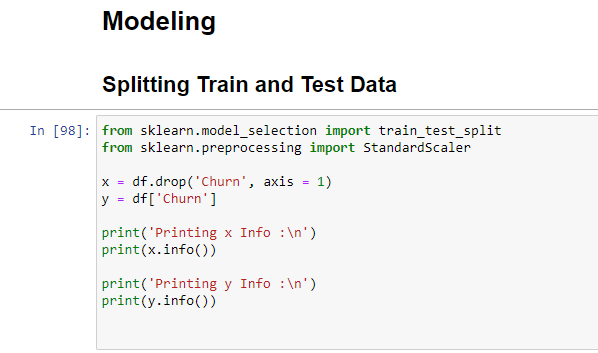
**Chapter - 4**

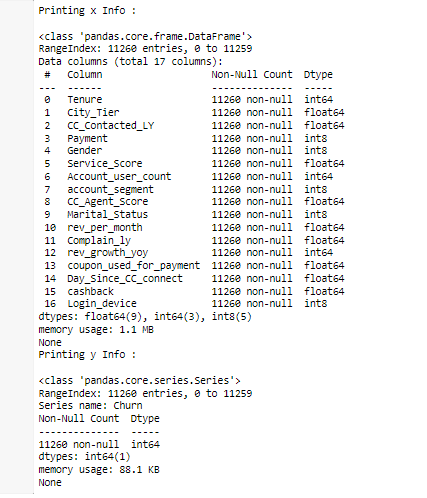
**Model building and interpretation.**

Model building is an essential part of data analytics and is used to extract insights and knowledge from the data to make business decisions and strategies. In this phase of the project, team needs to develop data sets for training and testing. These data sets enable data scientists to develop an analytical method and train it while holding aside some of the data for testing the model.

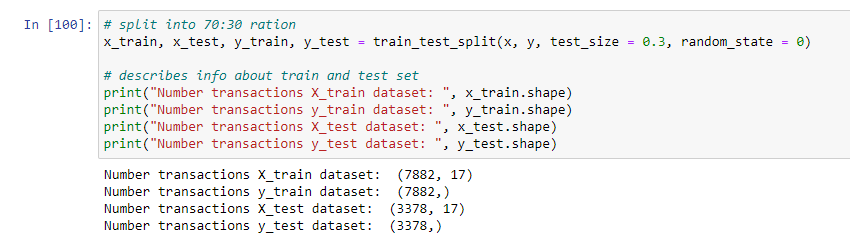
Model building in data analytics is aimed at achieving not only high accuracy on the training data but also the ability to generalize and perform well on new, unseen data. Therefore, the focus is on creating a model that can capture the underlying patterns and relationships in the data, rather than simply memorizing the training data.



Scikit-learn alias **sklearn** is the most useful and robust library for machine learning in Python. The **scikit-learn library** provides us with the model\_selection module in which we have the splitter function train\_test\_split().



After Data cleaning and transformation out of total 11260 Rows of data, we will be splitting the data to train and test data in 70:30 ratio.



Train subset – we will use this subset to fit/train the model

Test subset – we will use this subset to evaluate our model

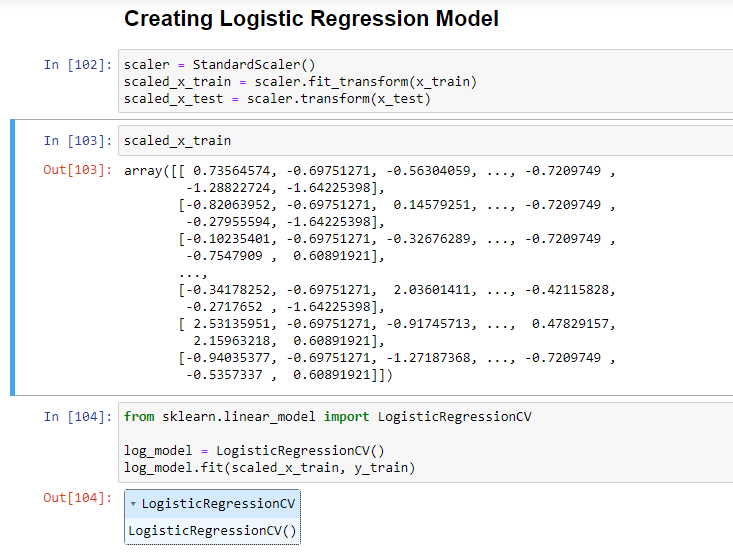
1. Building Various Model

**Logistic Regression Model :**

Logistic regression Model used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as customer churned or not churned, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

To implement Logistic Regression, we will use the Standard Scalar library. We’ll start by building a base model with transformed data parameters scaled\_x\_train and scaled\_x\_test, then look at how to improve it with Tuning.

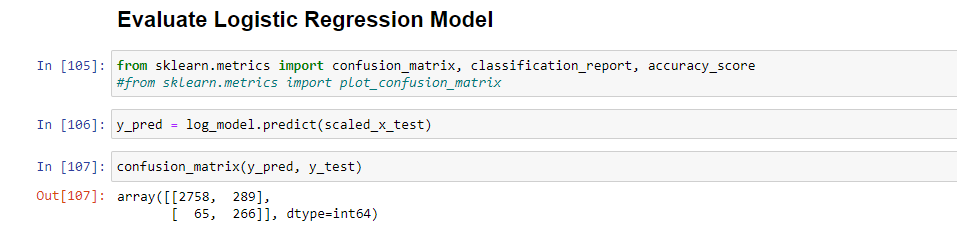
Let’s start by creating our base model with the code below.



After training our model on the training dataset, we used our model to predict values for the test dataset and recorded them in the LogisticRegressionCV.

Let’s look at which metrics to use and how to evaluate our base model.

To evaluate our Model we will be using “Confusion matrix” also the most crucial metric commonly used to evaluate classification models.



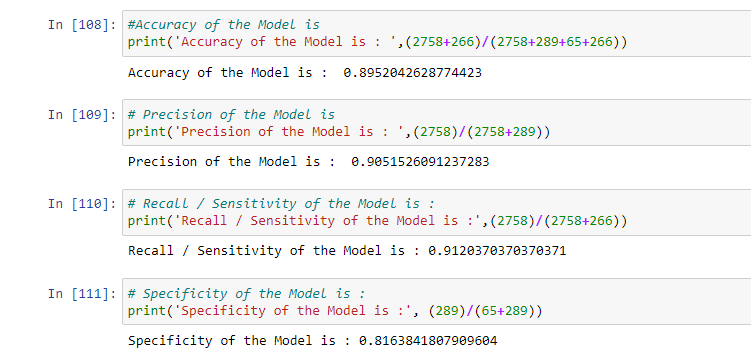
1. Testing predictive Model

The confusion matrix avoids "confusion" by measuring the actual and predicted values in a tabular format. In array above, value 2758 represent true positive data .

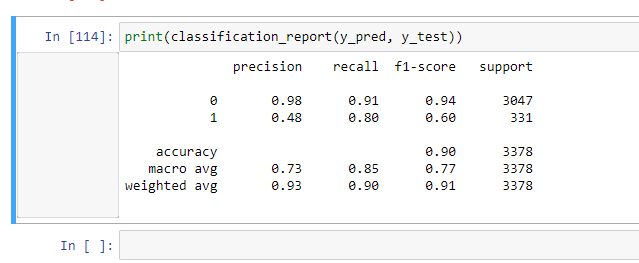
The above array represents the below tabular explanation which will be used to predict the metrics of our model.

|  |  |  |
| --- | --- | --- |
| **Confusion Matrix** | **1(Predicated)** | **0 (Predicted)** |
| **1 (Actual)** | TRUE positive 2758 | FALSE negative  289 |
| **0 (Actual)** | False positive 65 | TRUE negative 266 |

Following are the metrics we can derive from a confusion matrix:



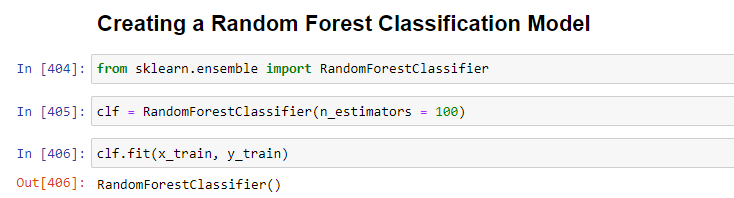
|  |  |
| --- | --- |
| **Metric** | **Predicted value** |
| Accuracy | 0.8952 |
| Precision | 0.9051 |
| Recall / Sensitivity | 0.9120 |
| Specificity | 0.8163 |



**Random Forest Classification Model :**

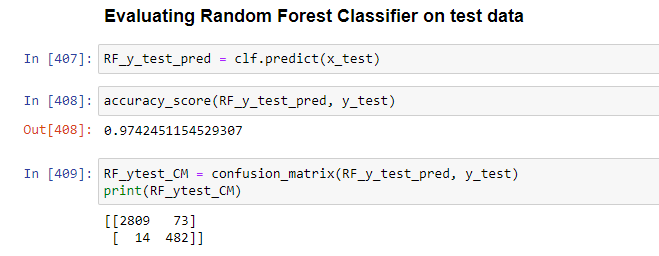
* Random forests are for supervised machine learning, where there is a labeled target variable.
* Random forests can be used for solving regression (numeric target variable) and classification (categorical target variable) problems.
* Random forests are an ensemble method, meaning they combine predictions from other models.

We first create an instance of the Random Forest model, with the default parameters. We then fit this to our training data and test data individually. So the model will predict the best data for the model. We pass both the features and the target variables, so the model can learn.



We have a trained Random Forest model, but we need to evaluate whether it is making accurate predictions.

We will evaluate the model with the Test data as a below code :

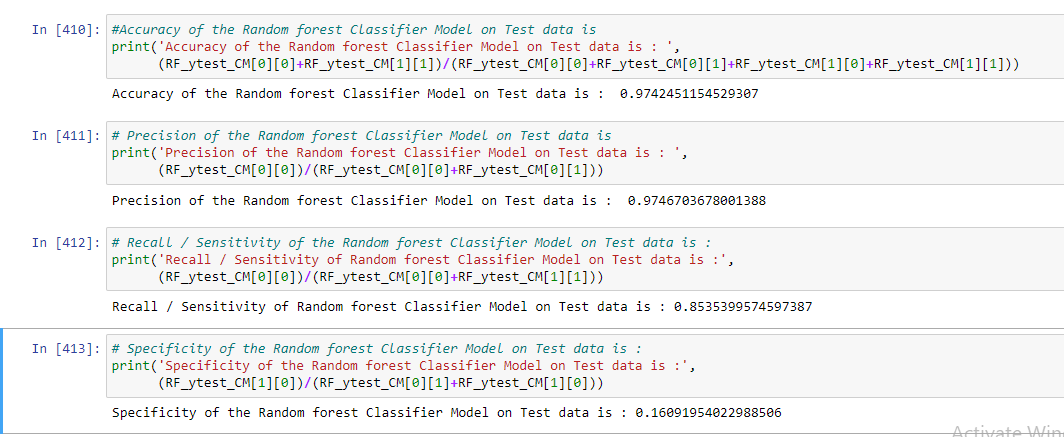


The Random Forest matrix measures the actual and predicted values in a tabular format. In array above, value 2809 represent true positive data .

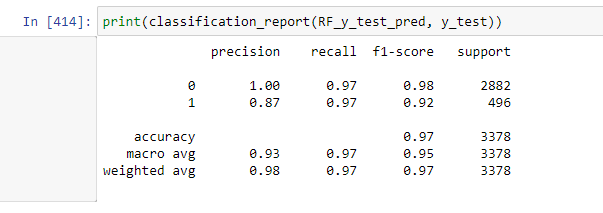
The above array represents the below tabular explanation which will be used to predict the metrics of our above model with Test data.

|  |  |  |
| --- | --- | --- |
| **Random Forest** | **1(Predicated)** | **0 (Predicted)** |
| **1 (Actual)** | TRUE positive 2809 | FALSE negative  73 |
| **0 (Actual)** | False positive 14 | TRUE negative 482 |

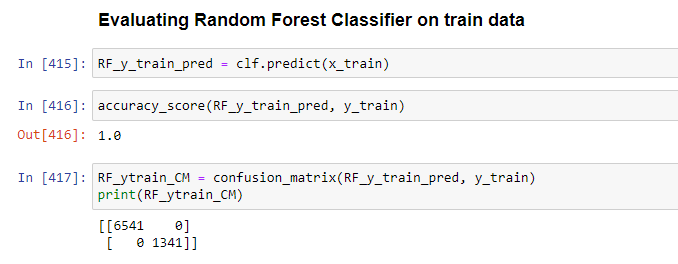
Following are the metrics we can derive from a Random Forest with Test data:



|  |  |
| --- | --- |
| **Metric** | **Predicted value** |
| Accuracy | 0.9742 |
| Precision | 0.9746 |
| Recall / Sensitivity | 0.8535 |
| Specificity | 0.1609 |



We will evaluate the model with the Train data as a below code :

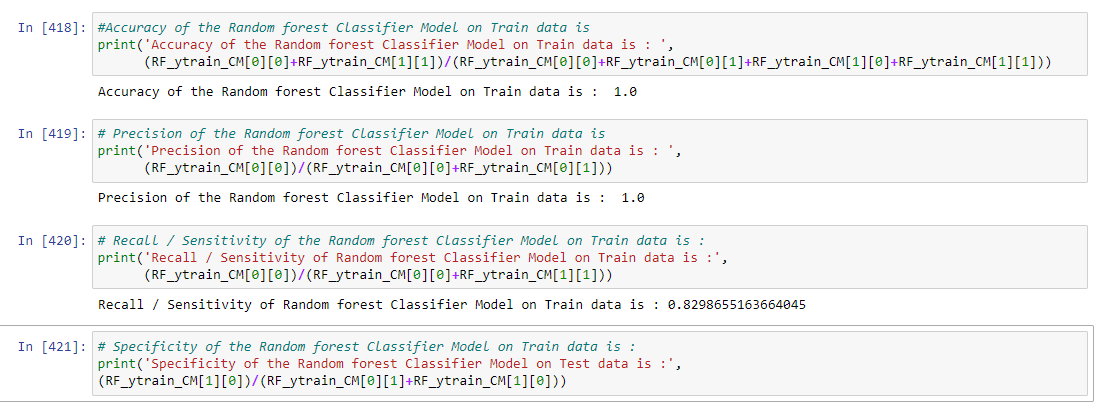


The Random Forest matrix measures the actual and predicted values in a tabular format. In array above, value 6541 represent true positive data .

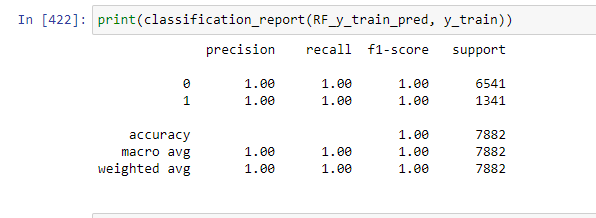
The above array represents the below tabular explanation which will be used to predict the metrics of our above model with train data.

|  |  |  |
| --- | --- | --- |
| **Random Forest** | **1(Predicated)** | **0 (Predicted)** |
| **1 (Actual)** | TRUE positive  6541 | FALSE negative  0 |
| **0 (Actual)** | False positive  0 | TRUE negative  1341 |

Following are the metrics we can derive from a Random Forest with train data:



|  |  |
| --- | --- |
| **Metric** | **Predicted value** |
| Accuracy | 1 |
| Precision | 1 |
| Recall / Sensitivity | 0.8298 |
| Specificity | Cannot calculate as TN is 0 |



C ) Interpretation of the Model :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Evaluation Metrics of models created using various classification algorithms** | | | | | |
| **Algorithm** | **Type of Data** | **Accuracy** | **Precision** | **Recall / Sensitivity** | **Specificity** |
| Logistic Regression | Test Data | 0.89 | 0.9 | 0.91 | 0.18 |
| Train Data | 0.88 | 0.89 | 0.91 | 0.2 |
| Random Forest Classifier | Test Data | 0.97 | 0.97 | 0.85 | 0.17 |
| Train Data | 1 | 1 | 0.82 | Cannot calculate as TN is 0 |