CREDIT CARD APPROVAL

Problem Identification

- For banks, Risk management has always been a crucial part in issuing credit card.
- Banks need to determine whether a customer will be able to repay the credit amount based on various factors before issuing credit cards.
- Through robust risk solution, banks can prevent credit default risk to a large extent.

Summary

Attempts to predict whether credit card applications get approved or not based on factors such as credit scores, number of delinquencies, hard inquiries, credit card utilization rate, income and credit history of the customer

Description

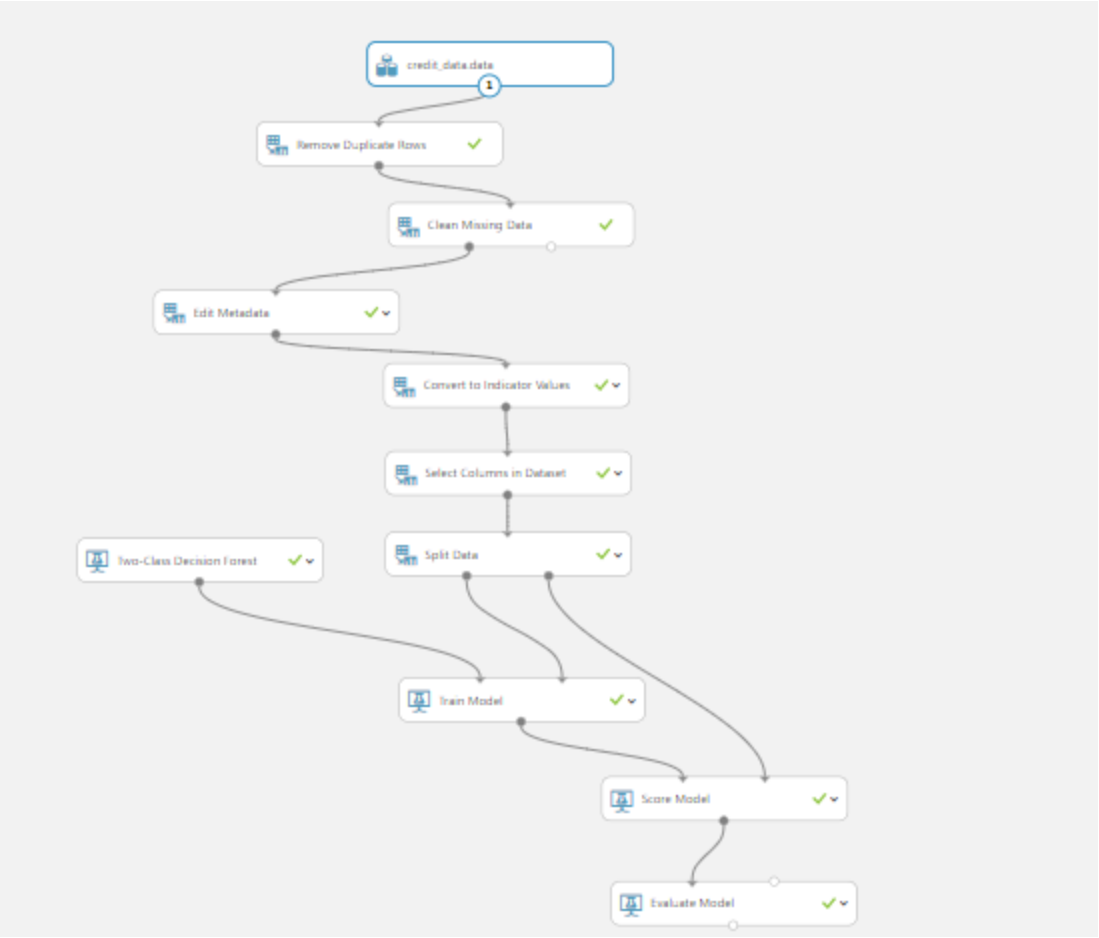
The purpose of this experiment is to use Machine learning techniques to predict whether a particular credit card application gets approved or not. We will use a Machine learning technique called Classification for this task. Classification is considered an instance of supervised learning, i.e., learning where a training set of correctly identified observations is available.

Dataset

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15	A16
0	b	30.83	0.000	u	g	w	v	1.25	t	t	1	f	g	00202	0	+
1	a	58.67	4.460	u	g	q	h	3.04	t	t	6	f	g	00043	560	+
2	a	24.50	0.500	u	g	q	h	1.50	t	f	0	f	g	00280	824	+
3	b	27.83	1.540	u	g	w	v	3.75	t	t	5	t	g	00100	3	+
4	b	20.17	5.625	u	g	w	v	1.71	t	f	0	f	s	00120	0	+

The publishers of this dataset have masked the real information of customers to protect their confidentiality and the good thing is our machine learning model will still be able to understand

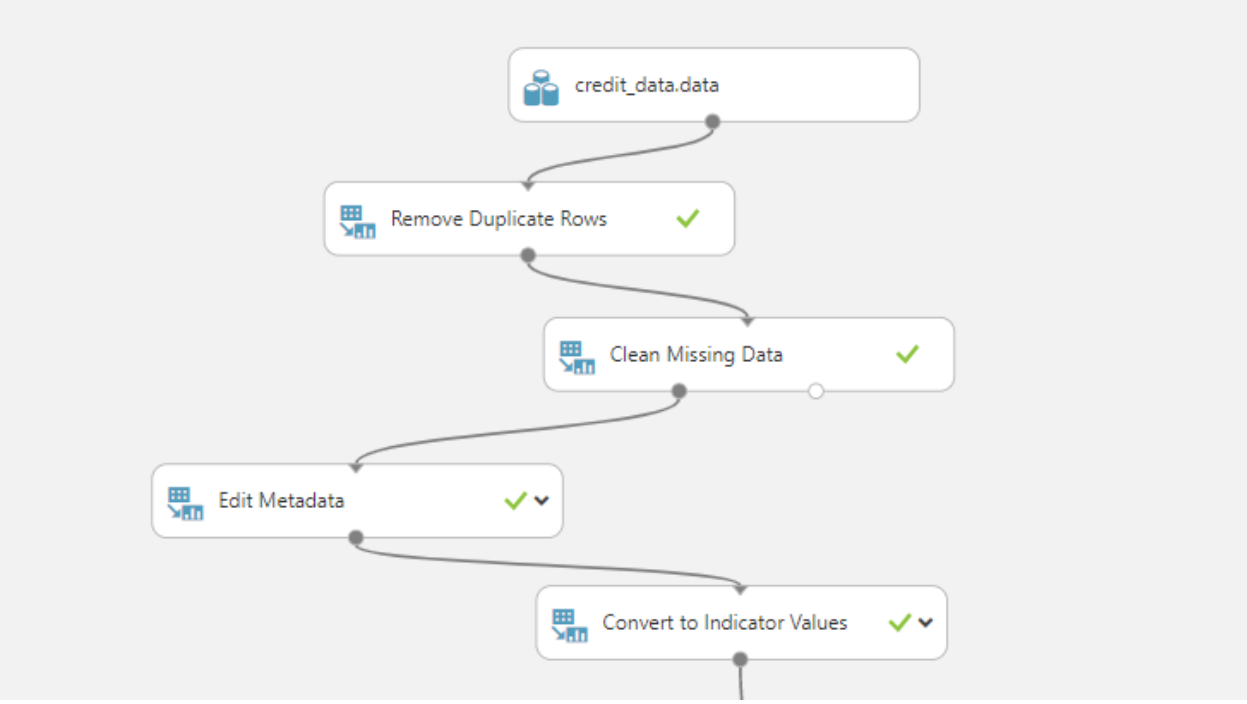
the relationships among attributes and can predict whether a credit card application get approved or not. The data consists of credit card applications. All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data. The dataset consists of 690 records with 15 columns and 1 label column. Every record in our table is a unique customer and every customer is represented by 15 different factors such as credit scores, number of delinquencies, hard inquiries, credit card utilization rate, income and credit history etc. The features are a mix of numeric and categorical types. The original data and the description of individual columns can be found in the UCI data repository.



Model

Pre-process Data

The first few steps consists of preparing the data for the downstream modules.



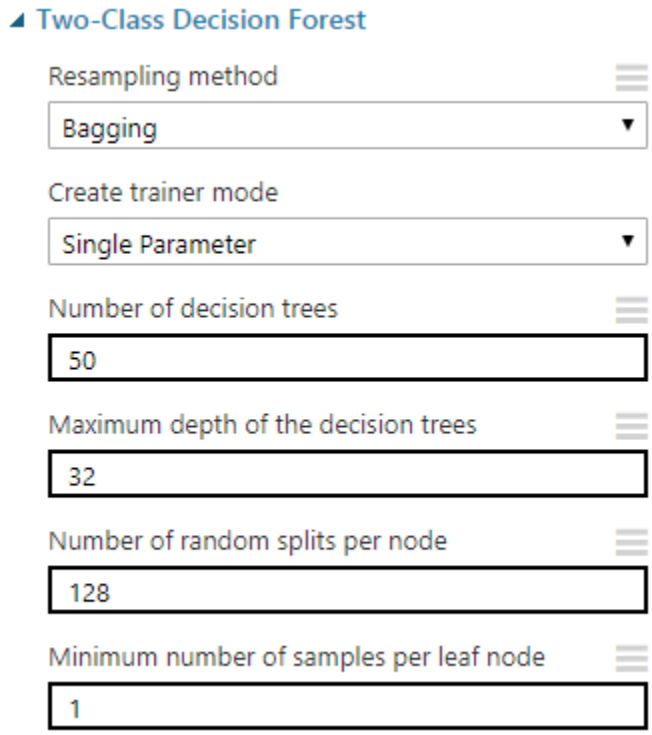
1. After importing the data from the source, we remove duplicates in the data if any. This step is optional.
2. We understand there are some missing values in the data. Hence, we will handle it by cleaning the missing values. There are multiple options such as Replacing the missing values with mean, median, mode etc. But here we are going to remove the entire row from the dataset.
3. Edit metadata cell helps us to change the datatype of our attributes, convert them to categorical etc.
4. Next, we will convert the attributes which was convert to categorical type in the previous step to one hot encoded value. The name one hot encoding comes from the way we design the values where one bit is hot/on and rest are set off.

Learners

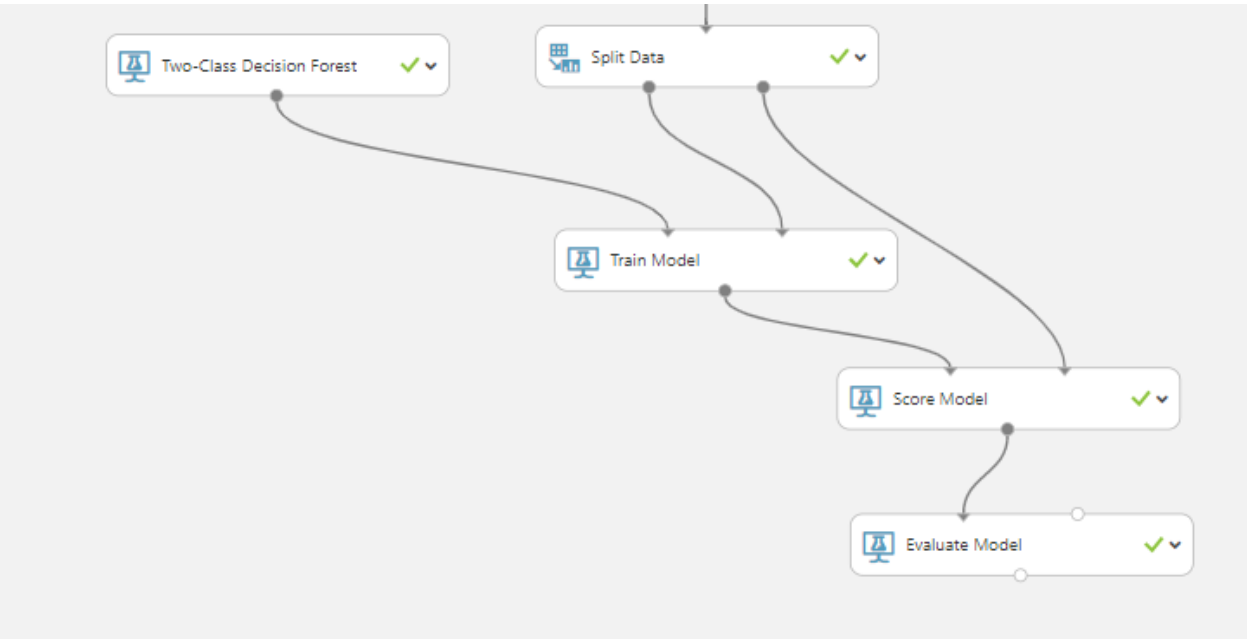
Two-class Decision Forest: This decision forest algorithm is an ensemble learning method intended for classification tasks. Ensemble methods are based on the general principle that rather than relying on a single model, you can get better results and a more generalized model by creating multiple related models and combining them in some way. Generally, ensemble models provide better coverage and accuracy than single decision trees.

Learner parameters

The parameters that are specific to our learner is called the learner parameters. These are the knobs and handles of our learner which need to tune in order to extract the best out of our learner. These parameters vary with the learner.



1. Decision tree is a single instance/ single learner of a **Decision forest**. So, Decision forest is a collection of multiple decision trees.
2. **Depth** specifies how many levels each tree can grow to. A split means that features in each level of the tree (node) are randomly divided.
3. **Minimum number of samples per leaf node,** indicate the minimum number of cases that are required to create any terminal node (leaf) in a tree.



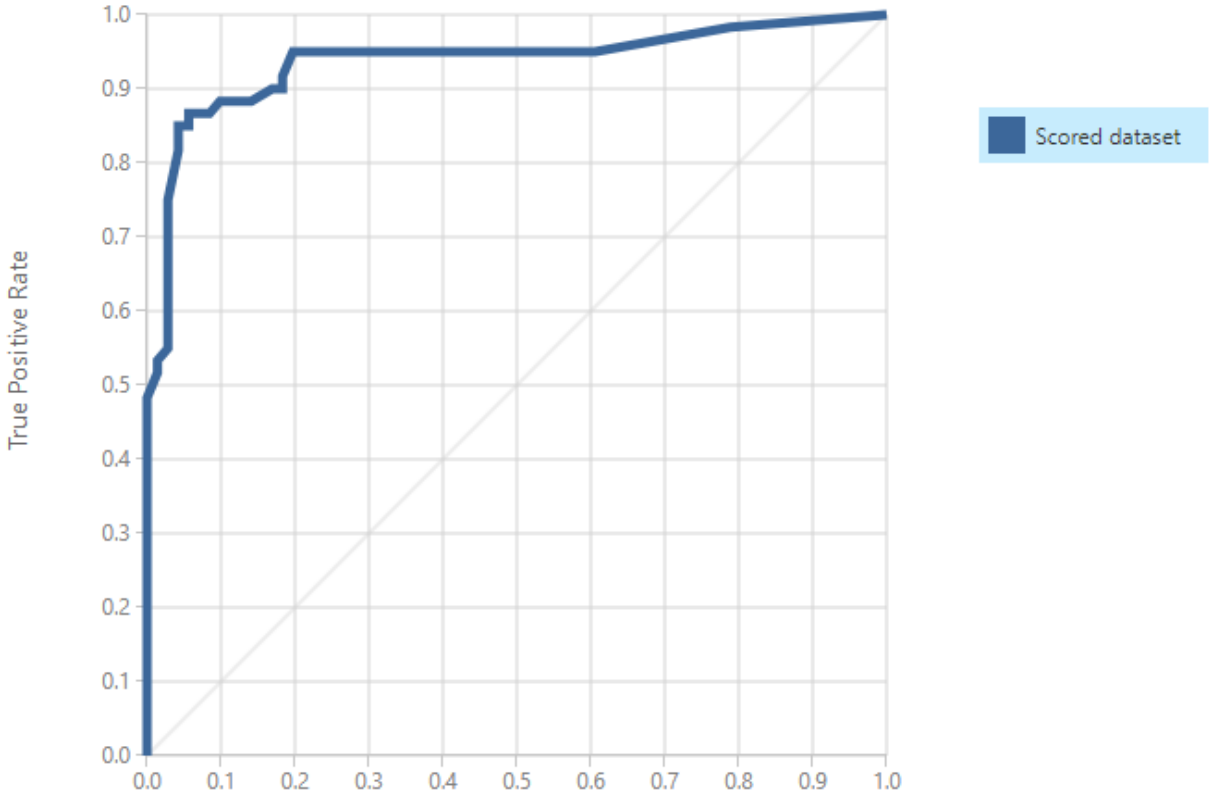
Scoring

Predictions from the classification algorithm can be obtained using the generic Score Model module. The module scores predictions for a trained classification or regression model.

Results

AUC - ROC curve is a performance measurement for classification problem at various thresholds settings. ROC is a probability curve and AUC represent degree or measure of separability. It tells how much model can distinguish between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between classes.

The below graph shows the ROC curve for our learner. The AUC Score of our learner is 93.4% which is very good. This shows our model is able to accurately distinguish 93 out of 100 samples that they belong to their original class.



The below table is called the confusion matrix. It represents the number of True positives, True negatives, False positives and False negatives counts of our model on the test set. Ideally, the number of True positives and True negatives should be maximum.

True Positive	False Negative
51	9
False Positive	**True Negative**
4	67

True Positive and True Negative of our learner is 51 and 67 which shows our learner can distinguish between the two classes.

Accuracy	Precision
0.901	0.927
Recall	**F1 Score**
0.850	0.887

Informally, accuracy is the fraction of predictions our model got right. Precision refers to the percentage of your results which are relevant. On the other hand, recall refers to the percentage of total relevant results correctly classified by your algorithm.

Credits

- https://www.wikipedia.org/
- https://docs.microsoft.com/en-us/azure/machine-learning/studio-module-reference/two-class-decision-forest
- https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
- https://gallery.azure.ai/Experiment/Anomaly-Detection-Credit-Risk-5
- https://developers.google.com/machine-learning/crash-course/classification/accuracy