**Customer Churn Analysis**

Weekly Report

Ahmedabad University

4rier Series

Prof. Mehul Raval

CSE523 - Machine Learning

After finding out features that are highly correlated, the next step to be performed is essentially building the classification model.

The questions that need to be answered while building the model are:

* What are the features/attributes that make the customer churn?
* What features should be used to train our model for efficient classification with high performance?

**Generalized Linear Model**

Generalized Linear Model can be used to figure out the statistical properties of features. Using a generalized model can help in figuring out the relationship between the target and the features without having to make any prior assumptions.

Library used: statsmodels.api (<https://www.statsmodels.org/stable/glm.html>)

Code: The dependent variable in the model is Churn. The remaining columns, excluding CustomerId, are used as the independent variables. The Binomial() family argument specifies that the dependent variable has a binary outcome (i.e., Churn or No Churn). The fit() method is then called on the GLM model object to fit the model to the data using maximum likelihood estimation. Finally, the results of the model are summarized.

The model provides the following measures:

* Estimated coefficients: represent the estimated effects of the predictor variables on the response variable. These coefficients are obtained through maximum likelihood estimation and are also known as regression coefficients or model parameters.
* Standard Errors: represent the standard deviation of the sampling distribution of the estimated coefficients.
* P values: used to determine the statistical significance of the estimated coefficients (i.e., the effect of each predictor variable) in the model. A small p-value (e.g., less than 0.05) suggests that the estimated coefficient is unlikely to have arisen by chance.

Based on the above-mentioned comparison measures, an absolute p-value of <0.05 affects customer churn in a significant way.

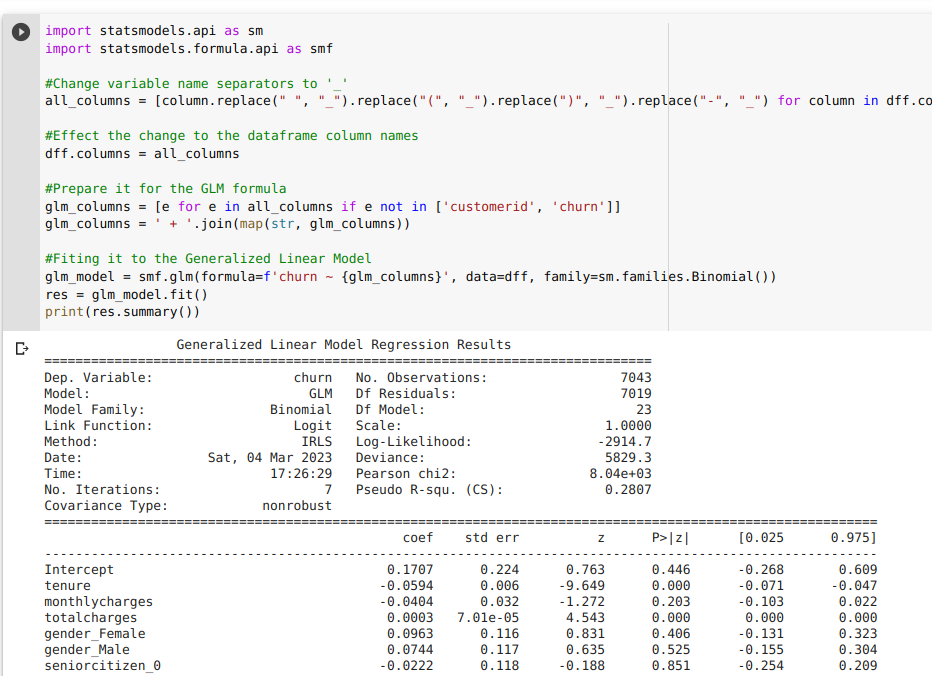
The features with p-values less than 0.05 are:

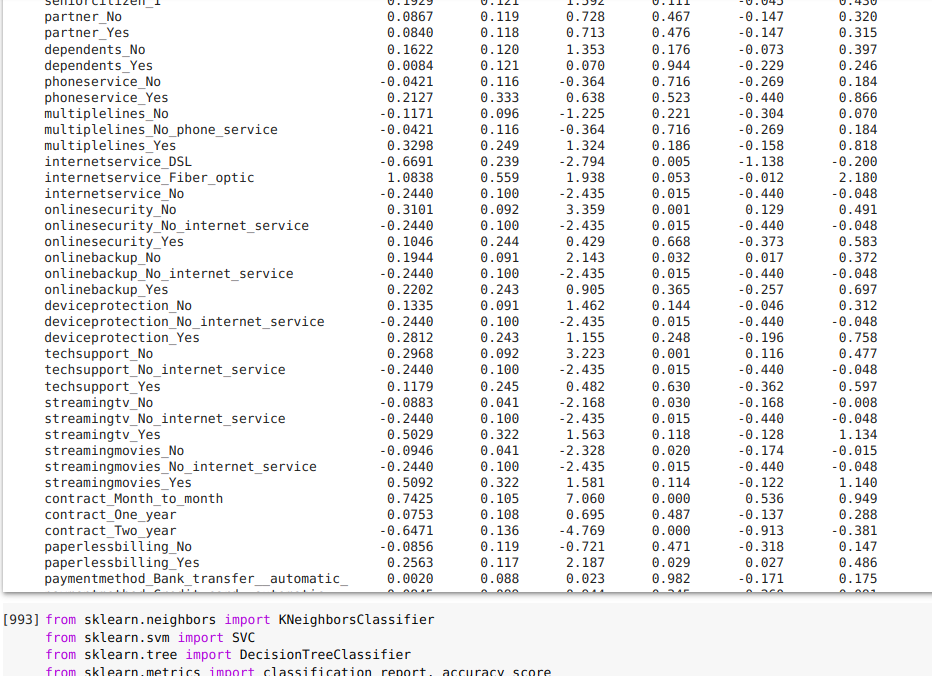
* SeniorCitizen
* Tenure
* PaperlessBilling
* TotalCharges
* MultipleLines\_No\_phone\_service
* MultipleLines\_Yes
* InternetService\_Fiber\_optic
* OnlineSecurity\_No\_internet\_service
* OnlineBackup\_No\_Internet\_service
* DeviceProtection\_No\_Internet\_service
* TechSupport\_No\_Internet\_service
* StreamingTV\_No\_Internet\_service
* StreamingMovies\_No\_Internet\_service
* Contract\_One\_Year
* Contract\_two\_Year
* PaymentMwthod\_Electronic\_check

Furthermore, exponentiating the estimated coefficients gives the exponential coefficients, also known as odds ratios. The exponential coefficients represent the change in the odds of the response variable for a one-unit change in the corresponding predictor variable while holding all other predictor variables constant. Values more than 1 indicate increased churn. Values less than 1 indicate that churn is happening less.

The features with exponential coefficients of greater than 1 are:

* Intercept
* Senior Citizen
* PhoneService
* PaperlessBilling
* TotalCharges
* MultipleLines\_No\_phone\_service
* MultipleLines\_Yes
* InternetService\_Fiber\_optic
* OnlineBackup\_Yes
* DeviceProtection\_Yes
* StreamingTV\_Yes
* StreamingMovies\_Yes
* PaymentMethod\_Electronic\_check





**Feature Scaling**

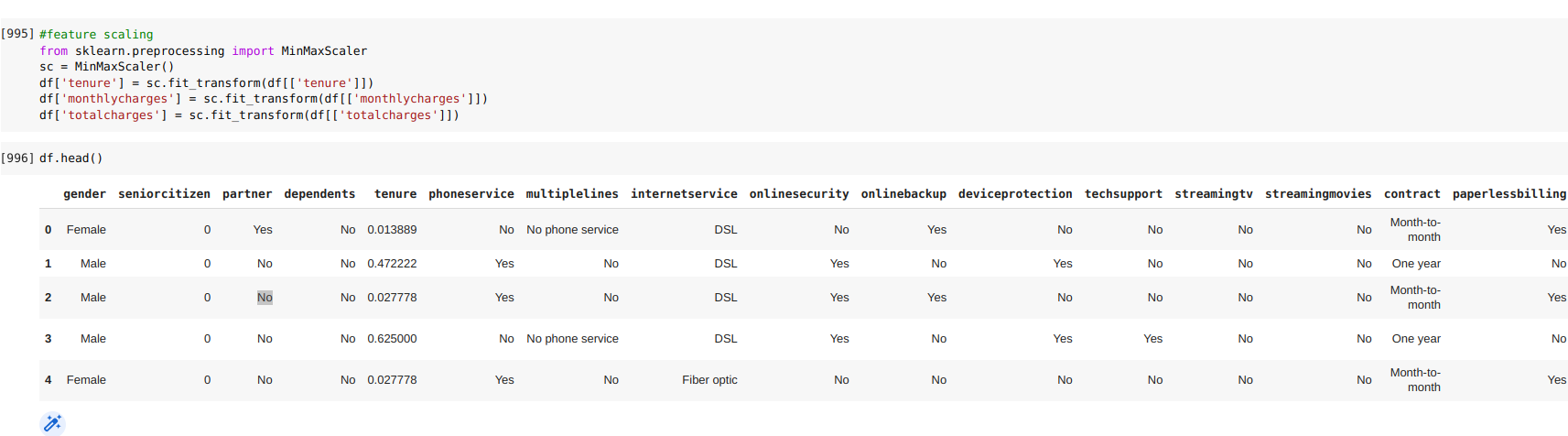
Numerical inputs/features need to be scaled to have their outputs in the same range. This is done because ML algorithms use distance measures to determine the similarity between data points. If the features have different scales, the distance between them may not be meaningful.

There are three numerical features/inputs:

* Tenure
* MonthlyCharges
* TotalCharges

Library used: sklearn

The MinMaxScaler scales the values of a column such that they lie between 0 and 1. The columns tenure, monthlyCharges, and totalCharges are scaled individually and then pushed back to the original database.



**Logistic regression**

Library used: sklearn

The essential procedure here is first to implement the model on the dataset and then judge the efficiency and accuracy of the model based on the performance metrics. The dataset is initially split into two sets: train and test. The training dataset, as is apparent from the name, is used to train the model, and the testing dataset is used to produce the results and performance metrics.

For this model, four performance metrics are used:

* Accuracy: It measures the proportion of correct predictions made by the model, which is determined by dividing the number of correctly classified observations by the total number of observations.
* Precision: Precision measures the accuracy of positive predictions made by the model, and specifically, it represents the model's capability to avoid incorrect positive predictions. Mathematically, precision is calculated by dividing the number of true positive predictions by the sum of true positive and false positive predictions.
* Recall: The term recall refers to the model's capacity to detect positive instances accurately. It is calculated by dividing the number of true positive predictions by the sum of true positive and false negative predictions.
* F1 score: The F1 score is the harmonic mean of precision and recall, and it provides a single value that combines both measures. F1 score lies between 0 and 1, where 1 represents the best possible score.

**Support Vector Classifier**

Support Vector Classifier is part of the Support Vector Machine algorithms.

The objective of the SVM (Support Vector Machine) algorithm is to generate an optimal decision boundary, commonly known as a hyperplane, that can divide an n-dimensional space into different classes. If the hyperplane that we are using for classification is in linear condition, then the condition is SVC.

Margin refers to the distance of vectors from the hyperplane, which is the separation of a line to the nearest points of a class. The objective is to select a hyperplane that maximizes the margin between classes.

The model used in Logistic Regression is also used to evaluate the performance of the algorithm on our dataset. The performance metrics also remain the same as above.

**References**

sklearn.linear\_model.LogisticRegression. (n.d.). Scikit-learn. <https://scikit-learn/stable/modules/generated/sklearn.linear_model.LogisticRegression.html>

*Generalized Linear Models — statsmodels*. (n.d.). Generalized Linear Models — Statsmodels. <https://www.statsmodels.org/stable/glm.html>

*Support Vector Machine (SVM) Algorithm - Javatpoint*. (n.d.). www.javatpoint.com. <https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm>

S, P. (2021, June 16). *The A-Z guide to Support Vector Machine*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2021/06/support-vector-machine-better-understanding/>