Capstone Project : Classification

Health Insurance Cross-sell Prediction

Created by – Ayush Kumar

Abstract:

Cross-selling is a great way to make more money for any insurance agency without starting from scratch.

We can build from a business from the book you already have with your current customer relationships. It is not only benign for the company but withal for the clients. It is not only the matter of making more profit. But It is withal about integrating value and bringing solutions to the indemnification-cognate challenges your customers face.

Just like medical insurance, there is vehicle insurance where every year the customer needs to pay a premium of certain amount to insurance provider company so that in case of unfortunate accident by the vehicle, the insurance provider company will provide a compensation (called 'sum assured') to the customer.

Our dataset is based on Health Insurance Customers database.

This Experiment can help to understand what can be affecting factors for cross-selling of Insurance Plans for the already existing customers.

The model can be used for any insurance plan dataset to predict cross-selling.

Keywords: machine learning, Supervised machine learning, Cross-selling, Predictive Model building, Decision Tree, Logistic Regression, Random Forest

1. Problem Statement

Our client is an insurance company that has provided Health insurance to its customers now they require our avail in building a model to prognosticate whether the policyholders from the past year will additionally be intrigued with the Vehicle insurance provided by the company.

An insurance policy is an arrangement by which a company undertakes to provide an assurance of emolument for a designated loss, damage, illness, or death in reciprocation for the payment of a designated premium. A premium is a sum of profit that the customer needs to pay customarily to an insurance company for this assurance.

About the Dataset:

The dataset consists of these important features: -

1. **ID** : Unique ID for the customer
2. **Gender** : Gender of the customer
3. **Age** : Age of the customer
4. **Driving_Licence** : whether Customer has DL
5. **Region_Code** : Unique code for the region of the customer
6. **Previously_Insured** : Whether Customer already has Vehicle Insurance
7. **Vehicle_Age** : Age of the Vehicle
8. **Vehicle_Damage** : Whether Customer got his/her vehicle damaged in the past.
9. **Annual_Premium** : The amount customer pay yearly for Insurance.
10. **PolicySalesChannel** : Anonymized Code for the channel of outreaching to the customer ie. 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

2. Introduction

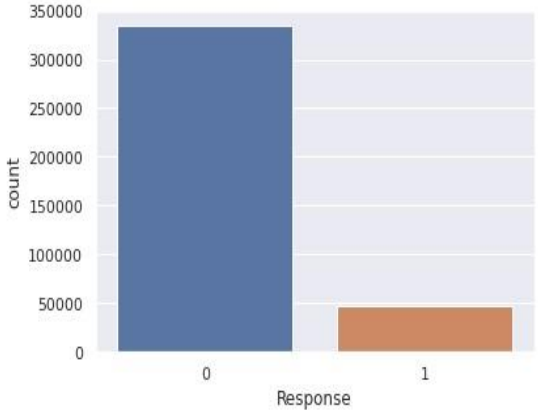
Cross-Selling is a new marketing strategy based on data analysis, found that different needs exist as well, who can become customers and meet their needs through sales of various related services or products.

We are utilizing the Decision Tree algorithm, Logistic Regression, Random Forest for building different prediction models, Decision tree algorithm build nodes based on certain decision and hence can be very useful for random data that has no specific distribution. Logistic Regression is a parametric algorithm and hence certain properties are to be verified for good results. The Random Forest engenders decision trees on randomly selected data samples, gets predictions from each tree, and selects the best solution by means of voting. Here we have features like age, driving licence, and vehicle_age which is going to give a relationship with response variable.

3. Response type:

1. Interested
2. Not Interested

Segregation of customers according to their interest is first step towards building of prediction model, we have features like age, vehicle age, annual premium of the policy taken. These features have correlation with response variable.



4. Need for cross-selling Model Building:

The need for Cross-Selling Models is-

1. For better understand the customers, need.
2. To drive profit maximization from already existing customers instead of launching new marketing campaigns.
3. To provide personal feel to the customers.
4. To make strategy for targeting the right customer segment to maximize the profit.
5. To make the changes in the products if required for better market.

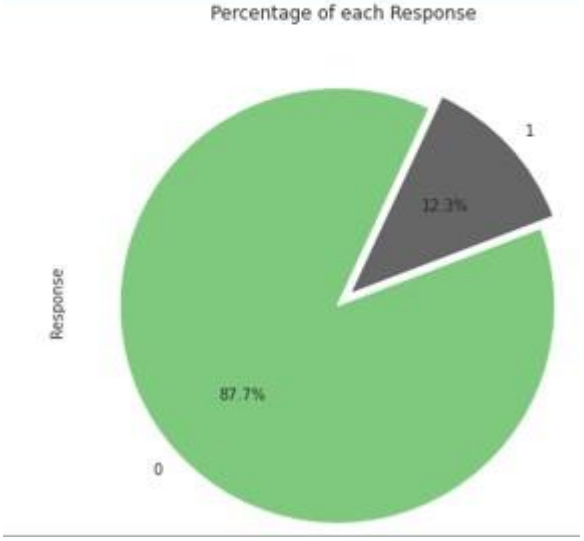
5. Steps involved:

Exploratory Data Analysis

Exploratory data analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA is primarily used to see what data can reveal beyond the formal modeling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today. An example of EDA can be seen in the adjacent figure that using plot we can visualize important information about data.

Response	Counts
0	333107
1	46582



We can see clearly in the adjacent plot that data set is imbalance hence we may require some statistical or synthetic technique to make this data balance. For this Problem we will use SMOTE technique to make this data balanced which we will explain in this document later on.

Feature Engineering:

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. In order to make machine learning work well on new tasks, it might be necessary to design and train better features. As you may know, a "feature" is any measurable input that can be used in a predictive model — it could be the color of an object or the sound of someone's voice. Feature engineering, **in simple terms, is the act of converting raw observations into desired features using statistical or machine learning approaches.**

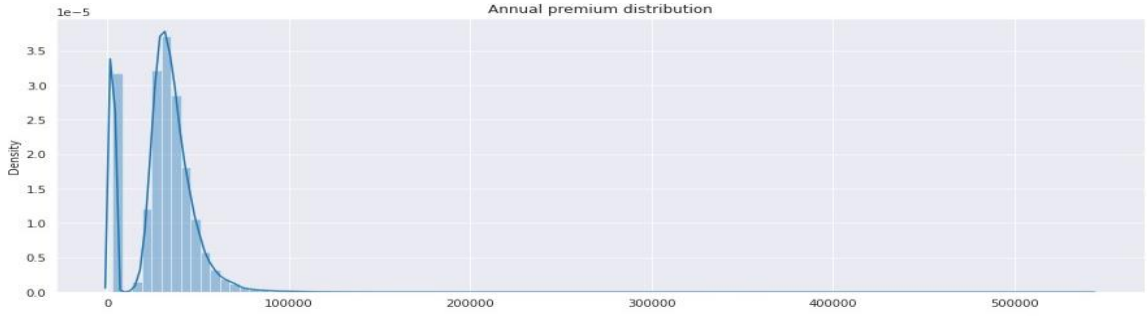
We used the following techniques;

1. Null values Treatment

While reading data in analytical tool (Python Google Colab) we found that we do not have any missing value in the dataset hence we have not done anything in Null value treatment process.

2. Handling possible outliers:

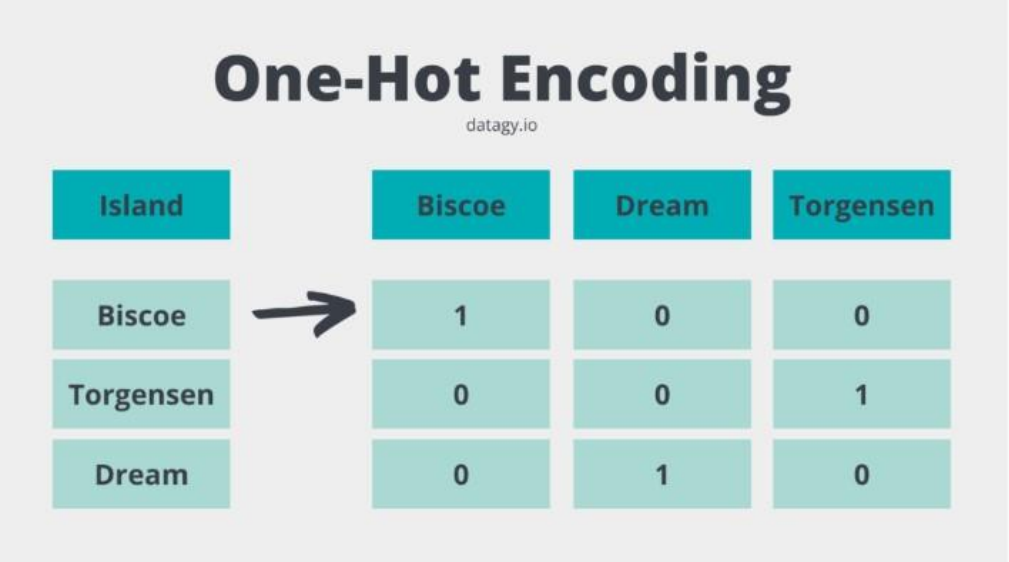
This is a very task in any machine learning project as if not properly then a machine learning technique may lead to poor results. While performing EDA we have identified two feature where there are some extreme values and these values can lead to poor predictions. Although there are many ways to treat outliers or extreme values but simply have dropped we had very high no. observation and remove some observation will not affect our dataset randomness.



In this plot we can see that after a certain value there are very less observations (Right long tail) which potentially is due to outliers. Using percentile method we have removed these outliers. Specially, we have considered an observation as outlier if it falls between 99.9 and 100 percentiles. With this we were able to remove approx. 400 observations. Similarly, we have removed outliers from Age feature and in this column an observation is considered as outliers if a person ages more than 80 years.

3. One hot encoding:

One hot encoding can be defined as the essential process of converting the categorical data variables to be provided to machine and deep learning algorithms which in turn improve predictions as well as classification accuracy of a model. One Hot Encoding is a common way of pre-processing categorical features for machine learning models.



4. Standardization:

Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization and is generally performed during the data preprocessing step. Just to give you an example — if you have multiple independent variables like age, salary, and height; With their range as (18–100 Years), (25,000–75,000 Euros), and (1–2 Meters) respectively, feature scaling would help them all to be in the same range, for example- centered around 0 or in the range (0,1) depending on the scaling technique.

In order to perform feature scaling in our project we have used specific scaling technique name as Standard Scaler which transforms the data using below formula-

$$x' = \frac{x - \bar{x}}{\sigma}$$

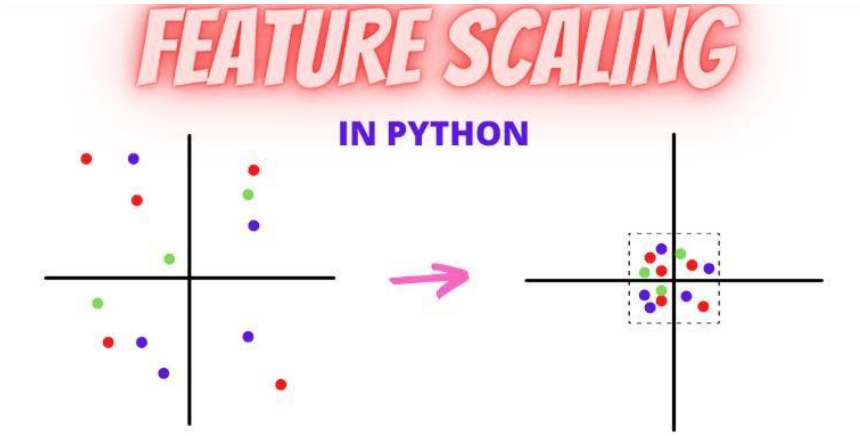
Where, x' = value after transformation

x = actual value of feature point

X_bar = mean of the all feature values

Sigma = Standard deviation of feature.

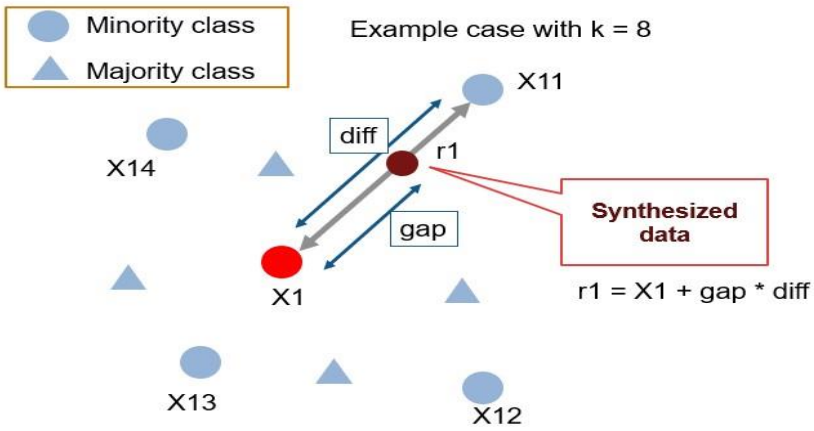
we can also see a figure here to understand how feature scaling works.

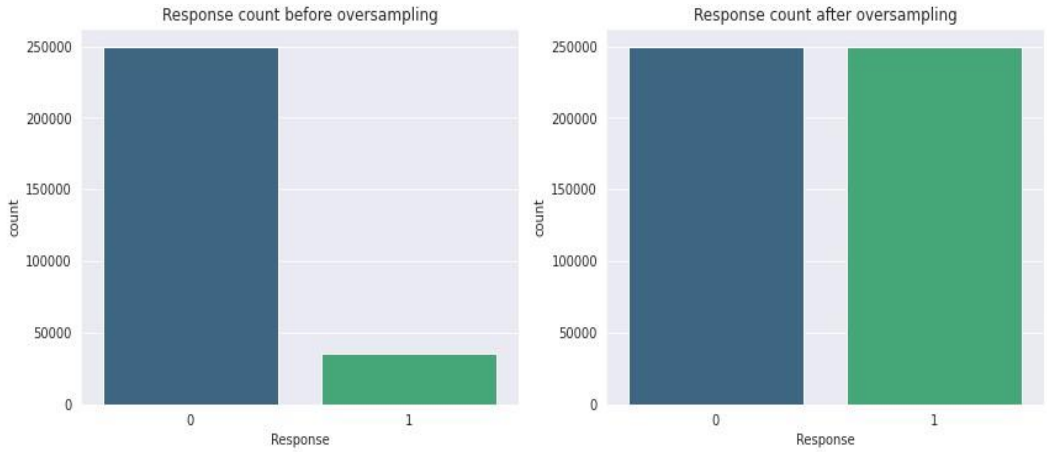


5. Handling imbalance data:

As we already have discussed that the given dataset is an imbalanced dataset which may lead to poor performance of a model as it higher weightage to majorly appearing class. To handle this kind of data set have used a technique name as SMOTE (Synthetic minority oversampling technique)

This procedure can be used to create as many synthetic examples for the minority class as are required. As described in the paper, it suggests first using random undersampling to trim the number of examples in the majority class, then use SMOTE to oversample the minority class to balance the class distribution.





6. Model implementation:

After we performed feature scaling and one hot encoding on respective features we moved next step i.e applying machine learning model to predict a class for a new observation. In this project we have used three models- **1. Decision Tree model**

2. Logistic Regression

3. Random Forest model

1. Decision tree for classification:

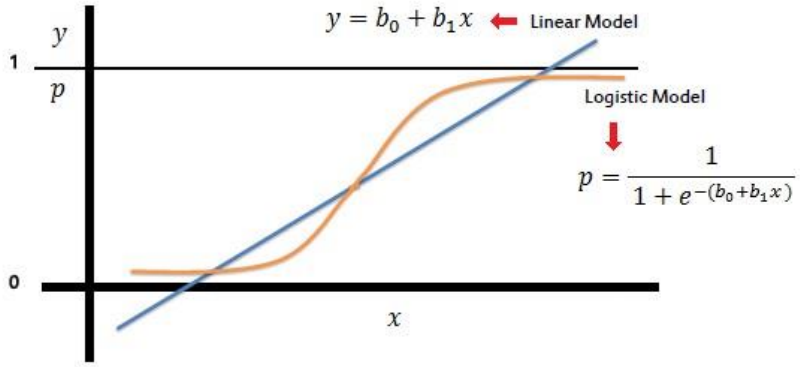
Decision tree builds classification or regression models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with **decision nodes** and **leaf nodes**. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy). Leaf node (e.g., Play) represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called **root node**. Decision trees can handle both categorical and numerical data.

2. Logistic Regression:

Logistic regression predicts the probability of an outcome that can only have two values (i.e. a dichotomy). The prediction is based on the use of one or several predictors (numerical and categorical). A linear regression is not appropriate for predicting the value of a binary variable for two reasons:

- A linear regression will predict values outside the acceptable range (e.g. predicting probabilities
  outside the range 0 to 1)
- Since the dichotomous experiments can only have one of two possible values for each experiment, the residuals will not be normally distributed about the predicted line.

On the other hand, a logistic regression produces a logistic curve, which is limited to values between 0 and 1. Logistic regression is similar to a linear regression, but the curve is constructed using the natural logarithm of the "odds" of the target variable, rather than the probability. Moreover, the predictors do not have to be normally distributed or have equal variance in each group.




```
> data <- read.csv("credit_scoring.csv")
> fit <- glm (DEFAULT~., family=binomial(logit), data=data )
> summary(fit)
```

	Coefficient	SE	Z	P(z)
(Intercept)	-5.706	1.157	-4.933	0.000
BUSAGE	0.008	0.004	2.219	0.027
DAYSDELQ	0.102	0.020	5.106	0.000

$$y' = b_0 + b_1 x_1 + b_2 x_2$$

$if\ p < 0.5\ then$ N $else$ Y

BUSAGE	DAYSDELQ	DEFAULT	y'	p	Prediction
87	2	N	-4.811	0.008	N
89	2	N	-4.795	0.008	N
100	26	Y	-2.261	0.094	N
275	54	Y	1.983	0.879	Y
42	57	Y	0.437	0.608	Y
88	53	N	0.395	0.597	Y

$$y' = -5.706 + 0.008 \times 87 + 0.102 \times 2 = -4.811$$

$$p = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + b_2 x_2)}} = \frac{1}{1 + e^{-y'}}$$

$$p = \frac{1}{1 + e^{4.811}} = 0.008 \xrightarrow{<0.5} \text{N}$$

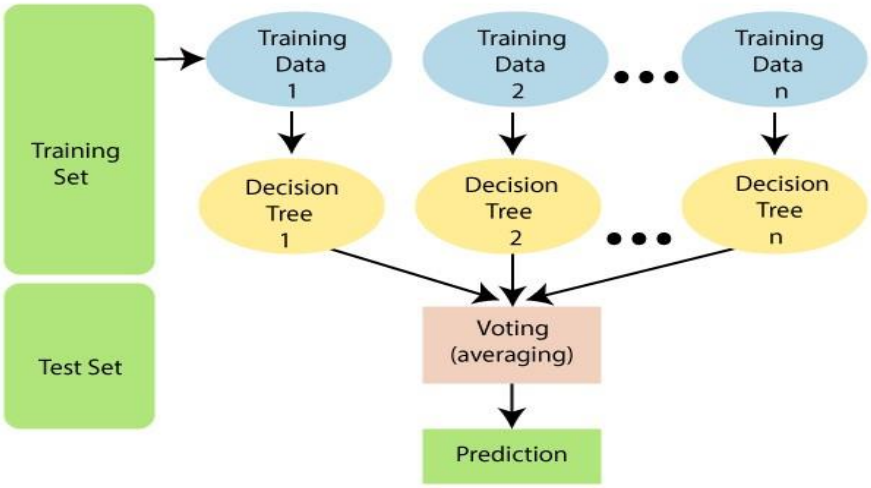
3. Random Forest for Classification:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning,** which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, ***"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."*** Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



Assumptions for Random Forest

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

Why use Random Forest?

Below are some points that explain why we should use the Random Forest algorithm:

<=”” li=””> ○ It takes less training time as compared to other algorithms. ○ It predicts output with high accuracy, even for the large dataset it runs efficiently. ○ It can also maintain accuracy when a large proportion of data is missing.

How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

7. Model Explainability:

When working with machine learning model it becomes very important to explain how and why a model is giving us this result because a wrong model can become a cause of very high loss and hence data scientists takes care of this problem and with the use of some advanced technique explain how and why a model is producing a specific result. To explain a model, we have two ways to give answer for such question –

1. Explain Globally : in this way we combine our all results and come up with an explanation that which feature is causing which class. To explain it we use an algorithm that was built to explain tree based model named SHAP (SHapley Additive exPlanations) Summary plot .

The summary plot combines feature importance with feature effects. Each point on the summary plot is a Shapley value for a feature and an instance. The position on the y-axis is determined by the feature and on the x-axis by the Shapley value. The color represents the value of the feature from low to high. Overlapping points are jittered in y-axis direction, so we get a sense of the distribution of the Shapley values per feature. The features are ordered according to their importance.

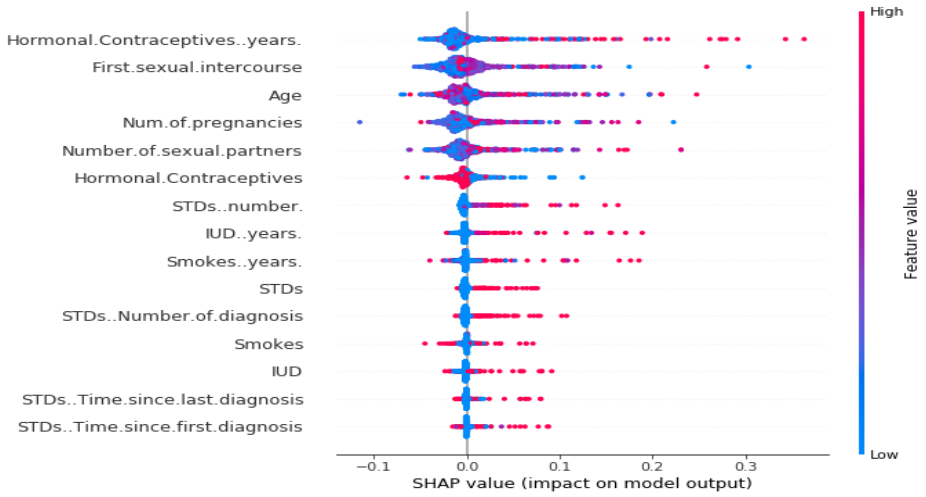
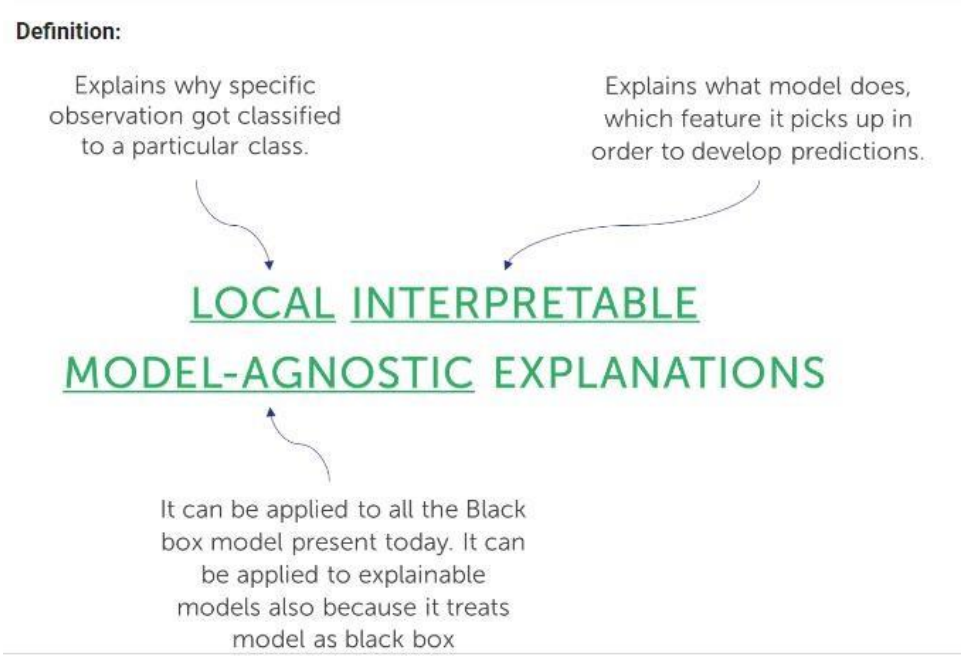


FIGURE: SHAP summary plot. Low number of years on hormonal contraceptives reduce the predicted cancer risk, a large number of years increases the risk. Your regular reminder: All effects describe the behaviour of the model and are not necessarily causal in the real world.

In the summary plot, we see first indications of the relationship between the value of a feature and the impact on the prediction. But to see the exact form of the relationship, we have to look at SHAP dependence plots

2. **Local interpretability:**

This technique is used to explain a particular observation that what are the factors that are causing it to push towards a particular class.



By generating an artificial dataset around the particular observation, we can try to approximate the predictions of the Black box model locally using a simple interpretable model. Then this model can be served as a ‘’local explainer’’ for the Black box model.

However, the representation would vary with the type of data. Lime supports Text data, Image data, and Tabular types of data:

- **For text data:** It represents the presence or absence of words.
- **For image:** It represents the presence or absence of super pixels. **For tabular data:** It is a weighted combination of columns

We see that Python gives us output is in 3 parts which can be interpreted as follows: The leftmost fig shows that this particular customer has a very high (almost 80%) chance of default.

- The contribution of each feature to the prediction is denoted just below the feature name in the middle fig.
- The table on the right shows the top 4 features(highlighted in orange) that favors class 1 and one feature (highlighted in blue) favors class 0 with actual values of the specific observation.

The reliability of this prediction can be given by R^2.

Advantages of LIME :

1. LIME can be implemented in Python (packages: lime, Skater) and R (Packages: lime, iml, DALEX). It is very easy to use.
2. Most of these packages are very flexible. You can specify m - the number of features for the model, how you want to permute your data, any simple model that would fit data.
3. Furthermore LIME is the interpretation technique that works for tabular, text, and image data.

Drawbacks of LIME :

1. The fitting of a linear model can be inaccurate (but we can check the R squared value to know if it is the case).
2. Lime depends on the random sampling of new points (so it can be unstable).
3. To be extra sure about the model understanding we can make use of SHAP in conjunction with LIME.

Libraries Used:

1. NumPy: NumPy stands for Numerical Python. It is the most fundamental library package in python and lays the foundation of multiple useful libraries in python. NumPy deals with array manipulation, basic mathematical and statistical operations, random number simulation, and much more.

2. PANDAS: Pandas is the data scientists go to python package. It is primarily used for data wrangling and analysis. It offers data structures and operations for manipulating numerical tables and time series. It is the most extensively used library in the project.

3. **Matplotlib:** Matplotlib is the plotting library closely associated with NumPy. It offers support for a number of different kinds of visualizations. This is the staple plotting package used by data analysts for basic and exploratory data analysis

4. **Seaborn:** Seaborn is another prominent plotting library used by analysts. Seaborn has extended support and compatibility for pandas and has flexible syntax in comparison to matplotlib. It includes a wider range of visualization types

5. **Scikit-learn**: is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

Lime: Lime package that helps to visualize and understand the results of an algorithm. It work a s a local interpreter which means it can explain why a particular observation is assigned to a particular call and what are the features that are affecting it

SHAP (SHapley Additive exPlanations) is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions (see papers for details and citations).