the distribution is close to 0, indicating that the data is approximately symmetric. However, the coefficient of variation is greater than 0.50, suggesting that there is a high degree of variability in the number of days spent with the company. Additionally, the negative kurtosis value suggests that the tails of the distribution are relatively light.

Despite the uniform distribution of days spent with the company, the median value for both groups is approximately 154, indicating that half of the customers have been associated with the company for less than 154 days, and half have been associated for more than 154 days.

the Policy Sales Channel appears to be heavily right-skewed, indicating that a large proportion of customers may have a similar policy sales channel while a relatively small proportion may have unique channels. The distribution is relatively concentrated between values of 26.0 and 152.0, with a peak around 133.0. Further analysis may be necessary to understand the implications of the policy sales channel for the insurance company and its customers.

Categorical Analysis:

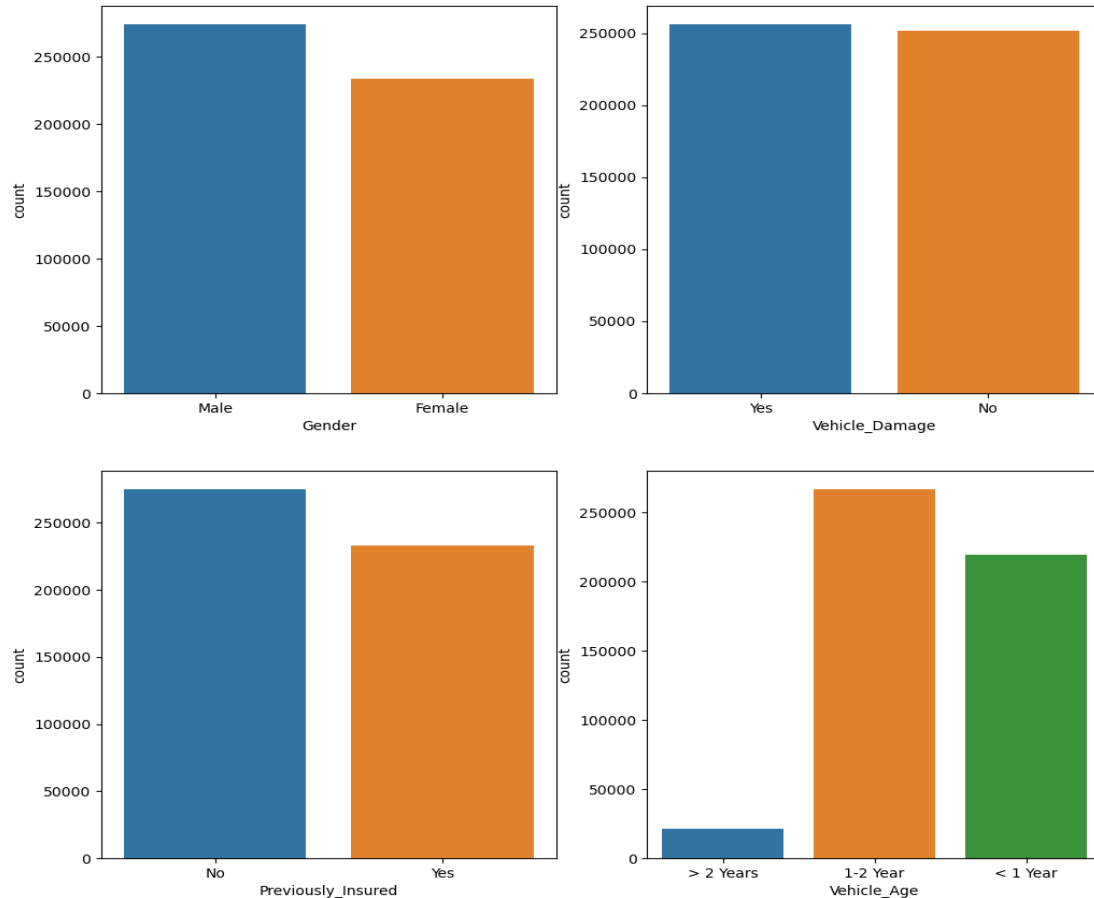


Fig 3. Categorical variables count plots

The dataset consists of seven discrete attributes named Gender, Vehicle Age, Driving License, Previously Insured, Region Code and Policy Sales Channel and the distribution of these attributes are analysed based on the Response feature.

We can see that our dataset is heavily imbalanced in the response field. With most of the values in class 0. We can see balanced Gender field. Certainly, most of our customers must have driving license in order to drive a car. Few of our customers don't have DL, maybe because they're underage, waiting to be eligible for a DL. A balanced Previously insured field, slightly larger portion of our customers don't have vehicle insurance. They are our potential clients. Most of our clients own new cars (<2 years of Vehicle Age). So, our further findings will mostly reflect for customers who own new cars, it may not be true for old car (> 2 years).

Bivariate Analysis:

The dataset consists of seven discrete attributes named Gender, Age, Vehicle Age, Driving License, Previously Insured.

For Gender attribute, there are 64.8% male and 35.2% female samples who are interested in buying the insurance and response is 1. Males are more likely to buy Vehicle Insurance. we can see in fig 4 that 64% of males are interested in buying the insurance.

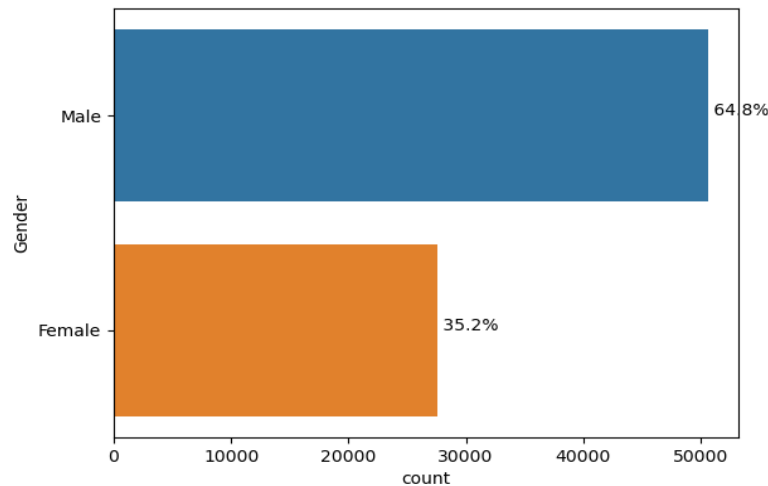


Fig 4. Response according to Gender

Based on the pair plot fig 5, it can be observed that the age attribute of policyholders has a significant impact on their likelihood of responding positively to the company's vehicle insurance offer. Specifically, middle age customers are more likely to be receptive to purchasing the insurance policy, indicating that they may be a more suitable target for cross-selling proposals. Adults are more likely to buy vehicle insurance.

However, the histogram plot of the age attribute in the figure shows a right-skewed distribution, with a significantly higher frequency of younger policyholders compared to older policyholders in the dataset. In fact, health insurance than 20 policyholders younger and those older than 60, are generally not interested in purchasing vehicle insurance from the company. It is right that young people account for less than 30% of customers who want insurance.

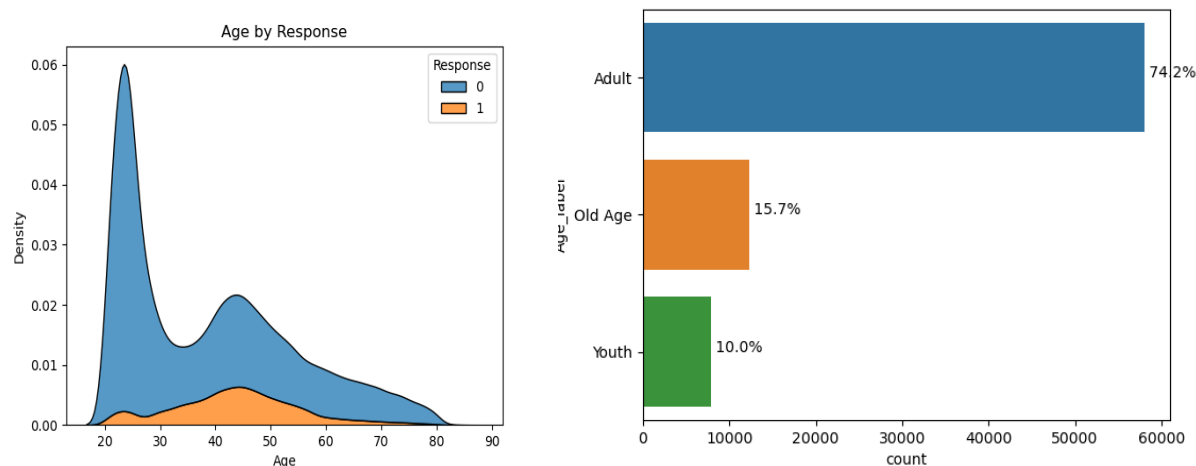


Fig 5. Response according to Age

The feature of vehicle age can provide valuable insights into the preferences of policyholders. Specifically, the data indicates that only a small percentage (8.8%) of customers whose cars are older than two years are likely to purchase vehicle insurance from the company.

On the other hand, the data shows that a slightly higher percentage (11.8%) of health policyholders with new vehicles are receptive to purchasing vehicle insurance. However, this still suggests that the majority of people are not highly interested in obtaining vehicle insurance when their car is new.

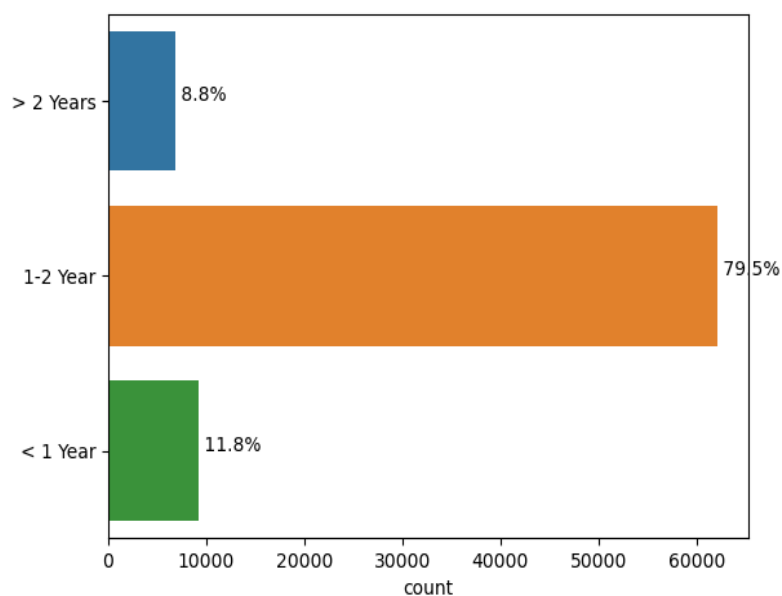


Fig 6. Response according to Vehicle Age

According to the data, an overwhelming majority (98.7%) of customers who own a damaged vehicle have responded positively to the offer of purchasing vehicle insurance. This suggests that customers who already have health insurance and own a damaged car are particularly interested in obtaining vehicle insurance, as illustrated in the fig 7.

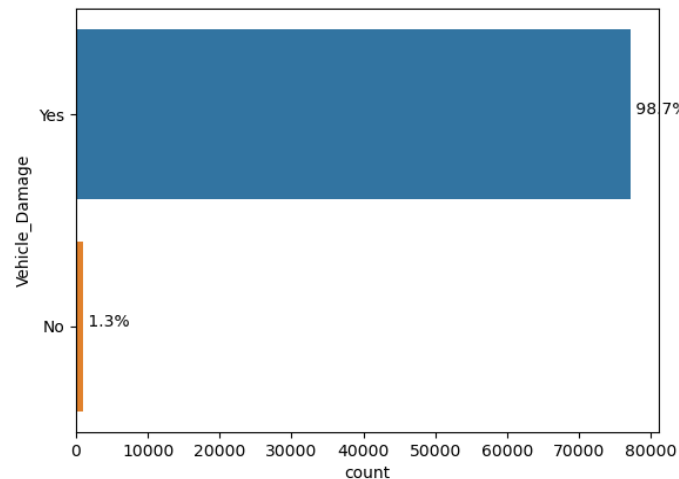


Fig 7. Response According to Vehicle damage

The dataset indicates that only a very small percentage (0.2%) of customers who have previously obtained vehicle insurance have confirmed their interest in purchasing another policy, as shown in the fig 8. This finding is consistent with expectations, as customers who already have vehicle insurance are less likely to respond positively to another offer for the same product.

Furthermore, the analysis reveals that Vehicle Damage is a strong indicator of the success of vehicle insurance cross-selling campaigns. Specifically, 99.8% of customers who responded positively to the offer of purchasing vehicle insurance did not have a previous insurance policy with the company.

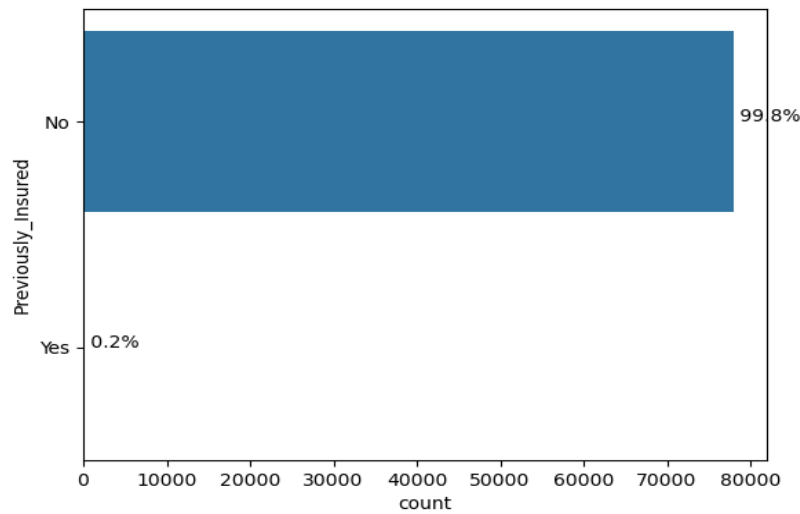


Fig 8. Response According to Previously Insured.

Regarding the Driving License attribute, it is noteworthy that 99% of the samples in the dataset possess a valid driving license, which is a legal requirement for obtaining vehicle insurance. Moreover, the majority of these policyholders have expressed interest in the cross-selling proposal.

The distribution of the Driving License attribute with respect to the Response attribute is shown in the following fig 9. Notably, there are no histogram charts available for customers who do not possess a driving license, as they are not eligible for vehicle insurance.

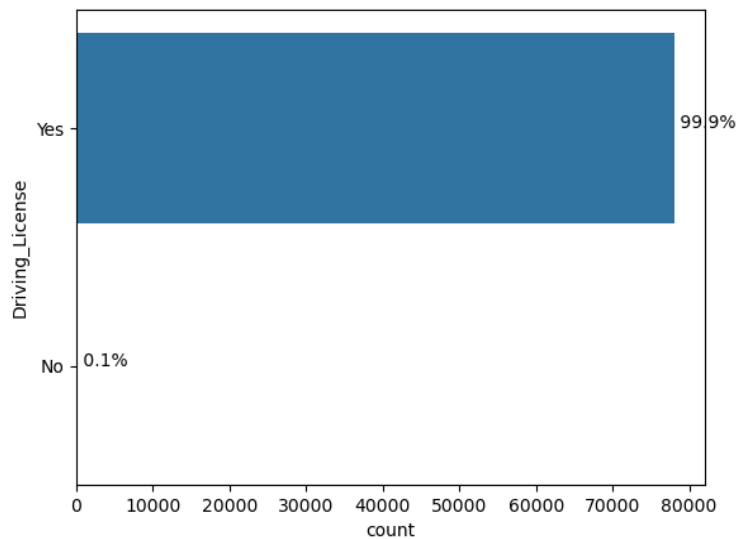


Fig 9. Response According to Driving license

5. Feature Engineering

5.1 Encoding Features

Machine learning algorithms typically require input variables to be numerical, so converting categorical variables to numerical variables is an important step in feature engineering.

Feature encoding is the process of converting categorical variables (also known as nominal variables) into numerical variables that can be used as input for machine learning algorithms. Categorical variables are variables that represent a finite number of distinct categories, such as gender (male or female), education level (high school, college, or graduate), or color (red, blue, or green).

There are several techniques for feature encoding, including:

1. **One-Hot Encoding:** This involves creating a binary vector for each category of a categorical variable. For example, if we have a categorical variable "color" with categories "red", "blue", and "green", we would create three binary variables: "color red", "color blue", and "color green". If an observation falls into the "red" category, the "color red" variable would be set to 1 and the other two variables would be set to 0.

This technique is useful when there are no inherent orderings or rankings among the categories of a variable. It creates a binary vector for each category, which can be used as input to machine learning algorithms.

2. **Ordinal Encoding:** This involves assigning numerical values to the categories of a categorical variable based on their order or rank. This technique is useful when there is a natural ordering or ranking among the categories of a variable. For example, if we have a categorical variable "education level" with categories "high school", "college", and "graduate", we could assign the values 1, 2, and 3, respectively.
3. **Label Encoding:** This involves assigning numerical values to the categories of a categorical variable without any particular order or rank. For example, if we have a categorical variable "color" with categories "red", "blue", and "green", we could assign the values 1, 2, and 3, respectively.

5.2 Feature Scaling

Feature scaling is the process of normalizing the range of values of input features (also called independent variables) used in machine learning models. Feature scaling is typically used when the input features have different ranges or units of measurement, which can affect the performance of certain machine learning algorithms.

There are two common techniques for feature scaling:

1. **Standardization:** This involves transforming the input features such that they have a mean of zero and a standard deviation of one. Standardization preserves the shape of the distribution of the original data, but centres it around zero.

2. Normalization: This involves transforming the input features such that they are scaled to a range between 0 and 1. Normalization maps the input features to the same scale, which can be useful for algorithms that use distance measures, such as k-nearest neighbors and support vector machines.

The choice of scaling technique depends on the specific requirements of the problem being solved and the nature of the data. In general, standardization is a good default choice, but normalization may be more appropriate if the input features have a skewed distribution or if the algorithm being used relies on distance measures.

It's important to note that not all machine learning algorithms require feature scaling. For example, tree-based algorithms like decision trees and random forests do not require feature scaling, as they are insensitive to the scale of the input features.

5.3 Balancing dataset

In order to ensure reliable predictions, having balanced output features is crucial in classification problems. However, the class label distribution of the Response column in the following figure is highly imbalanced, with 84.6% of samples indicating a value of 0 and only 15.4% indicating a value of 1.

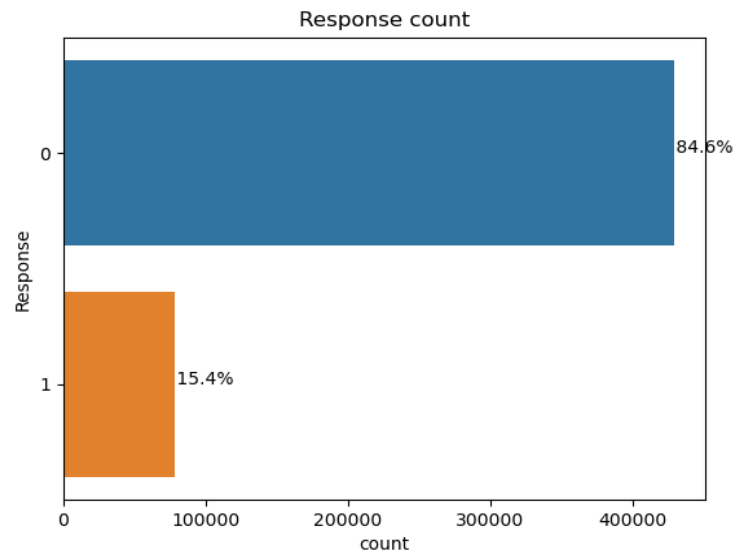


Fig 10. Unbalanced Dataset

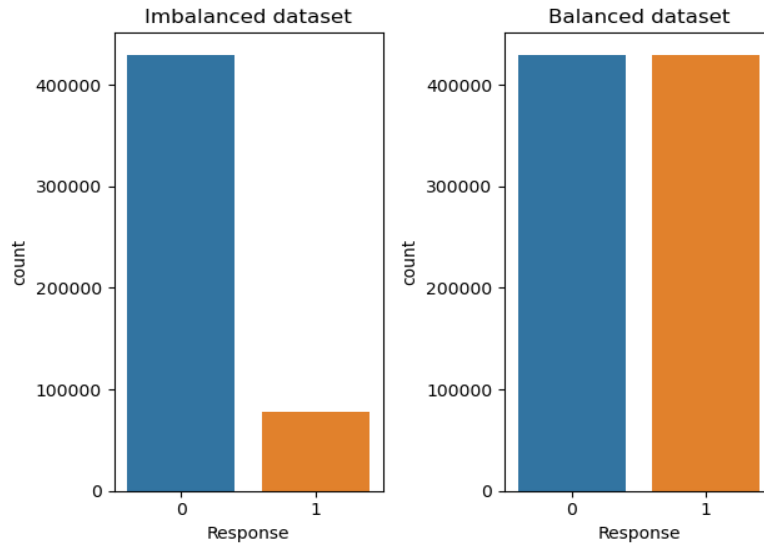


Fig 11. Balanced Dataset

To address the issue of imbalanced class-labelled datasets, under-sampling and oversampling techniques can be employed. Under-sampling involves reducing the quantity of the majority class label, which is 0 in this dataset, while oversampling involves increasing the proportion of the minority class, which is 1, through re-sampling.

To ensure the balance of the dataset, the SMOTE technique is applied only to the training portion of the dataset immediately after the test-train split phase, before delving into the specifics of machine learning. This helps to ensure that the number of response labels 1 and 0 are equal in the training set.

SMOTE:

SMOTE stands for Synthetic Minority Oversampling Technique. It's an popular oversampling technique used to address imbalanced class distributions in datasets.

Steps:

- First, SMOTE selects a minority class observation.
- Next, it finds the k-nearest neighbours of the selected observation in the minority class.
- It then generates new synthetic observations by interpolating between the selected observation and its k-nearest neighbours.
- This process is repeated until the desired balance between the minority and majority classes is achieved.

Essentially, SMOTE creates new synthetic data points for the minority class by "borrowing" features from its k-nearest neighbours. This results in an increase in the number of minority class observations, which helps to balance the class distribution in the dataset. By doing so, SMOTE can improve the performance of machine learning models trained on imbalanced datasets.

6. Result and Discussion

Logistic regression, Naïve bayes, KNN, random forest, decision tree and XGboost machine algorithm were used to predict the cross sell of insurance. Among the given algorithms XGBoost Machine algorithm was the best performing one as it provided the highest accuracy of 0.94.

Report:

```
In [27]: # Generate the classification report
y_pred = xgb_model.predict(x_test)
class_report = classification_report(y_test, y_pred)
print("Classification report:\n", class_report)
```

```
Classification report:
              precision    recall  f1-score   support

     0       0.90      0.98      0.94      107325
     1       0.98      0.89      0.94      107577

 accuracy          0.94
 macro avg          0.94
weighted avg          0.94
```

Based on the classification report, the model has an overall accuracy of 94%, which indicates that the model is performing well in predicting the target variable.

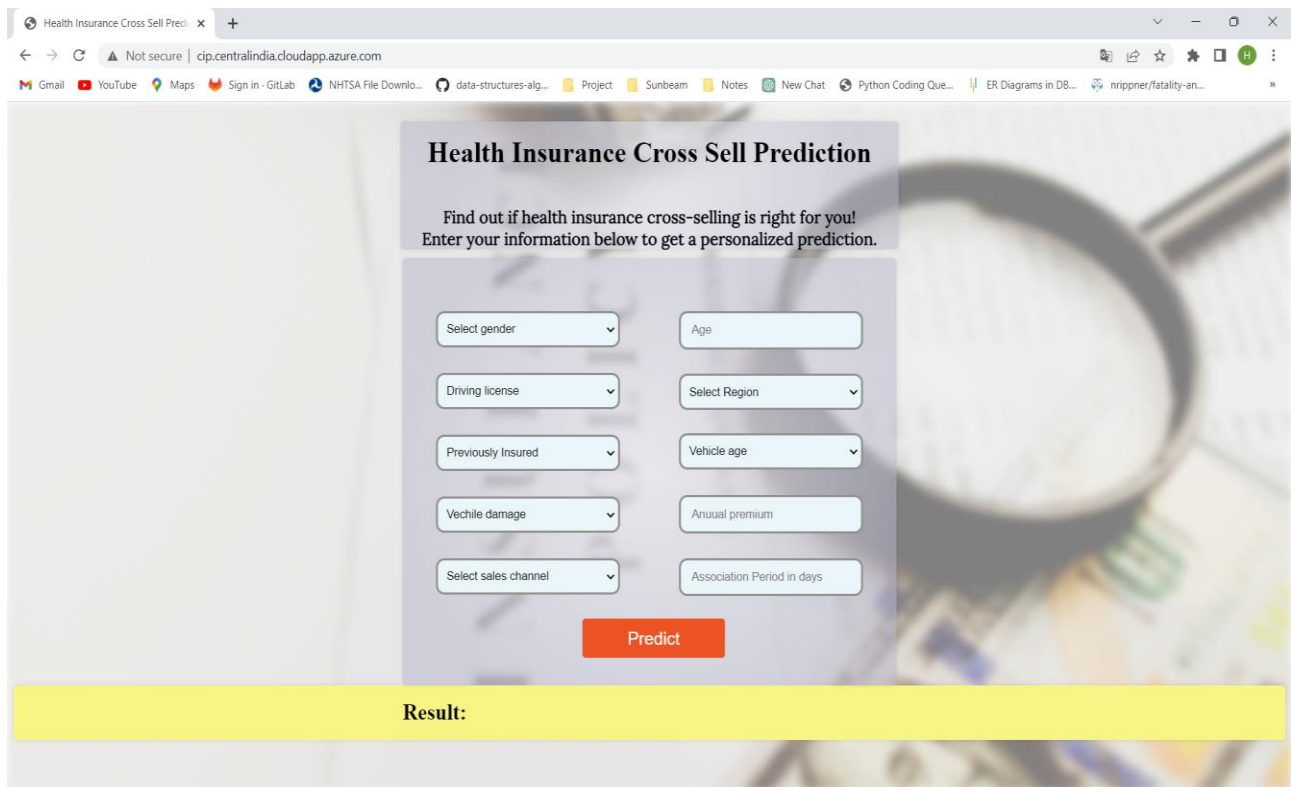
The precision and recall scores for both classes are relatively high, with class 0 having a precision score of 0.90 and a recall score of 0.98, while class 1 has a precision score of 0.98 and a recall score of 0.89. This suggests that the model is able to predict both classes accurately with only minor differences in precision and recall scores between the two classes.

The macro and weighted average f1-scores are also high at 0.94, indicating that the model has good overall performance in terms of precision, recall, and accuracy across both classes.

Overall, based on the classification report, it can be concluded that the model is performing well in accurately predicting the target variable and is a reliable model for making predictions on new data.

7.Deployment

Our application's front-end was built using html and CSS while the back-end was powered by Flask - a Python-based micro web framework. Unlike other frameworks, Flask doesn't rely on any specific tools or libraries and doesn't include pre-existing components such as database abstraction layer or form validation. However, it allows for the integration of various extensions that enable the implementation of additional features within the framework. To deploy the application, we utilized an ec2 instance provided by Azure cloud services.



The screenshot shows a web browser window with the URL `http://cip.centralindia.cloudapp.azure.com`. The page title is "Health Insurance Cross Sell Prediction". Below the title, a message reads: "Find out if health insurance cross-selling is right for you! Enter your information below to get a personalized prediction." The form contains ten input fields arranged in two columns. The left column has five dropdown menus: "Select gender", "Driving license", "Previously Insured", "Vechile damage", and "Select sales channel". The right column has five text input fields: "Age", "Select Region", "Vehicle age", "Anuual premium", and "Association Period in days". A red "Predict" button is centered below the form. At the bottom of the page, there is a yellow banner with the text "Result:".

Fig 12. Application User Interface

8.Conclusion

This study aimed to compare the performance of various machine learning algorithms in classifying samples for cross-selling marketing campaigns, with pre-processing techniques such as encoding, scaling, and oversampling employed to handle unbalanced features. The main focus was to evaluate the impact of feature selection methods on the accuracy and F1 score classification metrics of the models, both before and after applying these methods. Six state-of-the-art machine learning algorithms were employed, and the results of the base models were analysed. Additionally, one feature selection method and one dimension reduction technique were implemented to compare the classification metrics of the base models with those of the models using feature selection. The results indicated that the algorithms performed differently depending on the feature selection method applied. Notably, the feature selection methods significantly reduced the training time of the models while maintaining or even improving their performance. Future studies could further optimize the models by utilizing hyperparameter optimization techniques to identify the best parameters for each algorithm after feature selection.

9. References

- [1] T. Yu, K. de Ruyter, P. Patterson, and C. F. Chen, "The formation of a cross-selling initiative climate and its interplay with service climate," *Eur. J. Mark.*, vol. 52, no. 7–8, pp. 1457–1484, 2018, doi: 10.1108/EJM_08-2016-0487.
- [2] Kwiatkowska, J. (2018). Cross-selling and up-selling in a bank. *Copernican Journal of Finance & Accounting* 7(4), 59– 70.
<http://dx.doi.org/10.12775/CJFA.2018.02>
- [3] Purnamasari et al 2020 *J. Phys.: Conf. Ser.* 1641 012010 - The Determination Analysis Of Telecommunications Customers Potential Cross-Selling with Classification Naive Bayes and C4.5.
- [4] Sahar F. Sabbeh, "Machine-Learning Techniques for Customer Retention: A Comparative Study" *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(2), 2018. <http://dx.doi.org/10.14569/IJACSA.2018.090238>
- [5] Zhang, Ren-Qian & Yi, Meng & Wang, Qi-Qi & Xiang, Chen, 2018. "Polynomial algorithm of inventory model with complete backordering and correlated demand caused by cross-selling," *International Journal of Production Economics*, Elsevier, vol. 199(C), pages 193-198. <https://www.sciencedirect.com/science/article/abs/pii/S0925527318301245>
- [6] Hell, Franz; Taha, Yasser; Hinz, Gereon; Heibei, Sabine; Müller, Harald; Knoll, Alois. 2020. "Graph Convolutional Neural Network for a Pharmacy Cross-Selling Recommender System" *Information* 11, no. 11: 525.
- [7] Salo, J., Cripps, H., & Wendelin, R. (2020). Developing cross-selling capability in key corporate bank relationships: the case of a Nordic Bank. *Journal of Financial Services Marketing*, 25(3-4), 45-52. <https://doi.org/10.1057/s41264-020-00076-8>
- [8] Fan, Zhi-Ping & Sun, Minghe, 2016. "A multi-kernel support tensor machine for classification with multitype multiway data and an application to cross-selling recommendations Author-Name: Chen, Zhen-Yu," *European Journal of Operational Research*, Elsevier, vol. 255(1), pages 110-120.