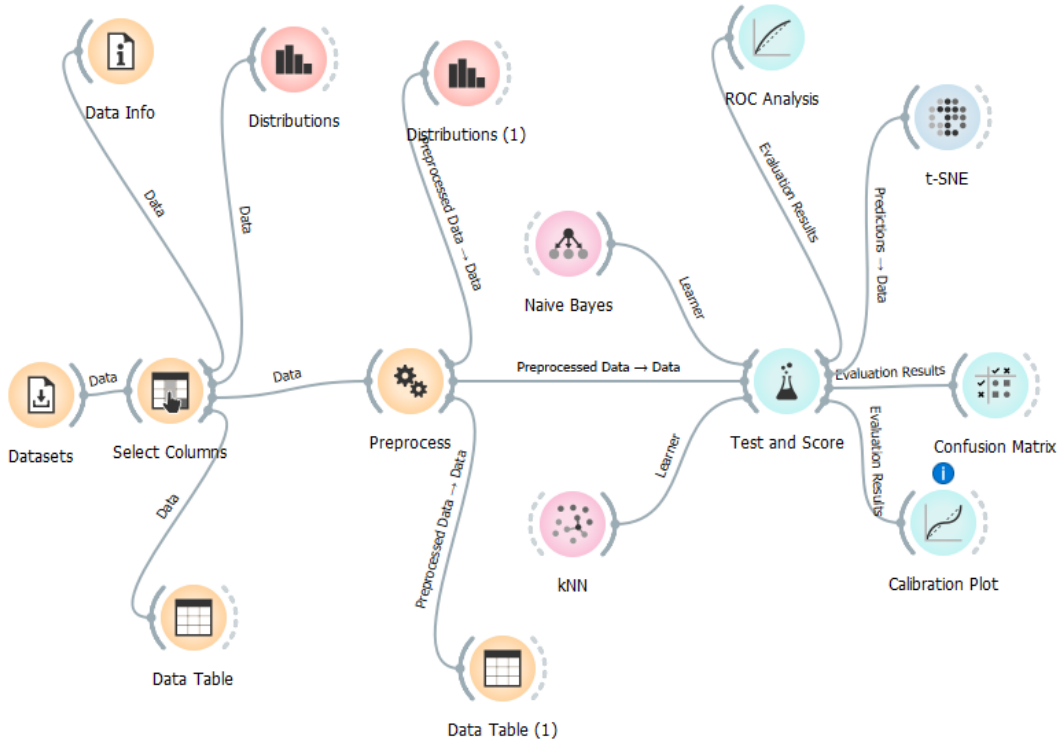
Anastasia Psarou - Assignment 7 - Machine Learning

This is my dataflow:



I used the “wine” dataset.

As far as the Naïve Bayes is concerned these are the results I get in the Test and Score widget:

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
Naive Bayes	0.157	0.031	0.999	0.978	0.977	0.978	0.978	0.083	0.989

Implementing different parameters in kNN widget we can see the below results in the Test and Score widget:

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.081	0.070	0.980	0.949	0.949	0.954	0.949	1.015	0.977

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.067	0.075	0.996	0.972	0.972	0.973	0.972	0.244	0.987

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.077	0.064	0.969	0.927	0.926	0.930	0.927	1.448	0.965

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.074	0.128	0.977	0.933	0.932	0.939	0.933	1.105	0.971

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.077	0.045	0.980	0.949	0.949	0.954	0.949	1.015	0.977

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.083	0.048	0.996	0.972	0.972	0.973	0.972	0.243	0.987

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.092	0.053	0.970	0.927	0.926	0.930	0.927	1.445	0.965

Model	Train time [s]	Test time [s]	AUC	CA	F1	Precision	Recall	LogLoss	Specificity
kNN	0.077	0.118	0.978	0.933	0.932	0.939	0.933	1.100	0.972

From these results we can observe that whichever parameter we use in the kNN widget the Classification Accuracy is not be bigger than the one of the Naïve Bayes widget. The same think happens also with F1, Precision, AUC and Specifity. Also, we can see that in Naïve Bayes the Train Time is almost twice as large compared to the kNN model while the Test Time is almost half in kNN compared to Naïve Bayes.

Feature engineering is the process of taking raw data and transforming it into features that can be used in machine learning algorithms. This process helps data scientists choose the features that will make the model accurate. So, better data quality can produce better quality results. Also, feature engineering helps understand what is important in the data.