**Department of Computer Engineering**

**Academic Year:** 2022-2023 **Semester:** VIII

**Subject:** Applied Data Science **Class / Division:** BE/CMPN/B

**Name:** Domingos J Pinto **Roll Number:** 007

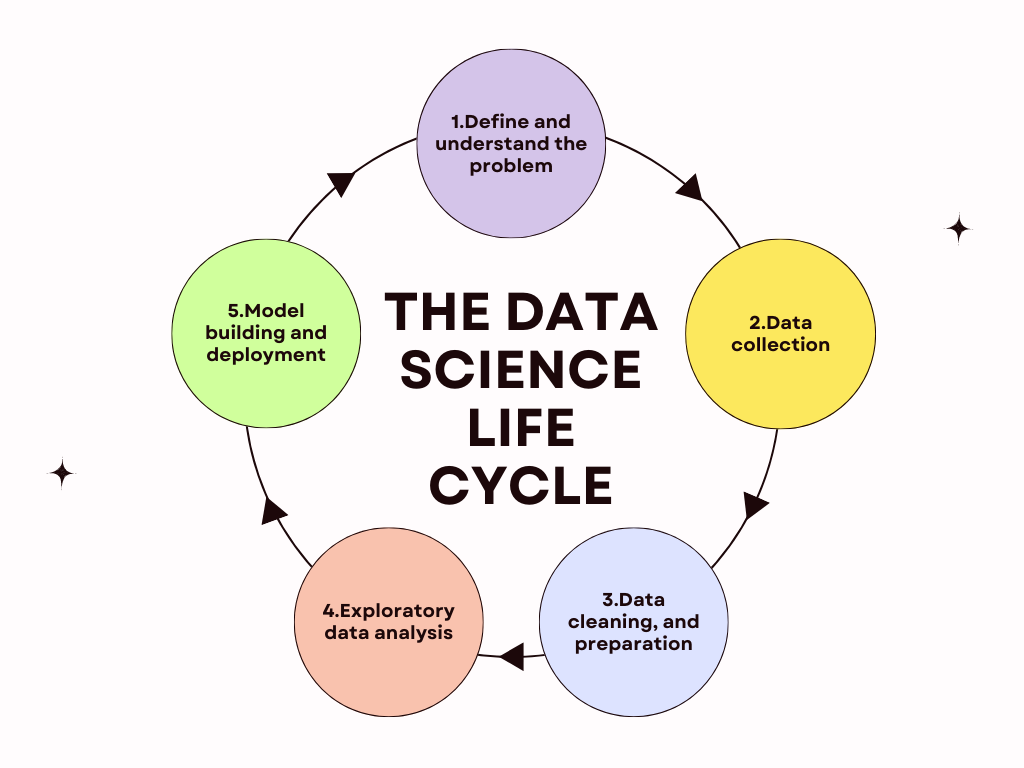
**Experiment No.: 9**

**Illustrate data science lifecycle for selected case study.**

**Aim :** Illustrate data science lifecycle for selected case study stock prediction.

**I THEORY**

Stock Price Prediction using machine learning helps you discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.

****

**Fig 1. Data science lifecycle process**

The data science life cycle for stock prediction may involve the following stages:

1. Problem definition: the goal is to predict the future price of a stock based on historical data and market trends.
2. Data collection: obtaining financial data such as stock prices, trading volume, news articles, economic indicators, etc.
3. Data cleaning and pre-processing: handling missing values, dealing with outliers, and formatting the data.
4. Exploratory data analysis: summarizing the main characteristics of the data, identifying relationships between features and target variable, and discovering patterns.
5. Feature engineering: transforming and creating new features from the existing ones to improve model performance.
6. Modeling: selecting and training a machine learning model such as time series forecasting, support vector regression, or deep learning on the pre-processed data.
7. Model evaluation: measuring the performance of the model using metrics such as mean absolute error, mean squared error, or accuracy.
8. Deployment: integrating the model into a web application or API for real-time predictions.
9. Monitoring and maintenance: continuously monitoring the model's performance, updating it with new data, and tuning its parameters to improve accuracy.

**RapidMiner**

RapidMiner is a powerful data science platform that provides an integrated environment for data preparation, machine learning, and model deployment. It's designed to help organizations and individuals solve complex data problems and create predictive models with minimal effort.

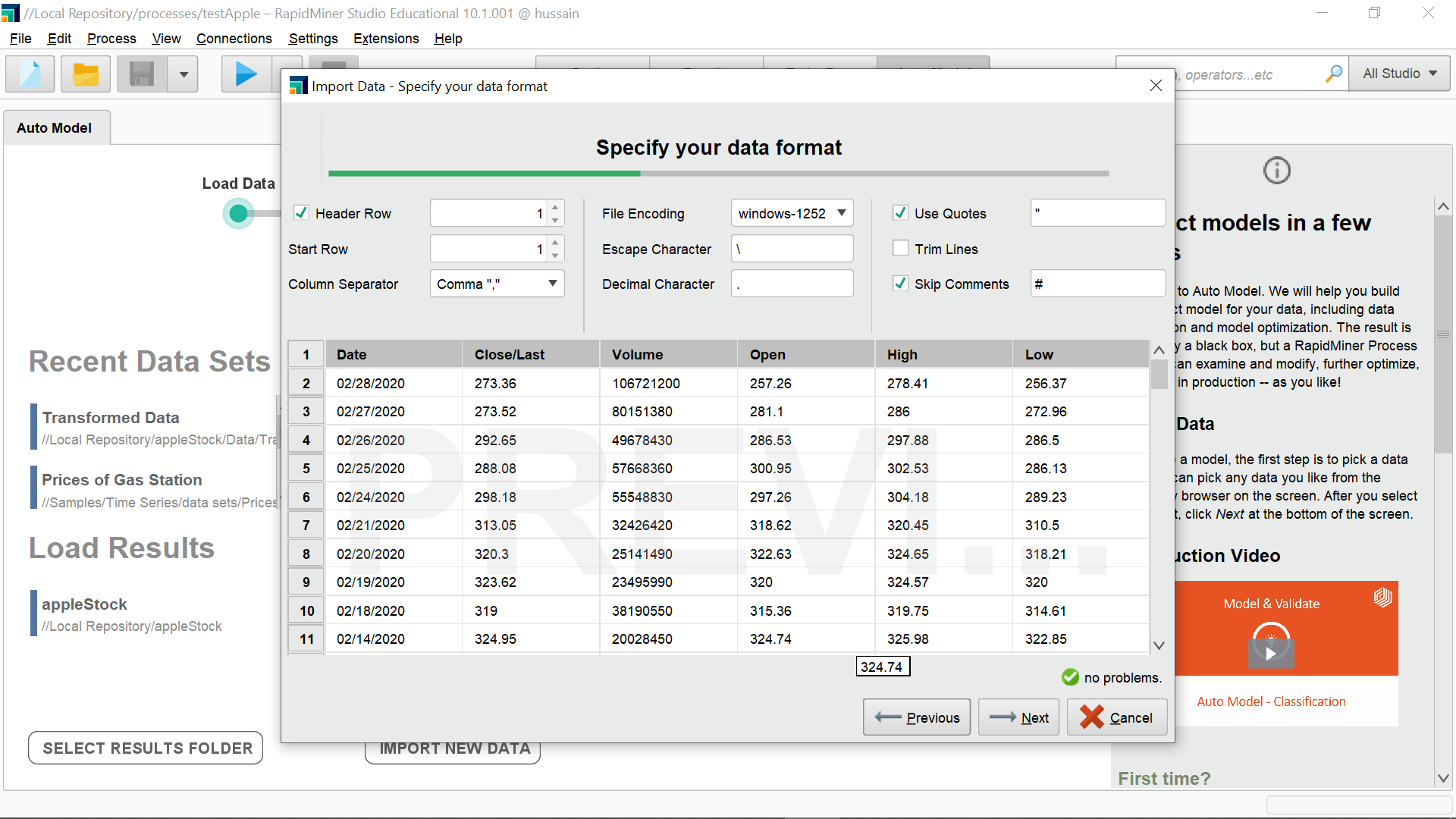
RapidMiner provides a user-friendly interface and a visual programming language called RapidMiner Studio, which makes it easy to perform data analysis, model building, and deployment without writing complex code. The platform also includes a library of pre-built templates, operators, and algorithms that can be used to build models for a variety of use cases, such as classification, regression, clustering, and association rule mining.

**II IMPLEMENTATION**

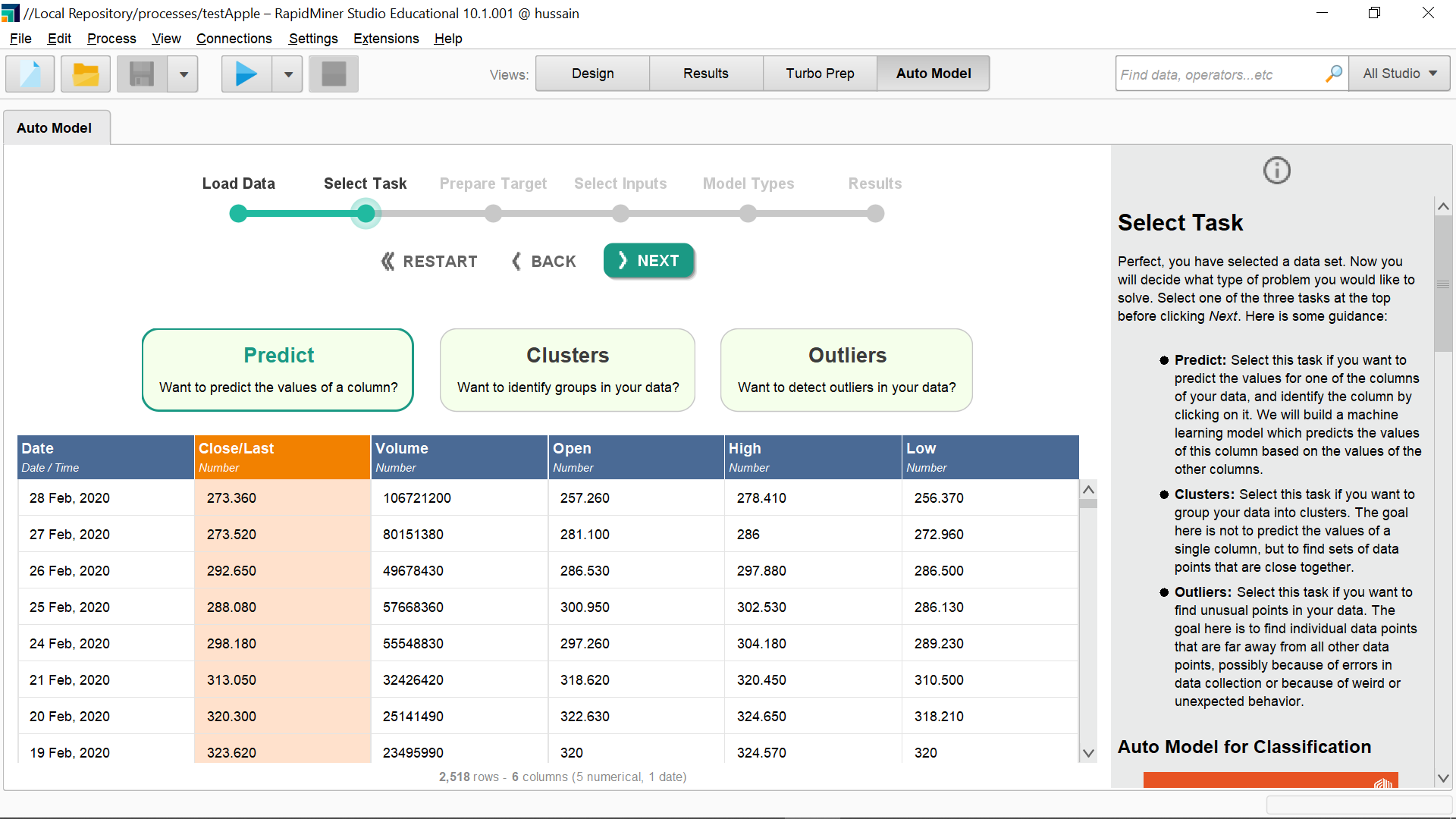
**Dataset :** <https://www.kaggle.com/datasets/tarunpaparaju/apple-aapl-historical-stock-data>

**Description :** The dataset contains Apple's (AAPL) stock data for the last 10 years (from 2010 to date).

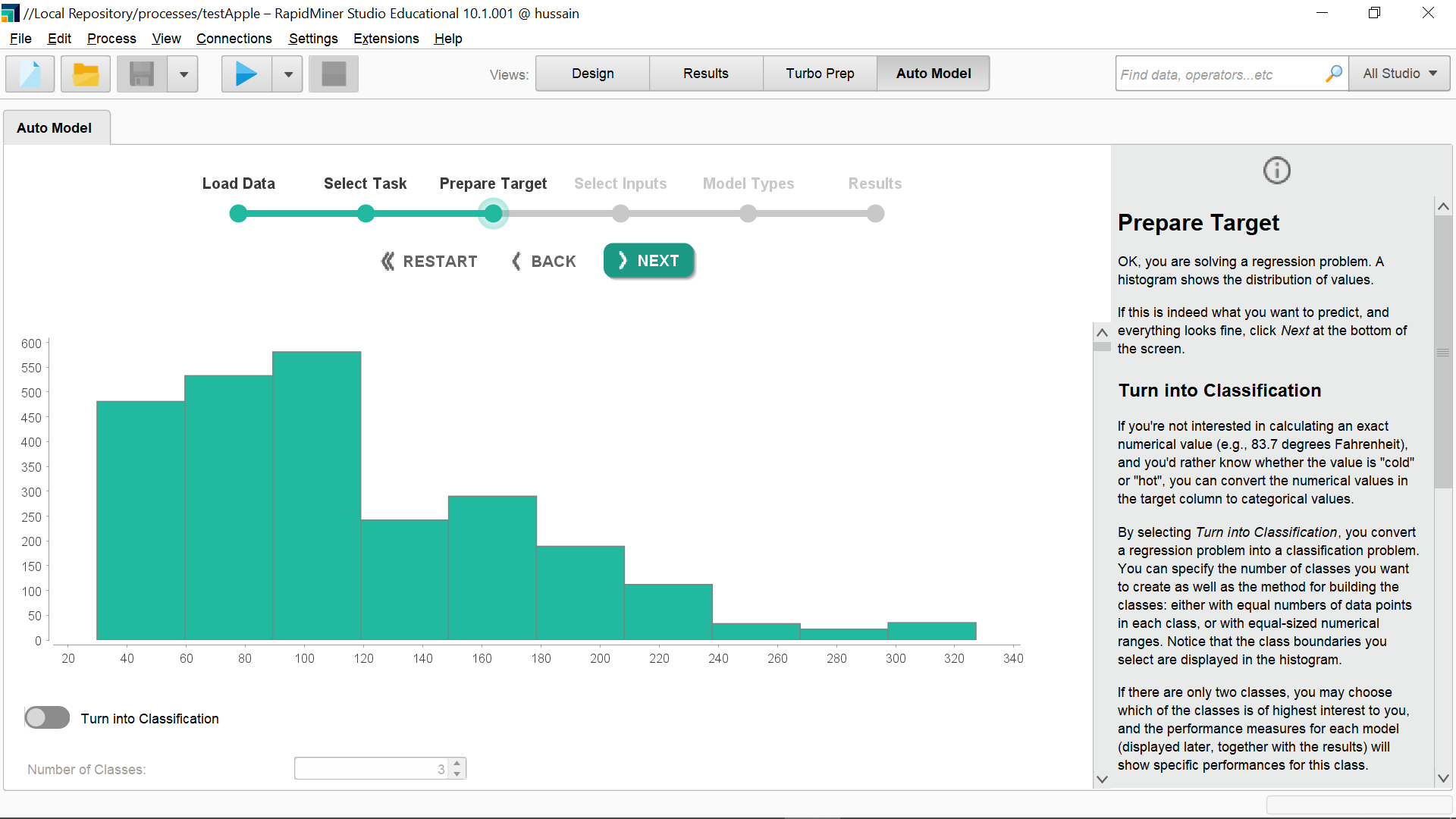
1. Loading the data in RapidMiner auto model



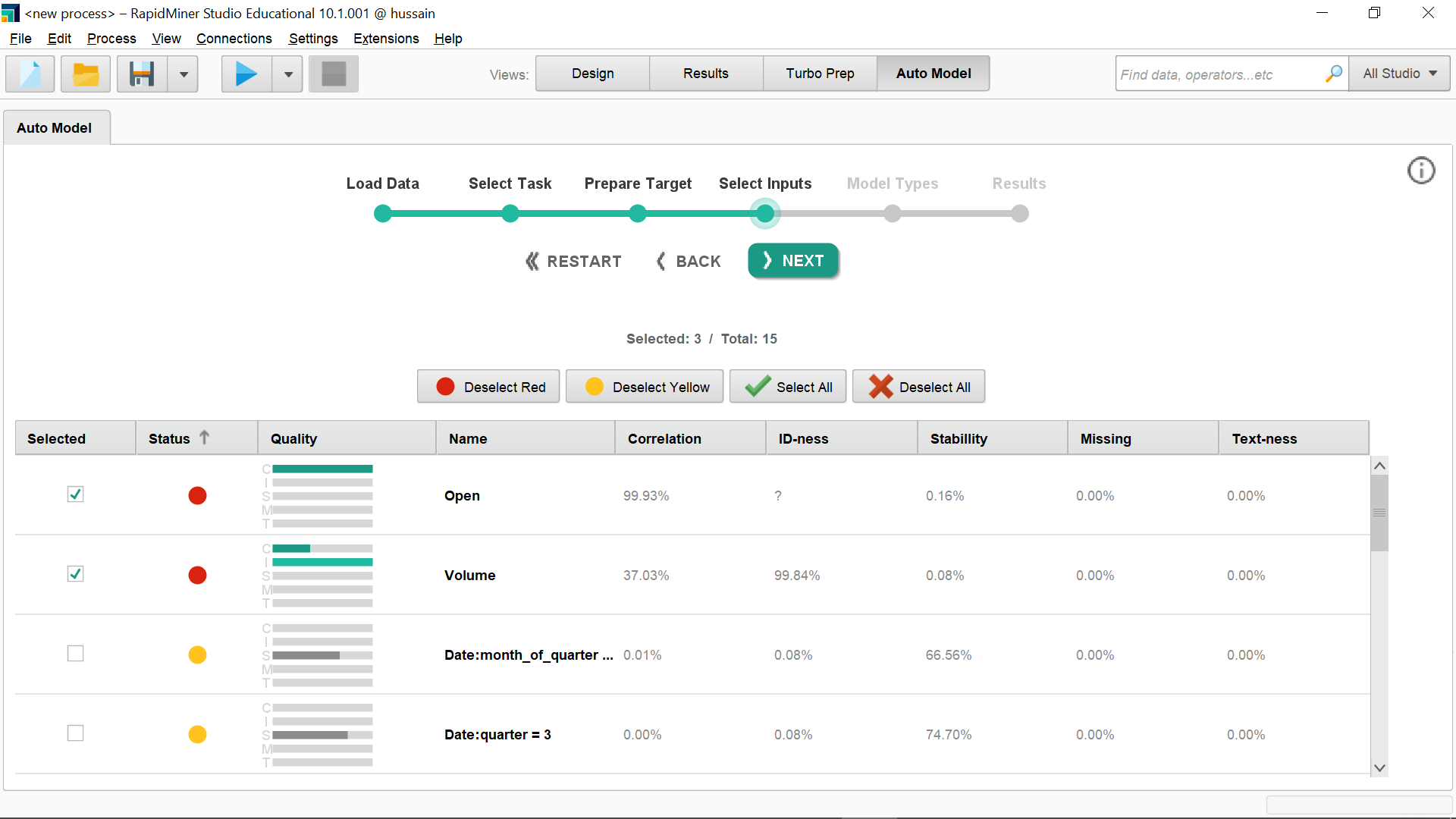
1. Selecting the target variable and what type of task need to be performed



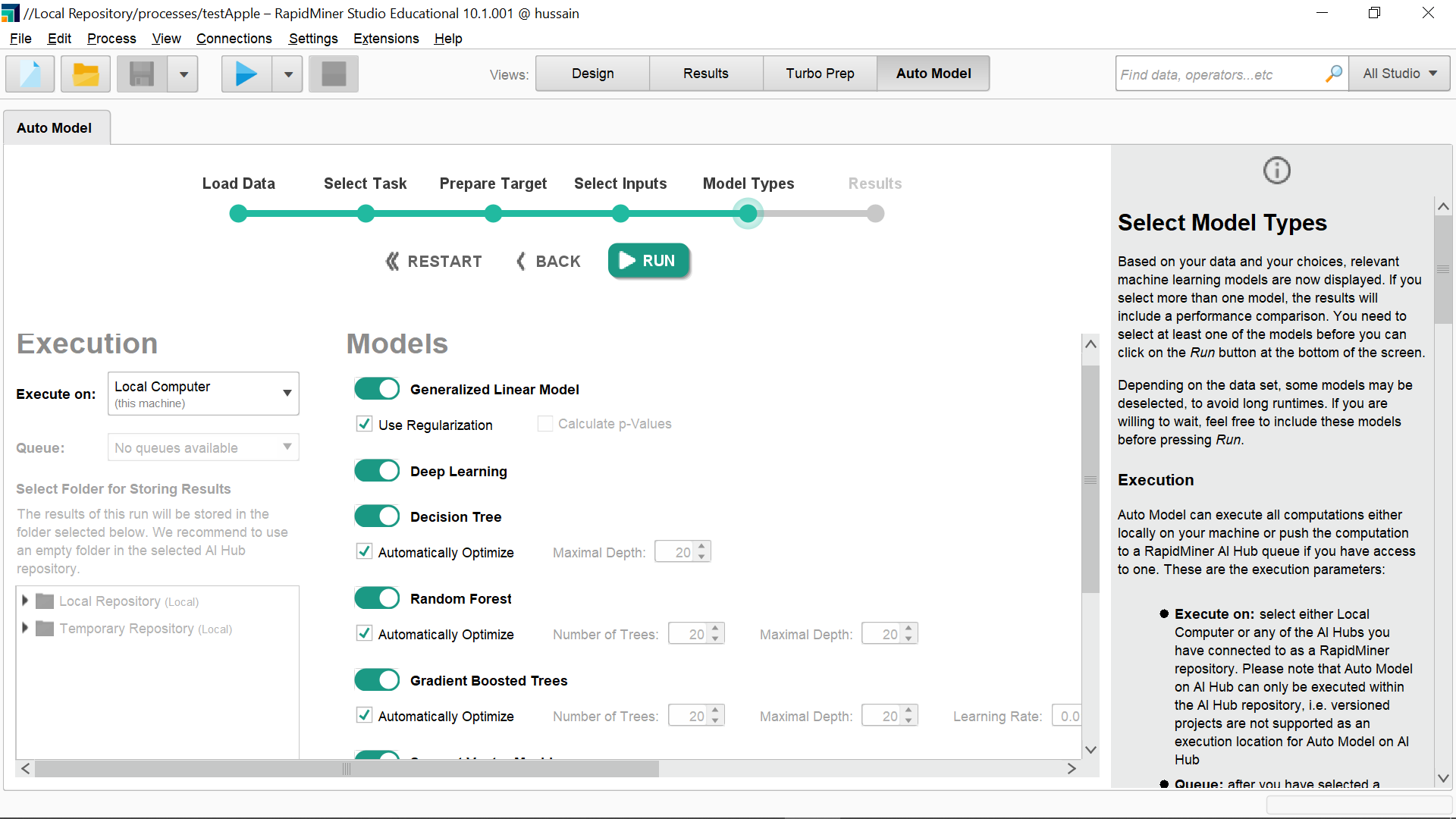
1. Preparing what type of target variable is and whether you are performing a classification



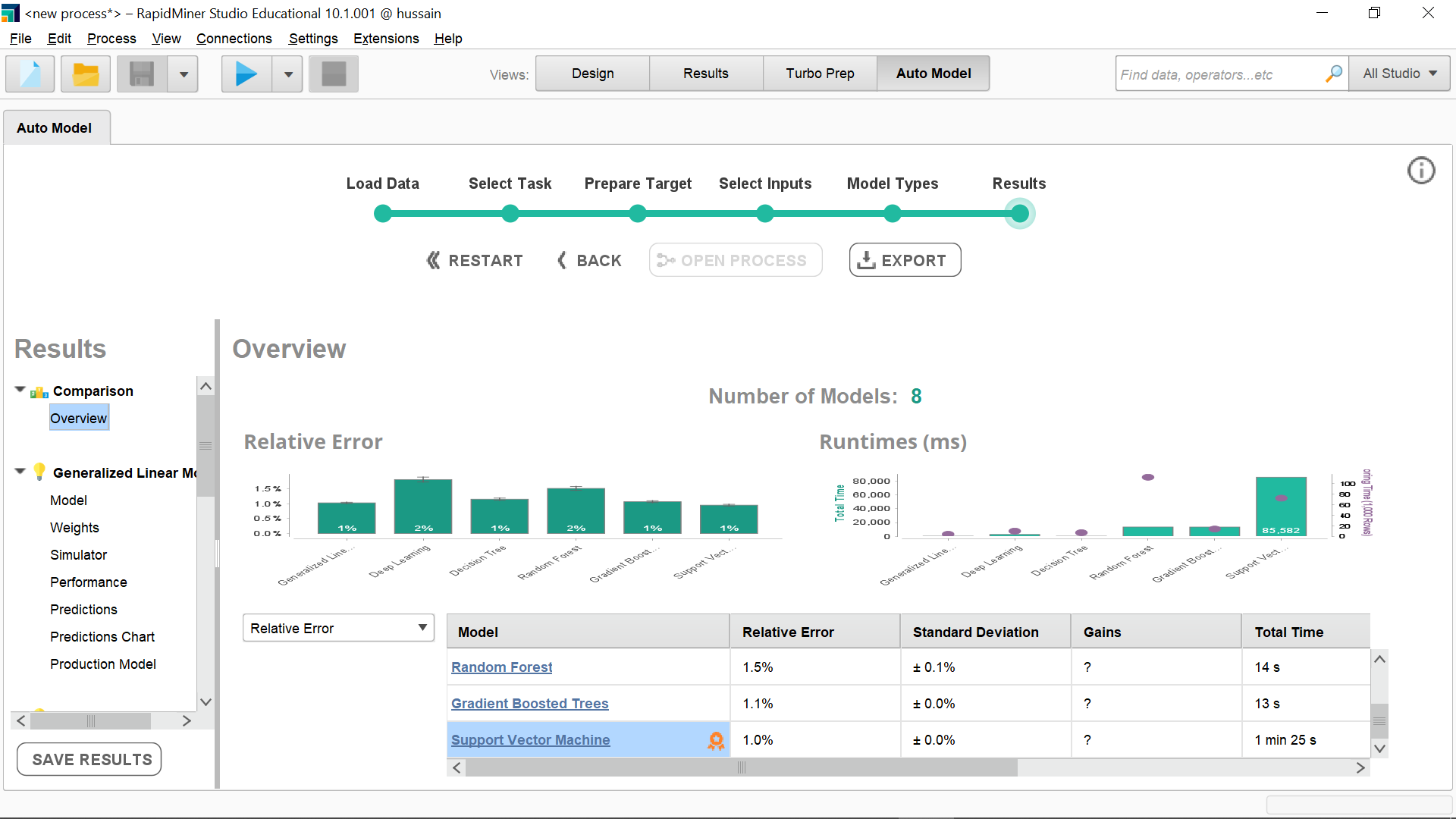
1. Selecting the inputs that will be considered to predict the target variable

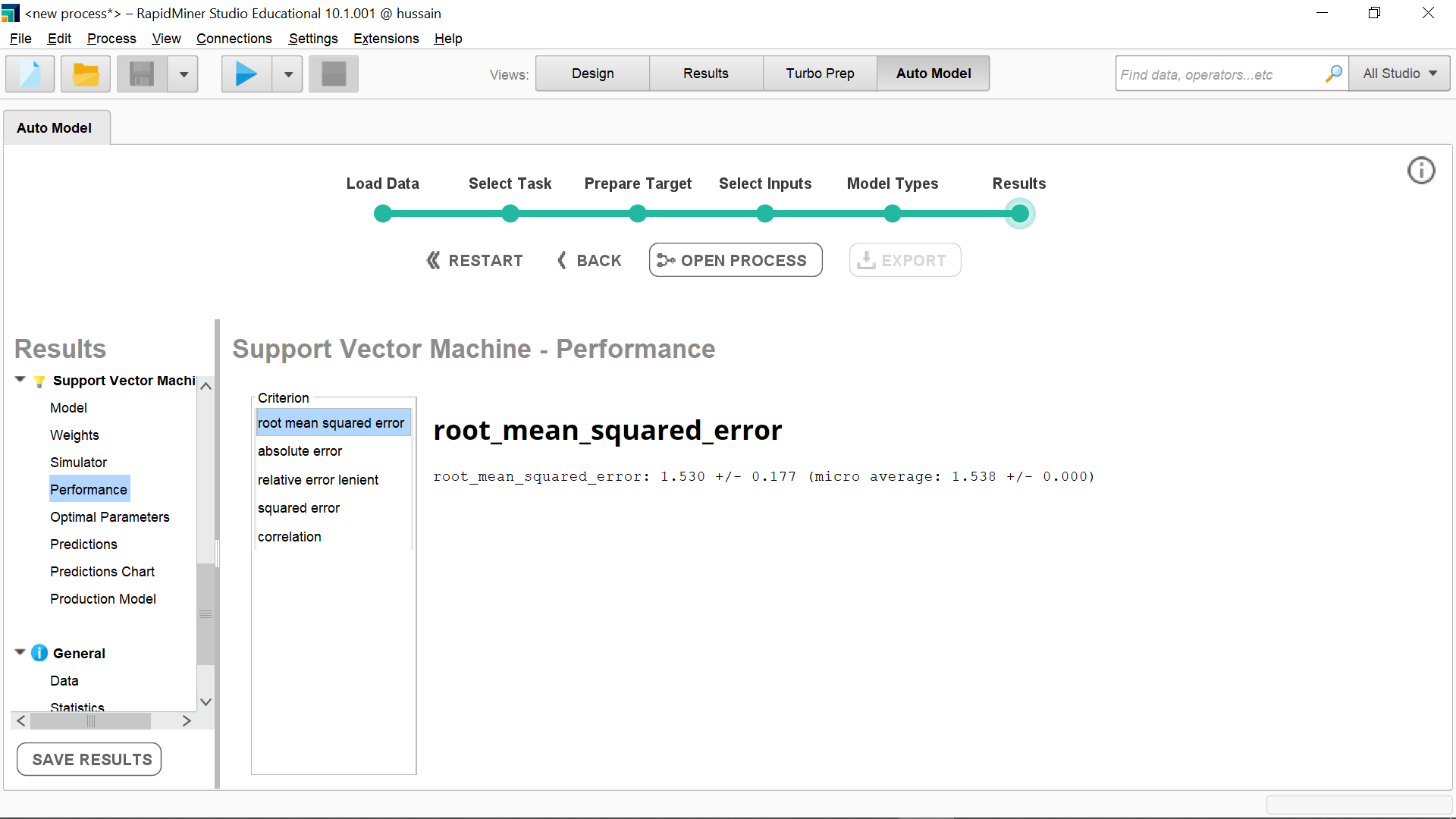


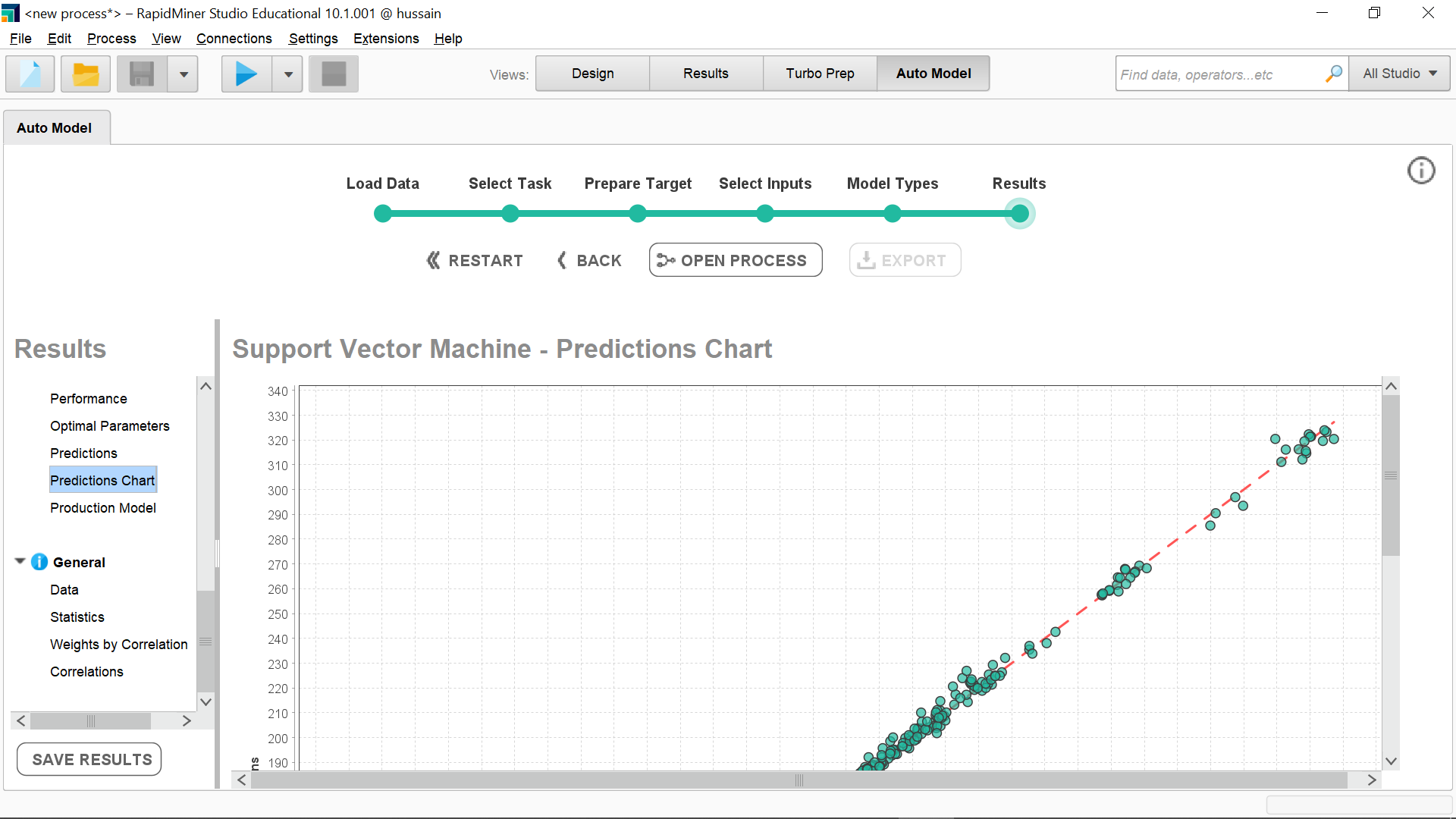
1. Selecting models to train on the data with their optimization parameters

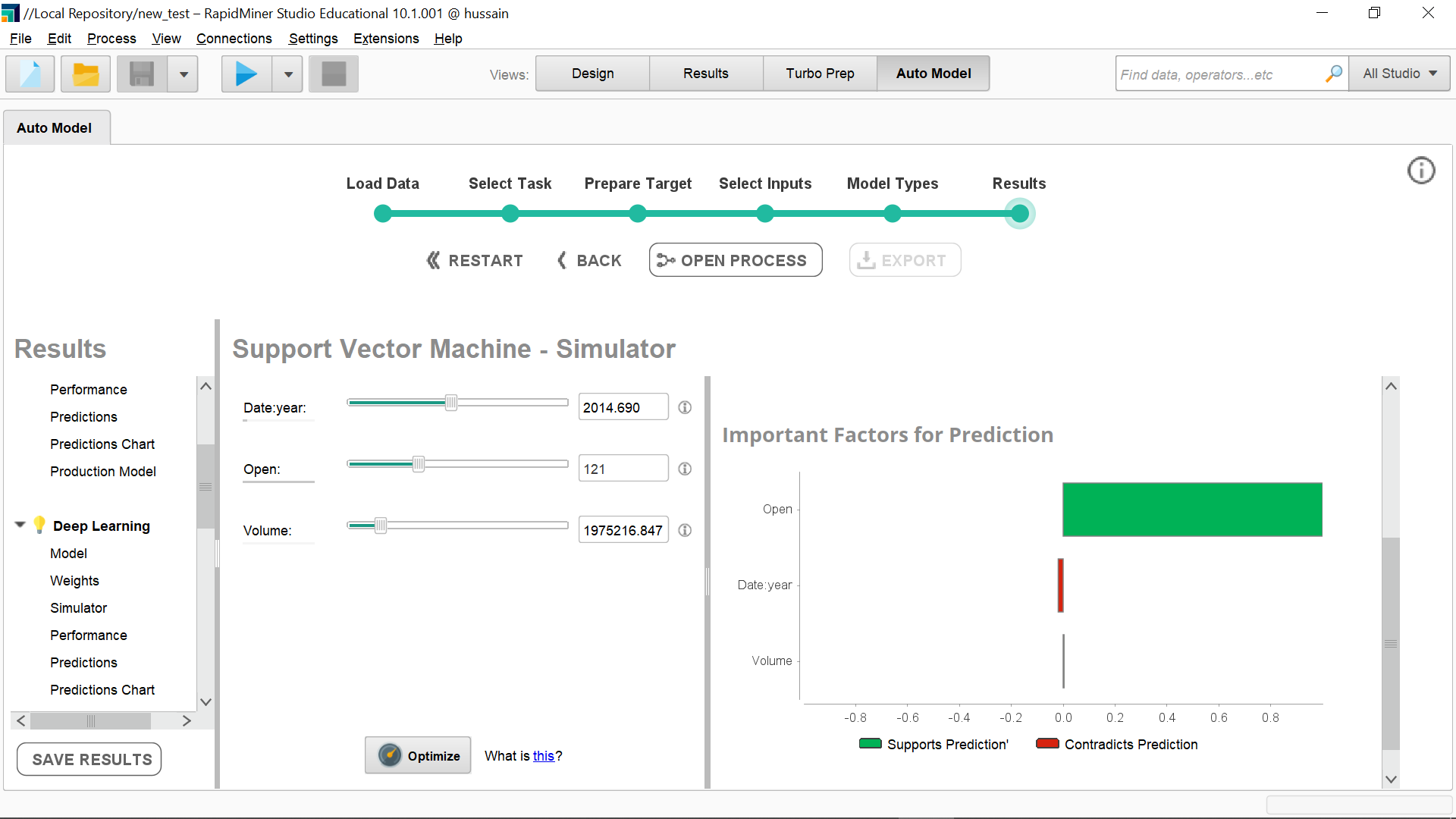


1. Comparing results of different models and checking model performance on various parameters

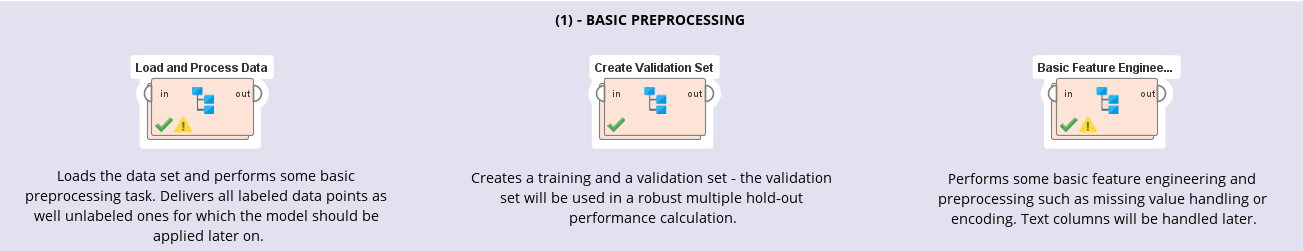


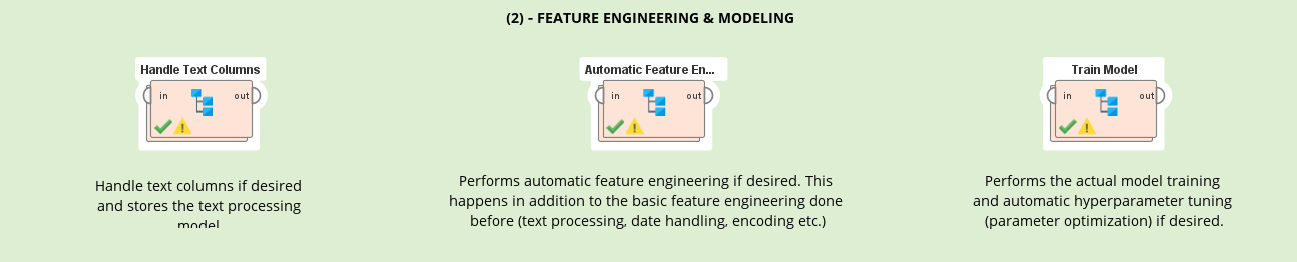


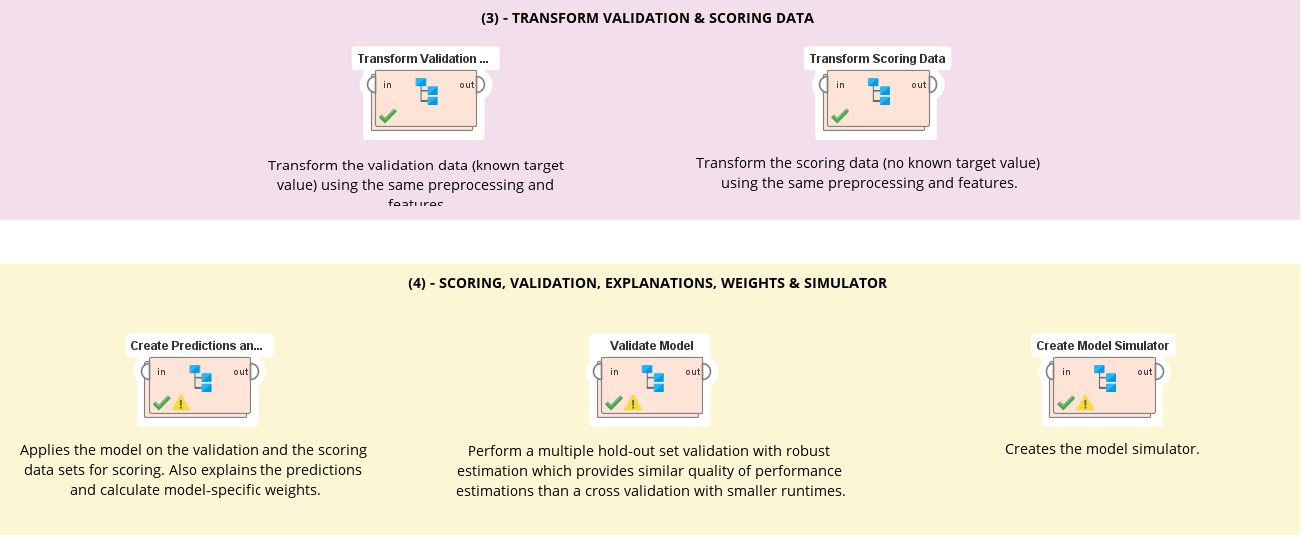


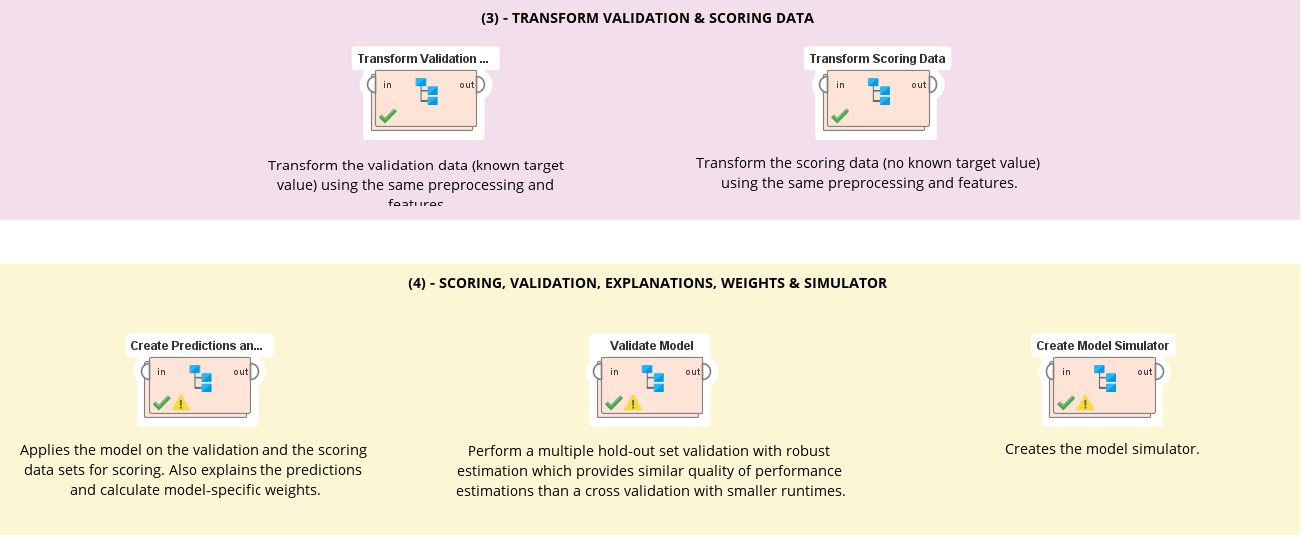


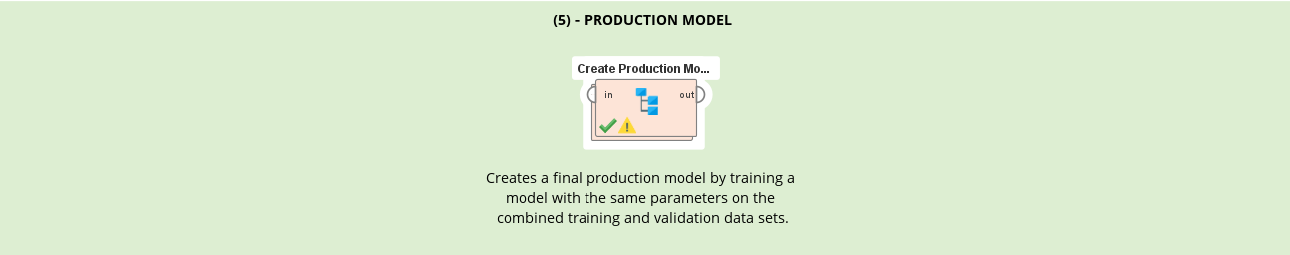
7. Checking the process of best performing model(SVM)











8. Chart of the prediction by the model

