Index

Creating a model to predict Employee's Attrition (Binary Classification task) using Data Science Platforms such as Azure ML Studio, RapidMiner, Knime Analytics)

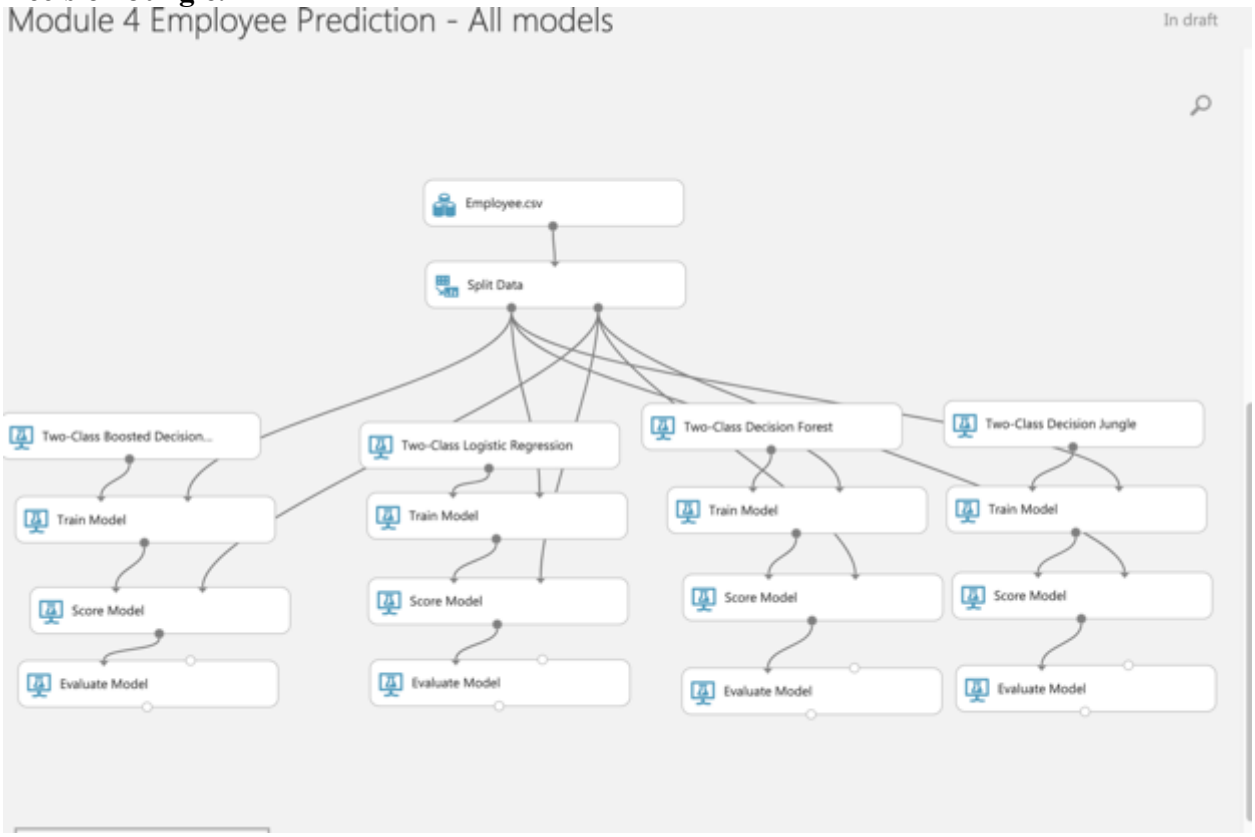
The task is to create predictive models using different Data Science platforms and compare the results of prediction choosing the best model for this particular task. I used the dataset from Kaggle.com https://www.kaggle.com/tejashvi14/employee-future-prediction
The task is to predict if an employee will leave the company or will keep working in the same company in the next 2 years, the model will be created based on the dataset from Kaggle. The task is a Binary Classification task.

We have the following data for the model creation:

1. Education
2. JoiningYear
3. City
4. PaymentTier
5. Age
6. Gender
7. EverBenched
8. ExperienceInCurrentDomain
9. LeaveOrNot – target field

1. Azure Machine Learning Studio

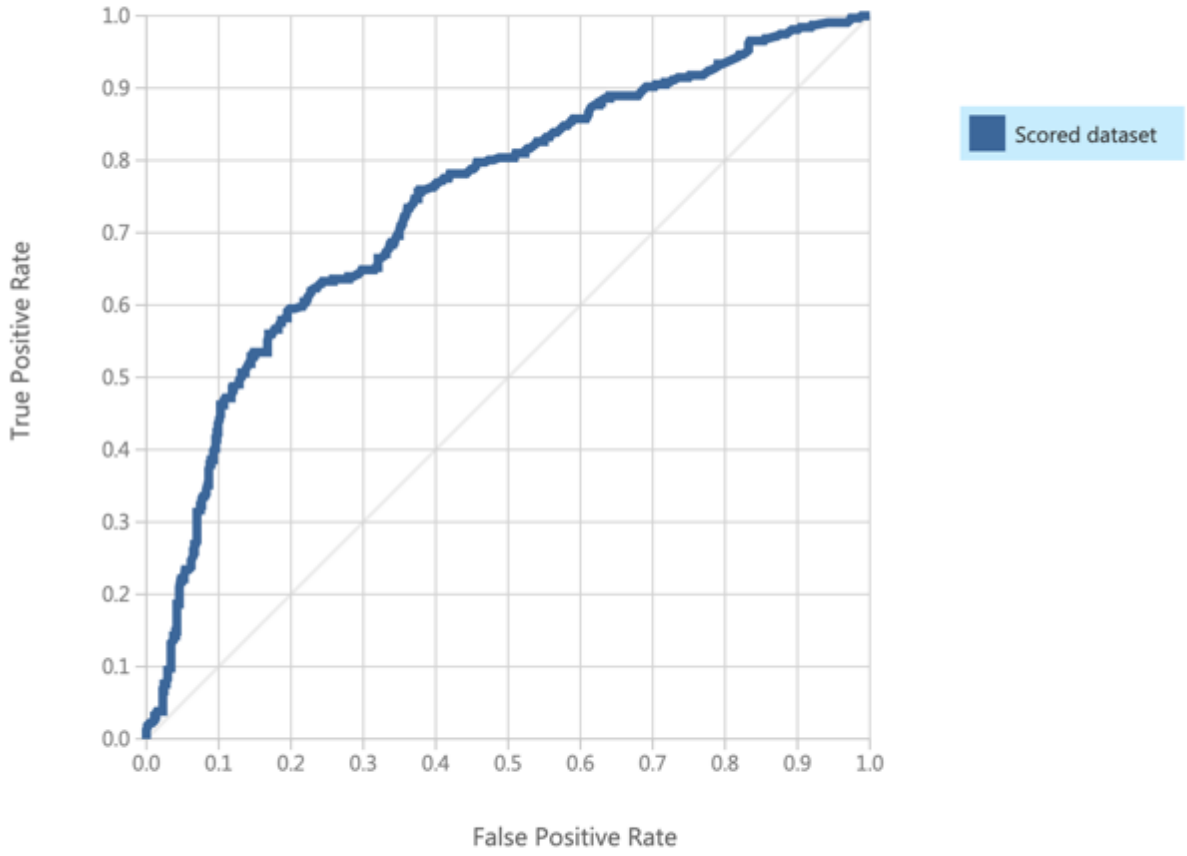
We will work with Azure ML Studio first.
Using Azure ML Studio (classic) we create different models, train the models, and evaluate them using the scoring metrics. The following models will be created: **Two-Class Logistic Regression, Two-Class Boosted Decision Tree, Two-Class Decision Forest, Two-Class Decision Jungle.**



Let’s compare the results we received after creating each model.

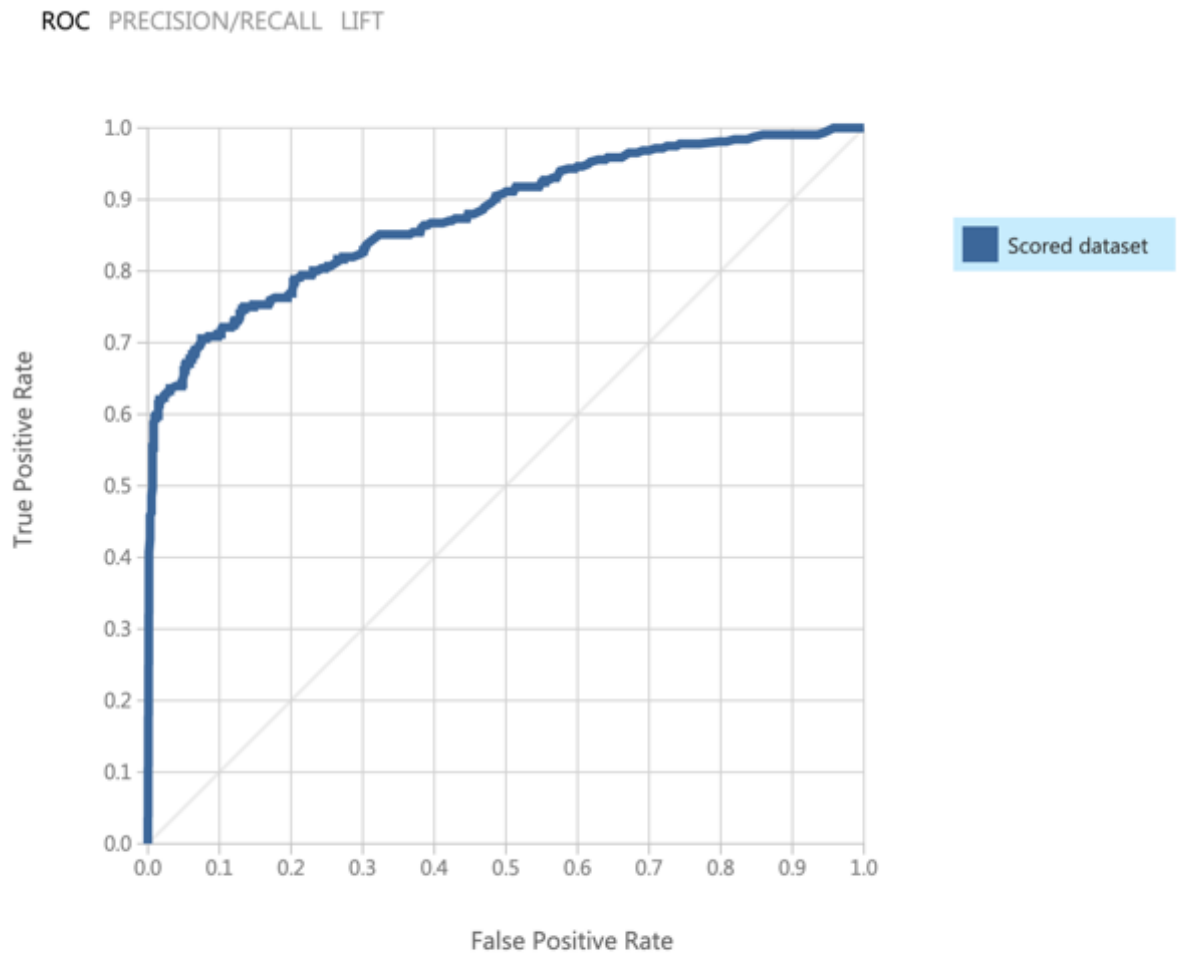
Two-Class Logistic Regression

ROC PRECISION/RECALL LIFT



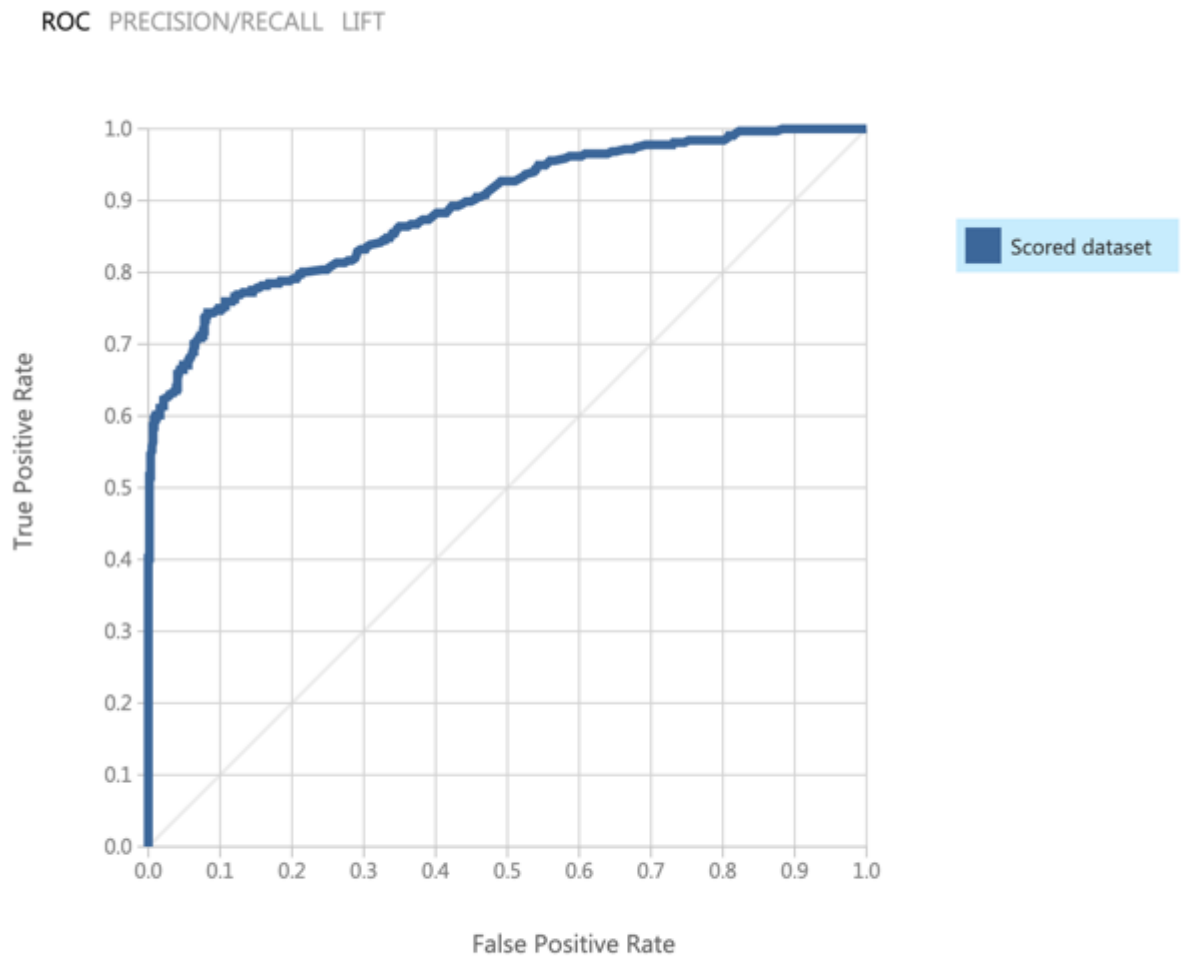
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
149	167	0.748	0.687	0.5	0.743

Two-Class Boosted Decision Tree



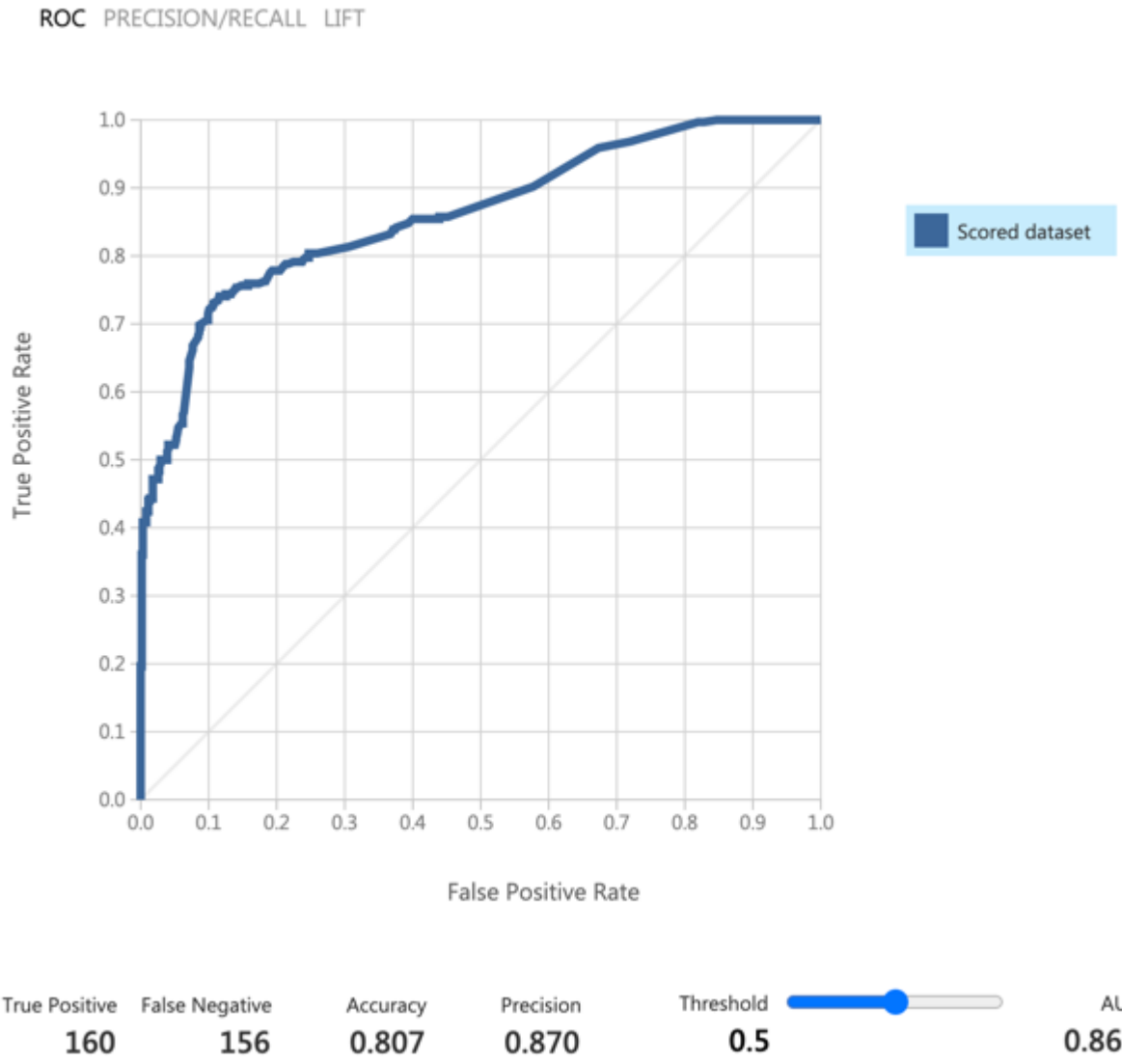
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
220	96	0.849	0.830	0.5	0.877

Two-Class Decision Forest



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
211	105	0.855	0.876	0.5	0.890

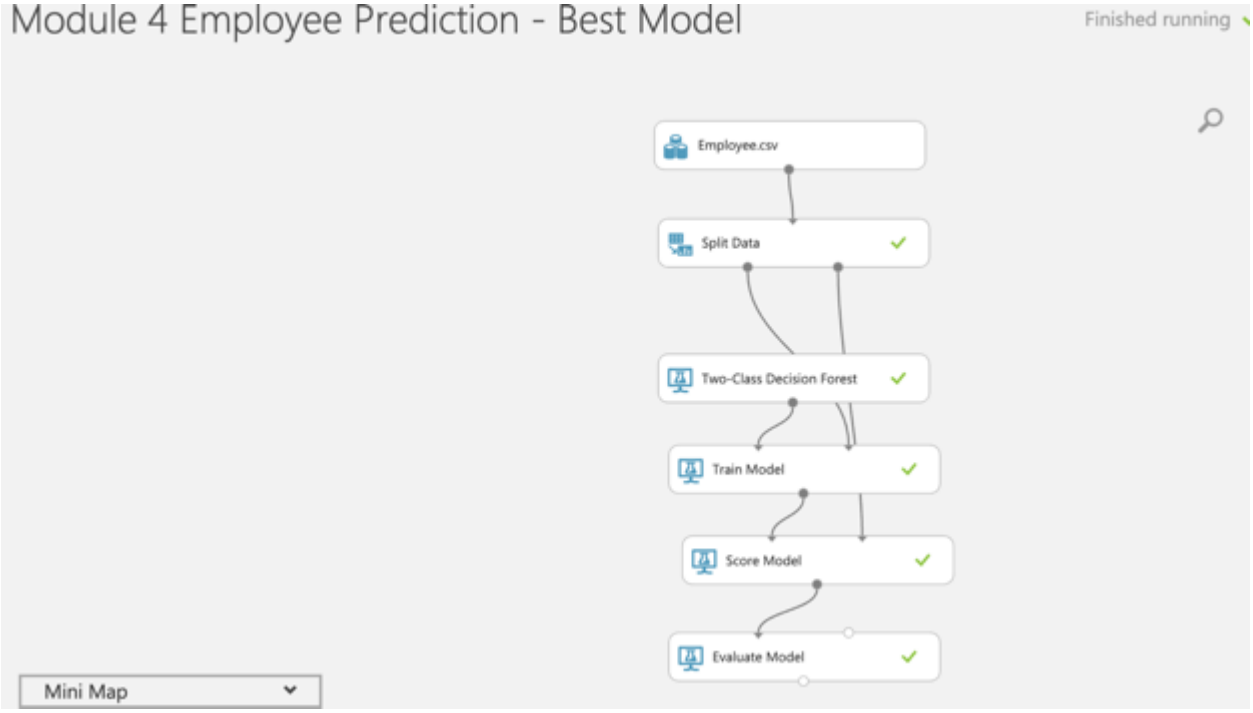
Two-Class Decision Jungle

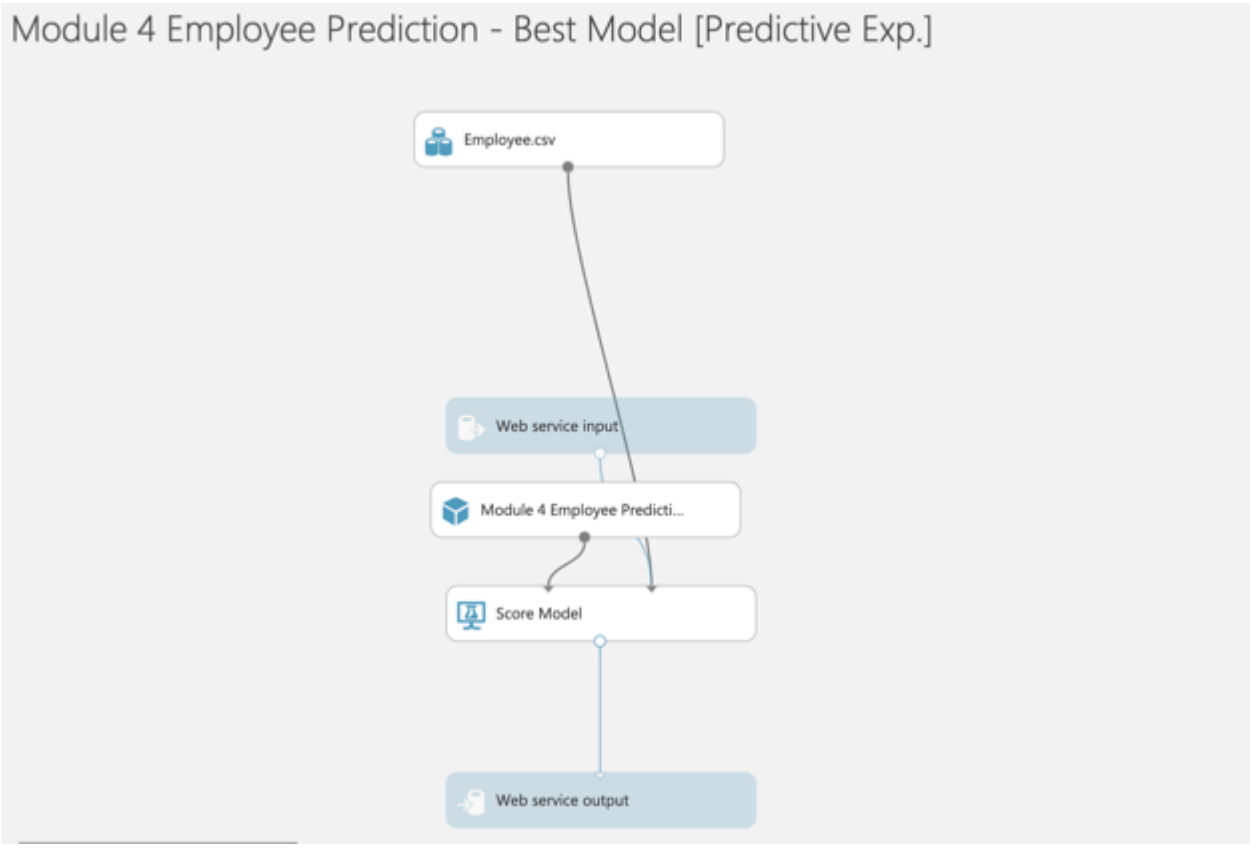


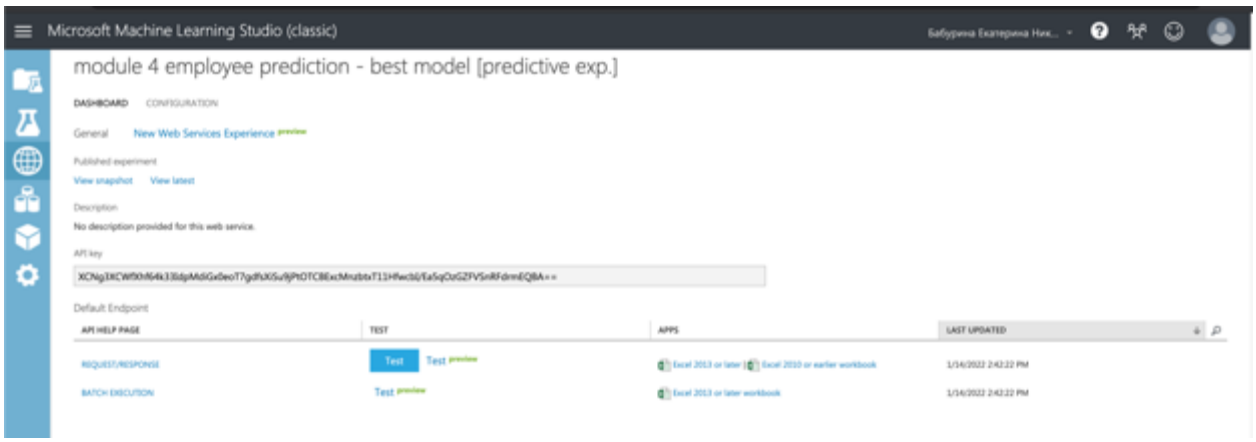
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
160	156	0.807	0.870	0.5	0.860

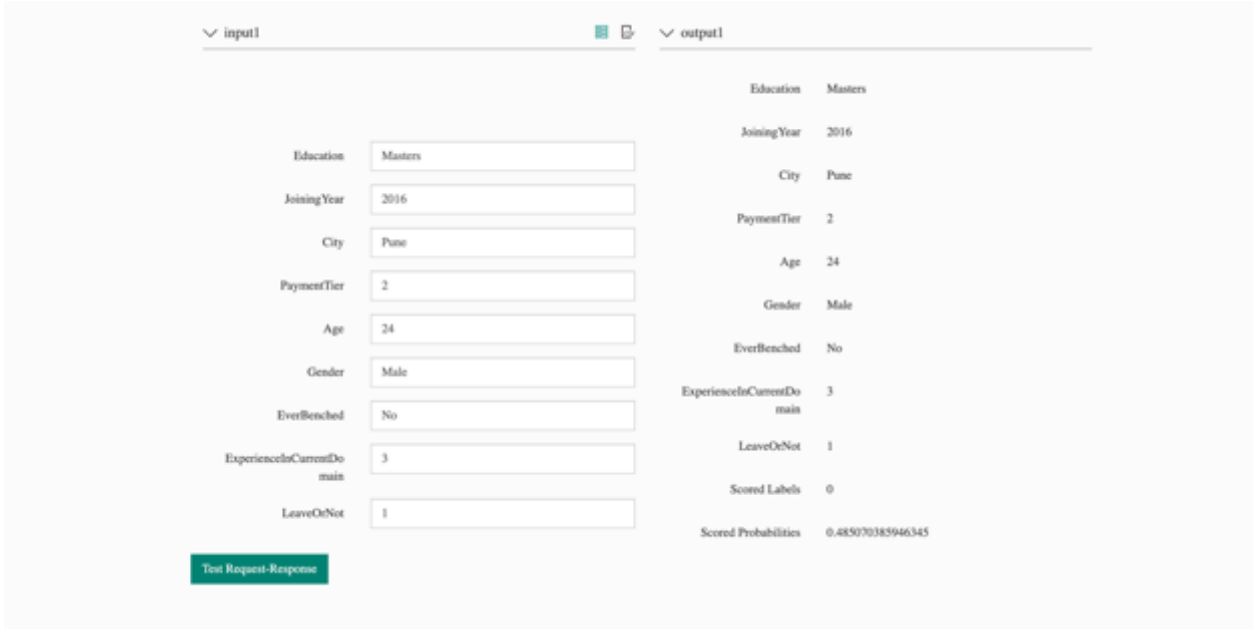
The best model is **Two-Class Decision Forest** with **AUC of 0.89**.

The Two-Class Decision Forest Model has been deployed as a web-service in Azure ML Studio (classic).


Module 4 Employee Prediction - Best Model
Finished running
Employee.csv
Split Data
Two-Class Decision Forest
Train Model
Score Model
Evaluate Model
Mini Map

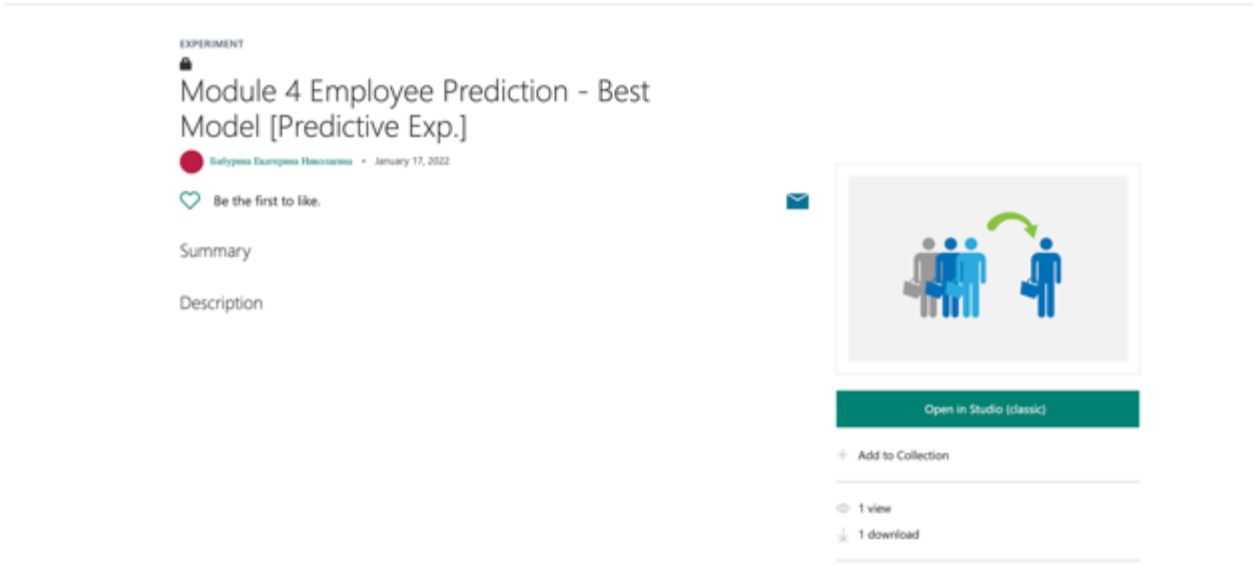

Module 4 Employee Prediction - Best Model [Predictive Exp.]
Employee.csv
Web service input
Module 4 Employee Predicti...
Score Model
Web service output





The experiment shared in the Gallery

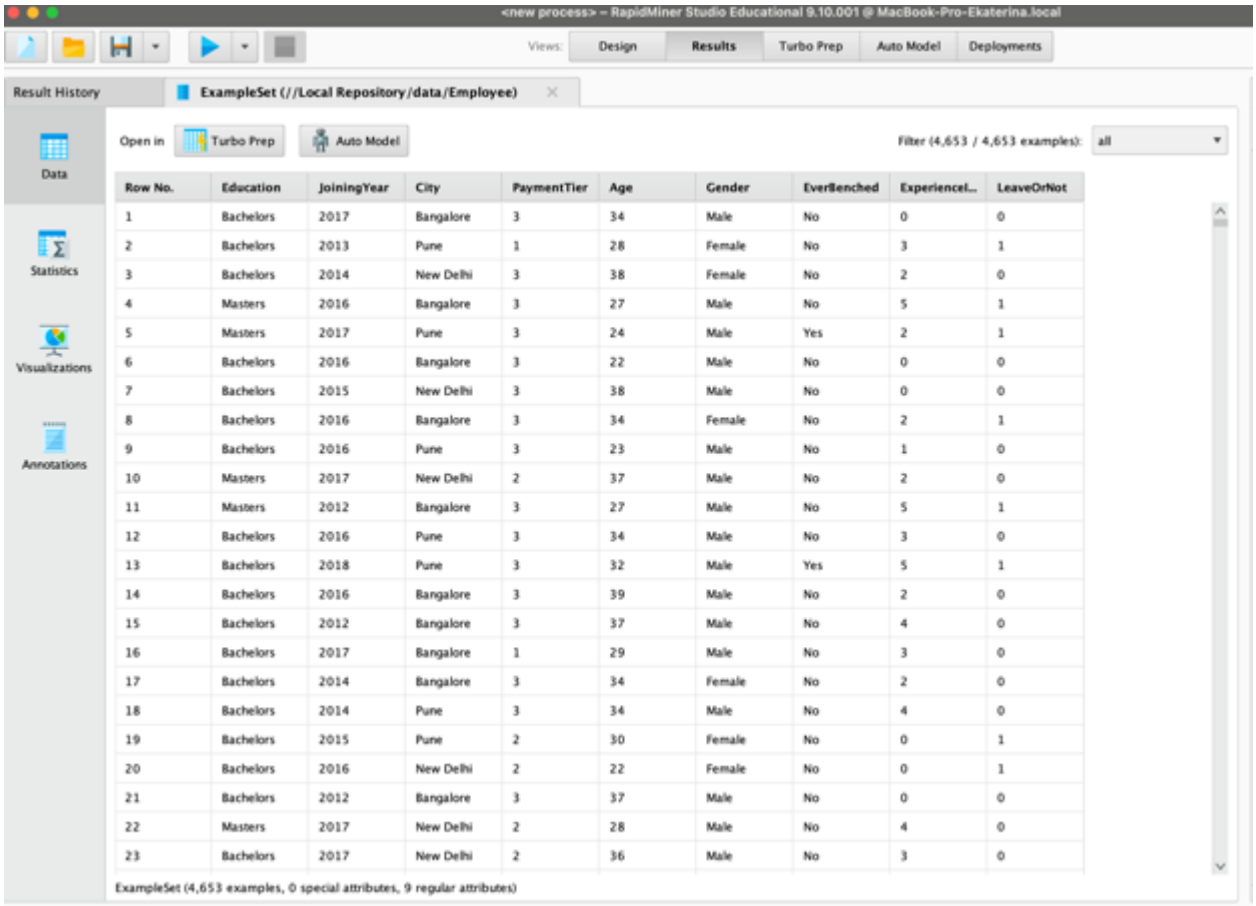
https://gallery.cortanaintelligence.com/Experiment/Module-4-Employee-Prediction-Best-Model-Predictive-Exp

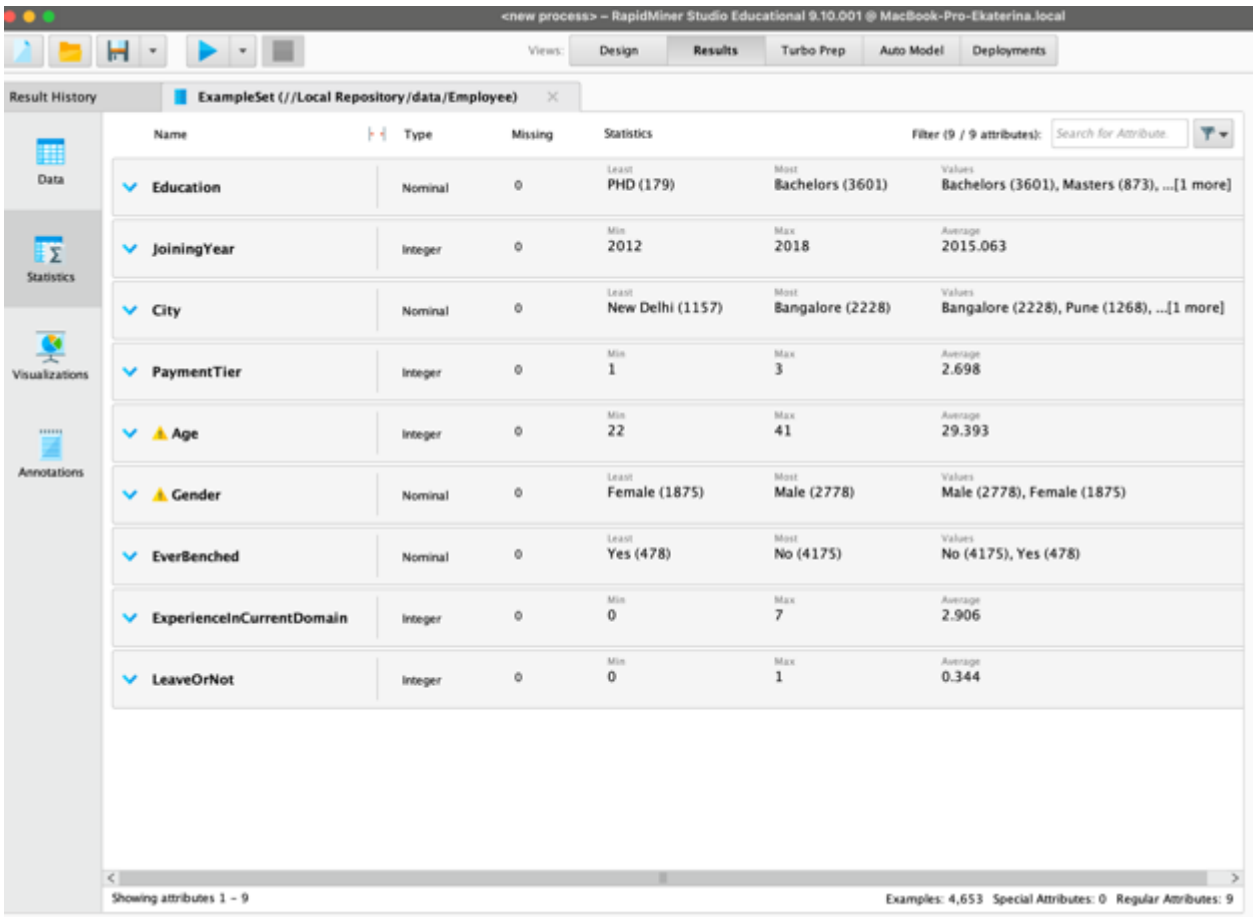


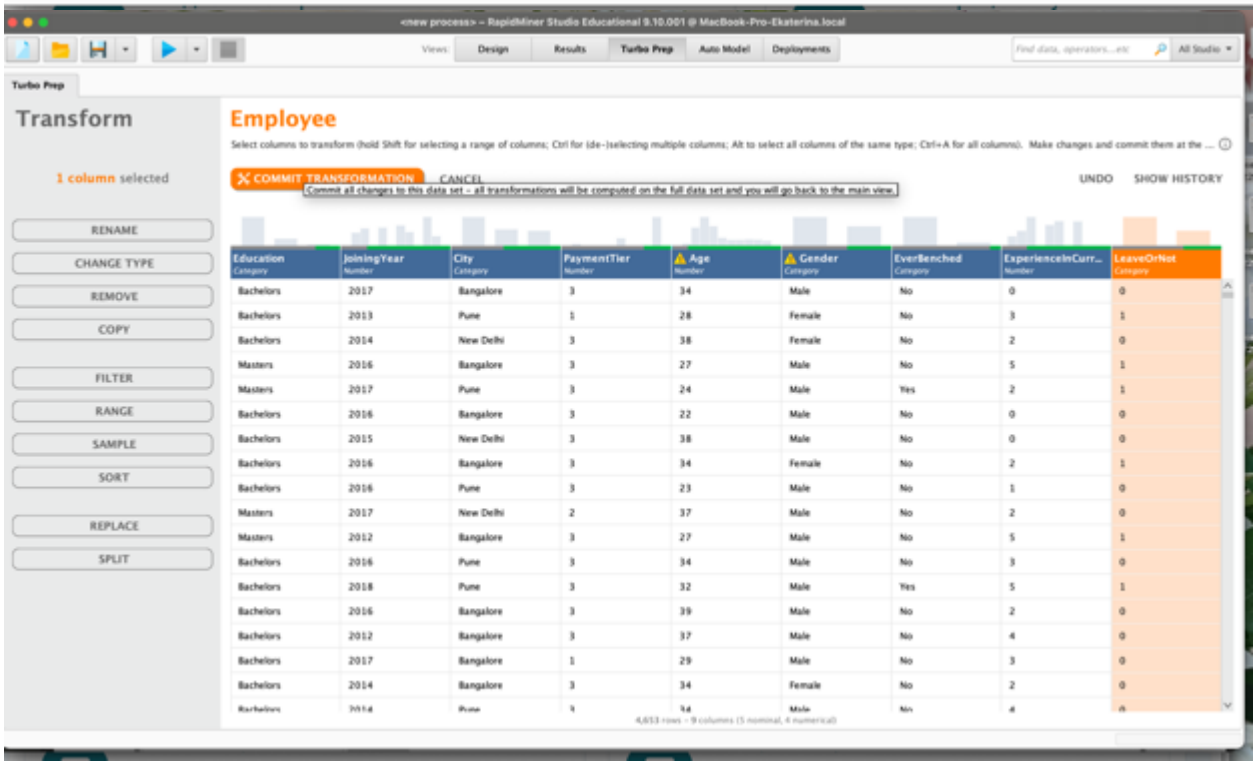
2. RapidMiner Studio

Data Preparation

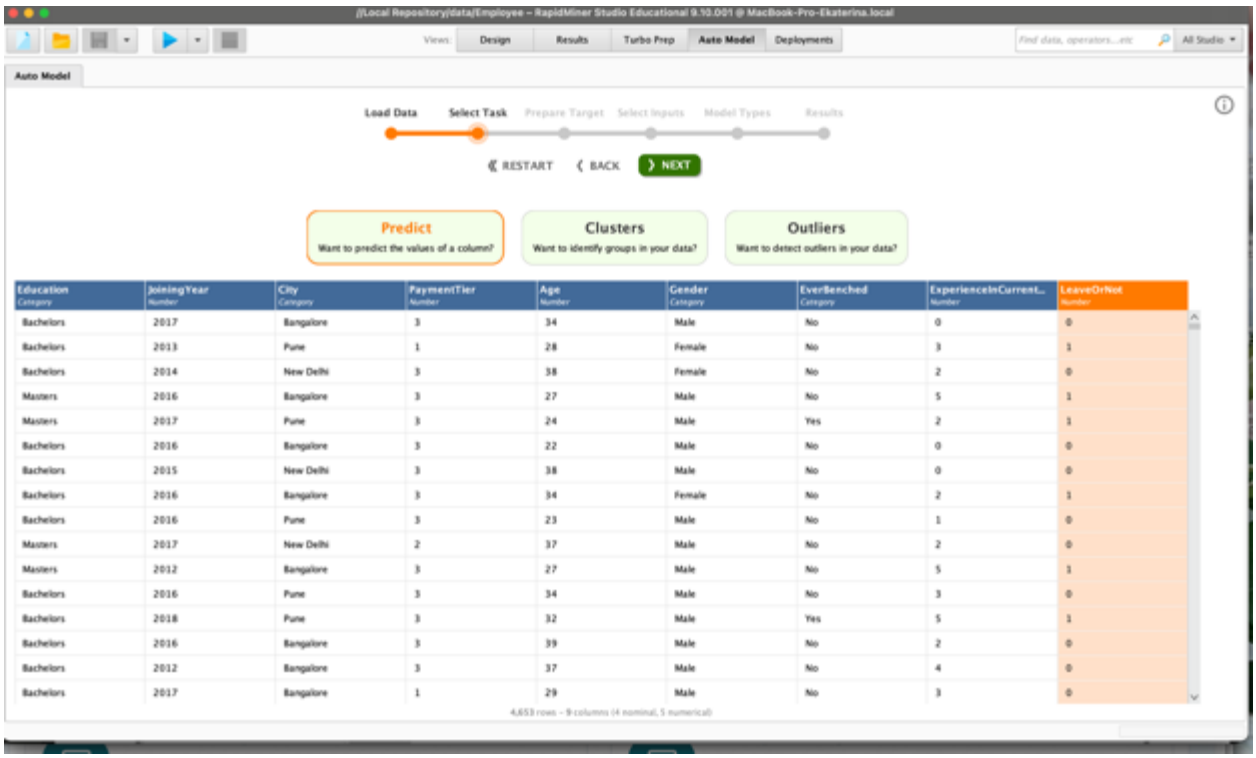
The dataset for training the models and evaluating them is imported into RapidMiner Studio. We use Turbo Prep and Auto Model technologies for data preparation and building the models.



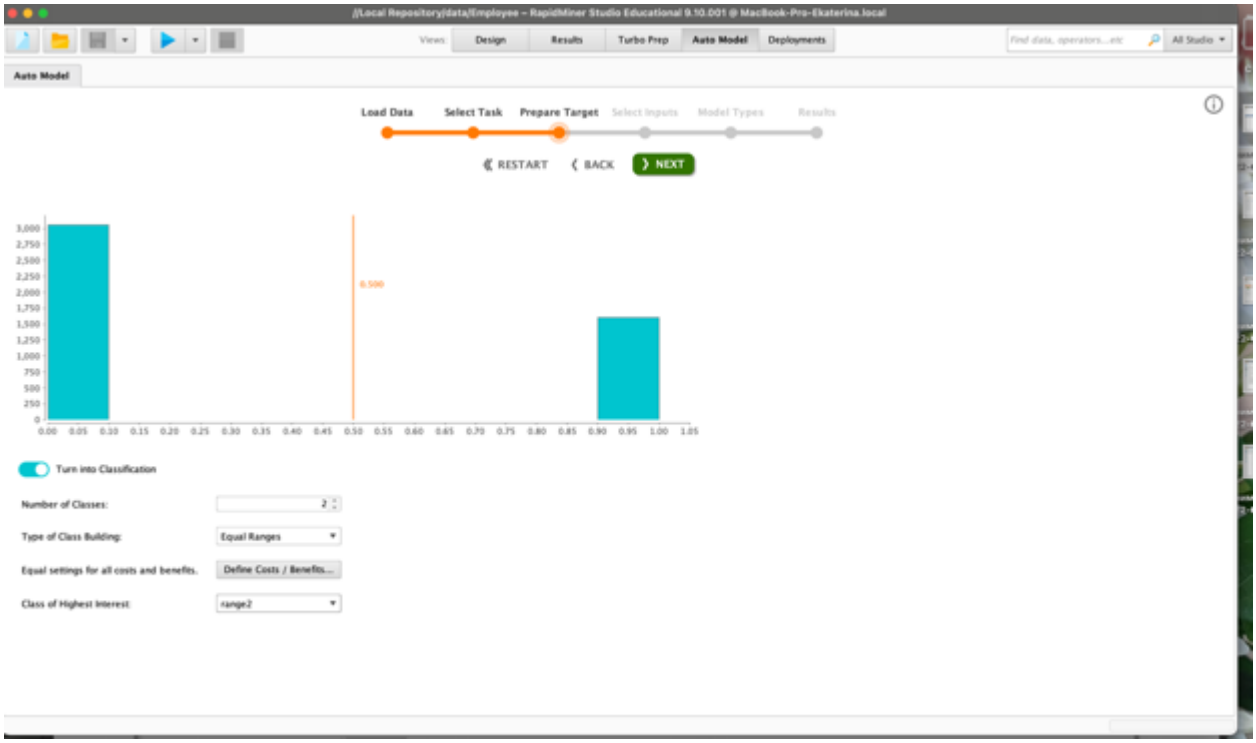




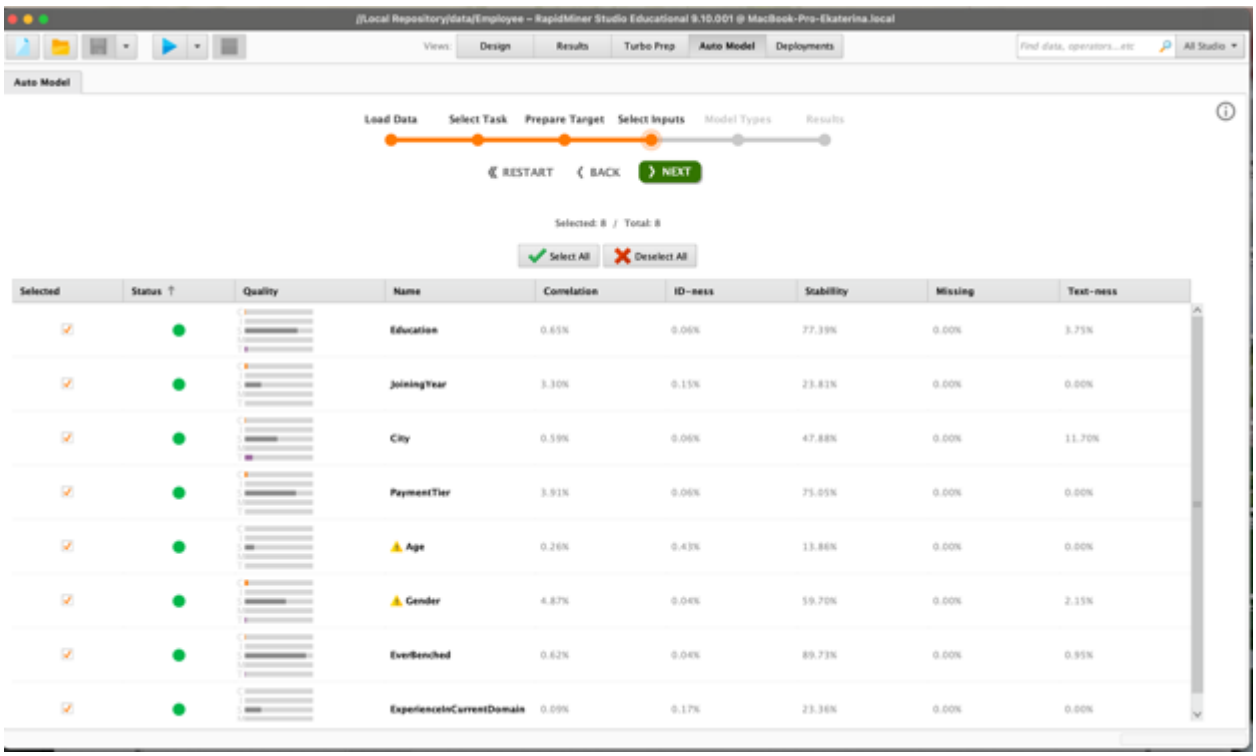
Using Auto Model, we tune the setting for model creation and select the output field.



We evaluate the dataset based on the target and see that we have a class imbalance.



We evaluate the input data and choose the fields that will be used to train the model. All fields in the provided dataset need to be included for the model creation.

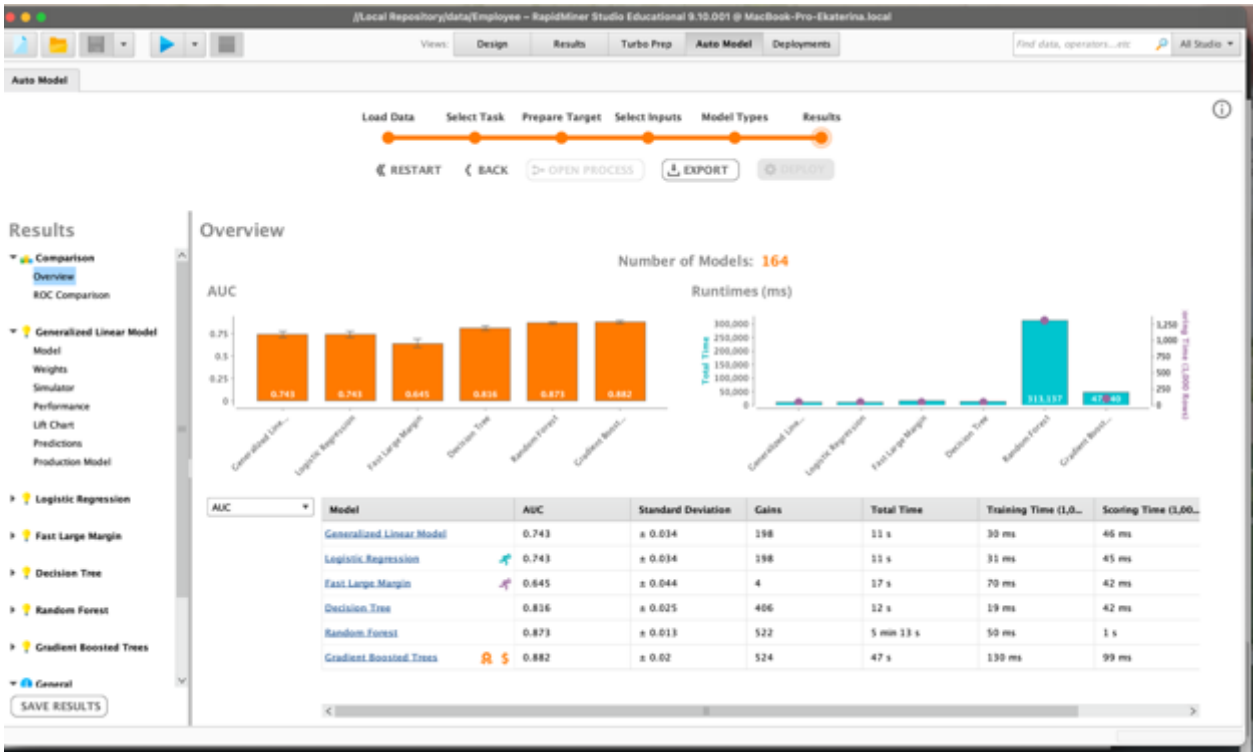


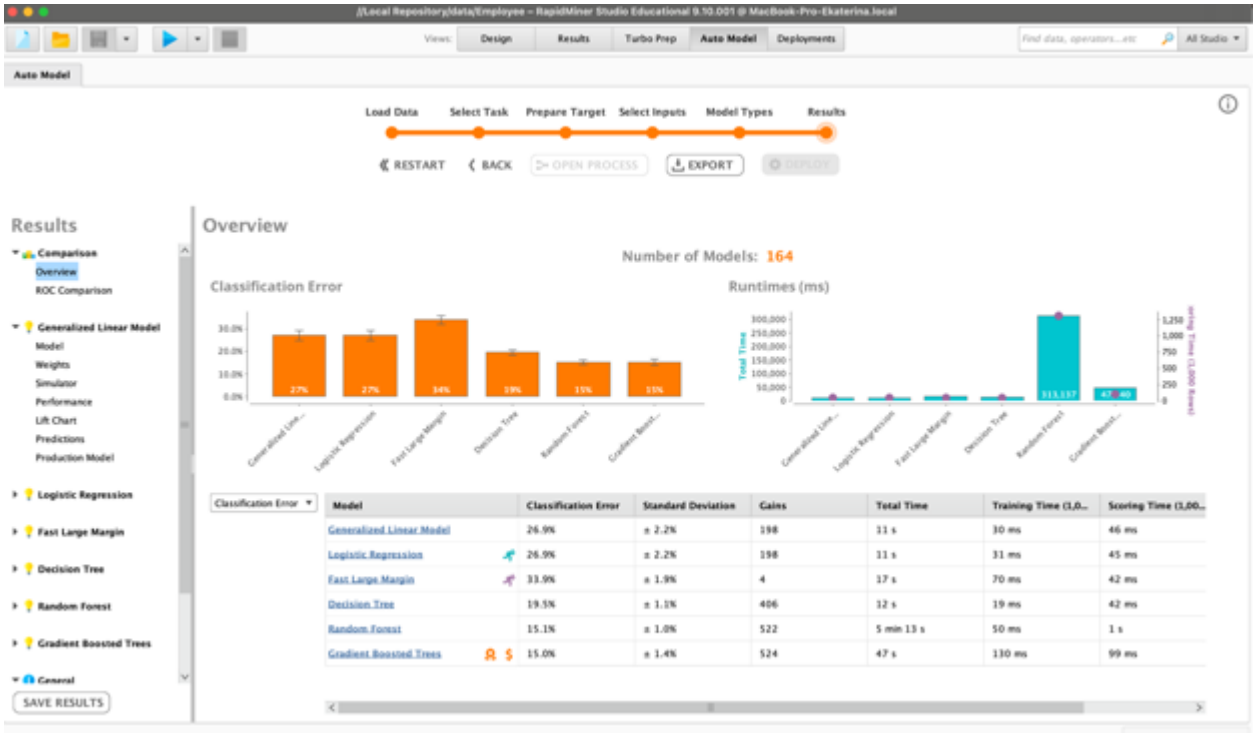
Choosing the models

We need to select what models will be trained, so we only keep those we want to use.

Evaluating the models

The next step is to evaluate the results of models we trained. We will select the best model based on **AUC** and **Classification error** parameters. The best model is **Gradient Boosted Trees.**

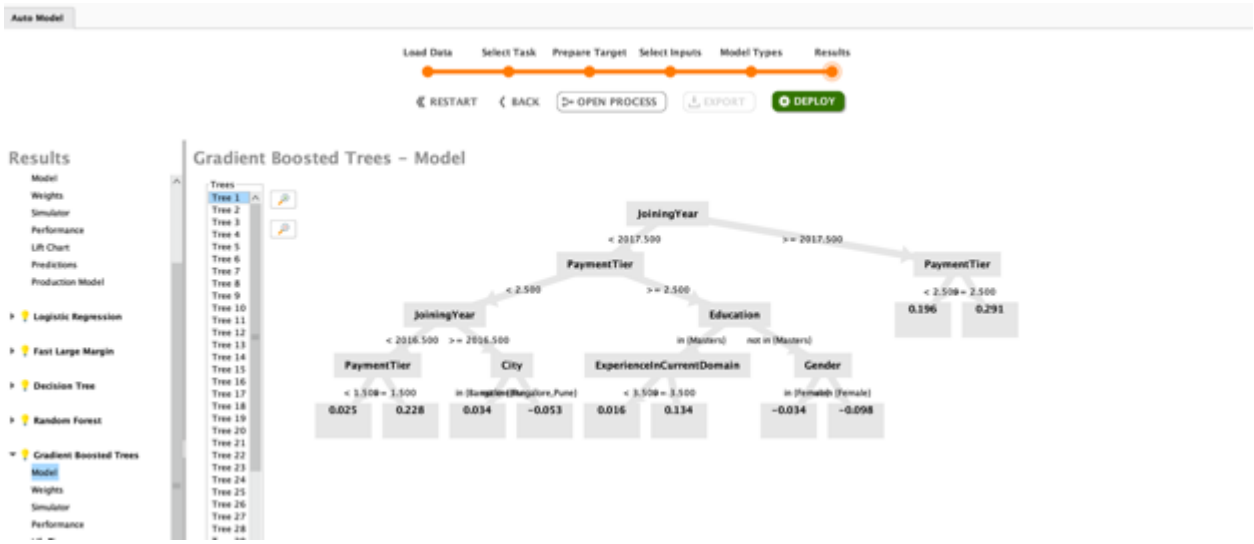




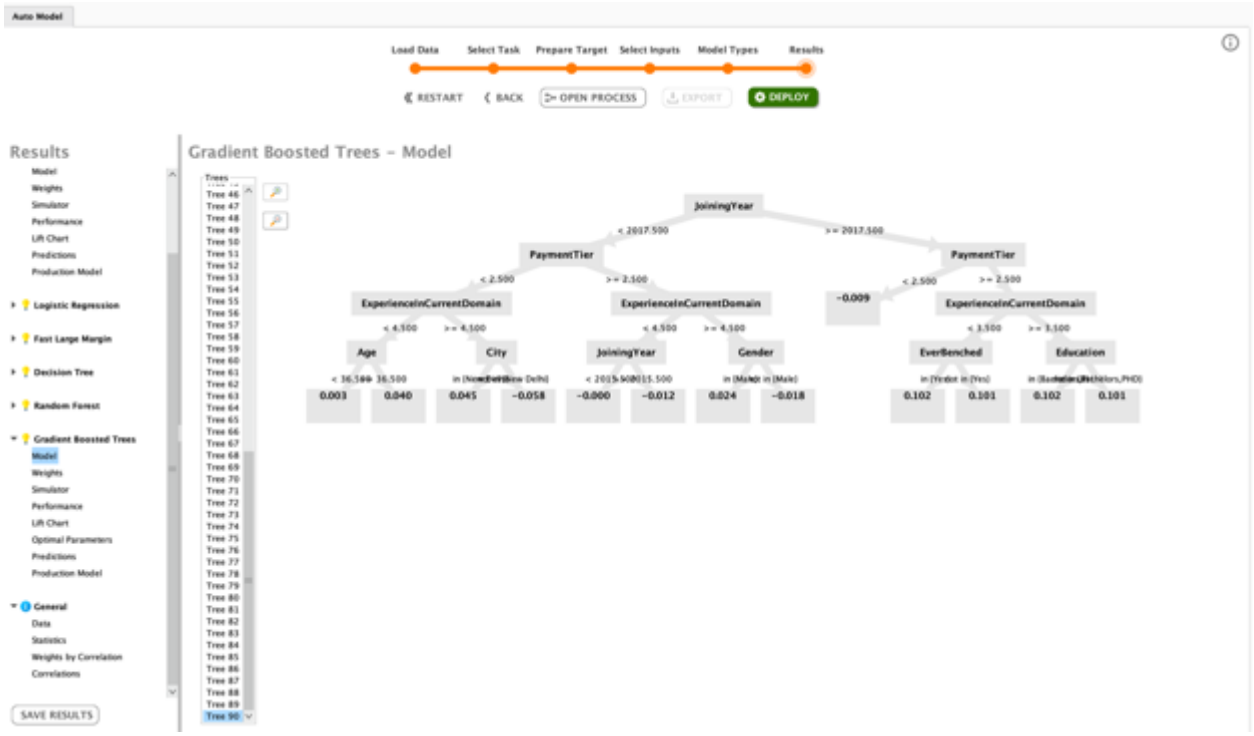
The best model in details

Let's have a closer look at the best model we received - Gradient Boosted Trees. The model has 90 Solution Trees.

The first Solution Tree has the following structure:

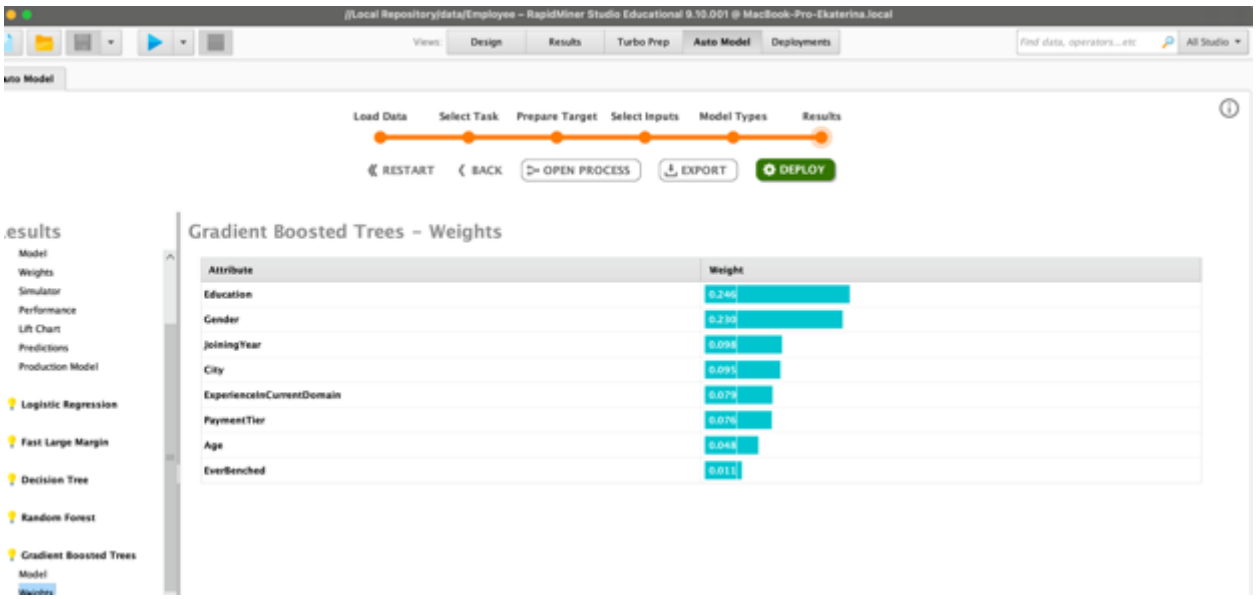


The best Solution Tree has the structure below:



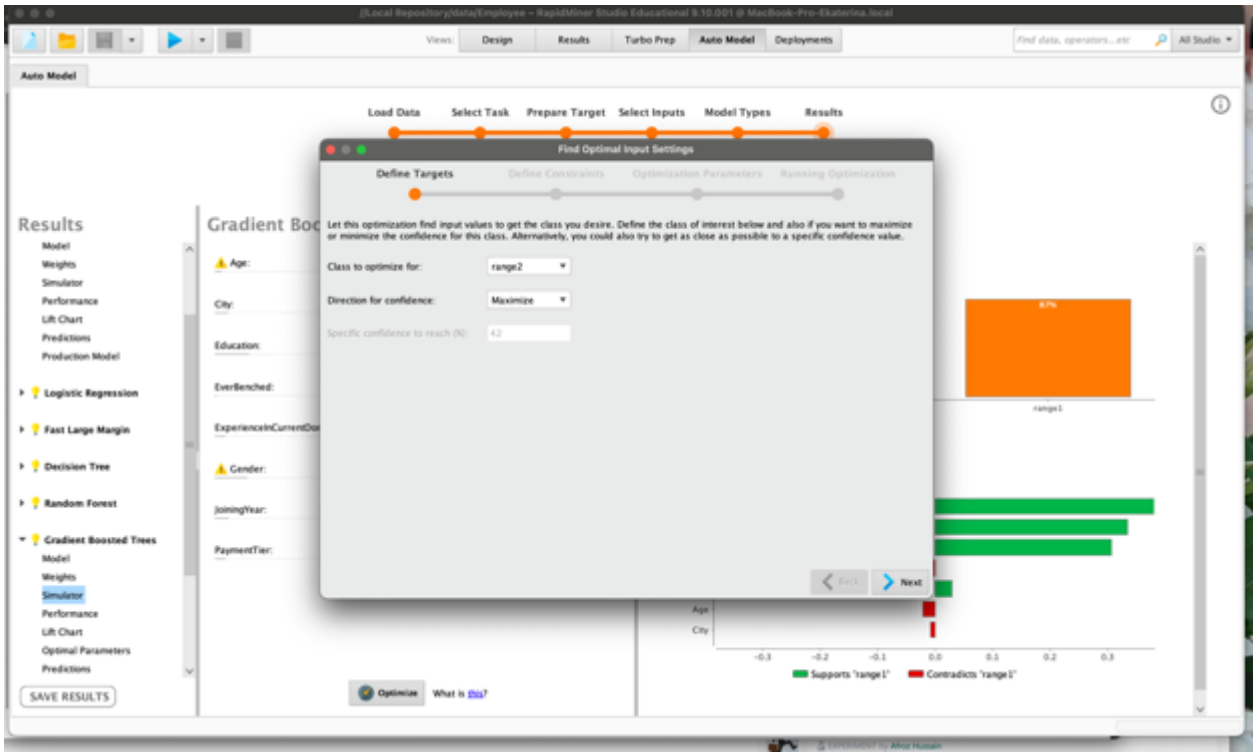
Most significant fields for the model

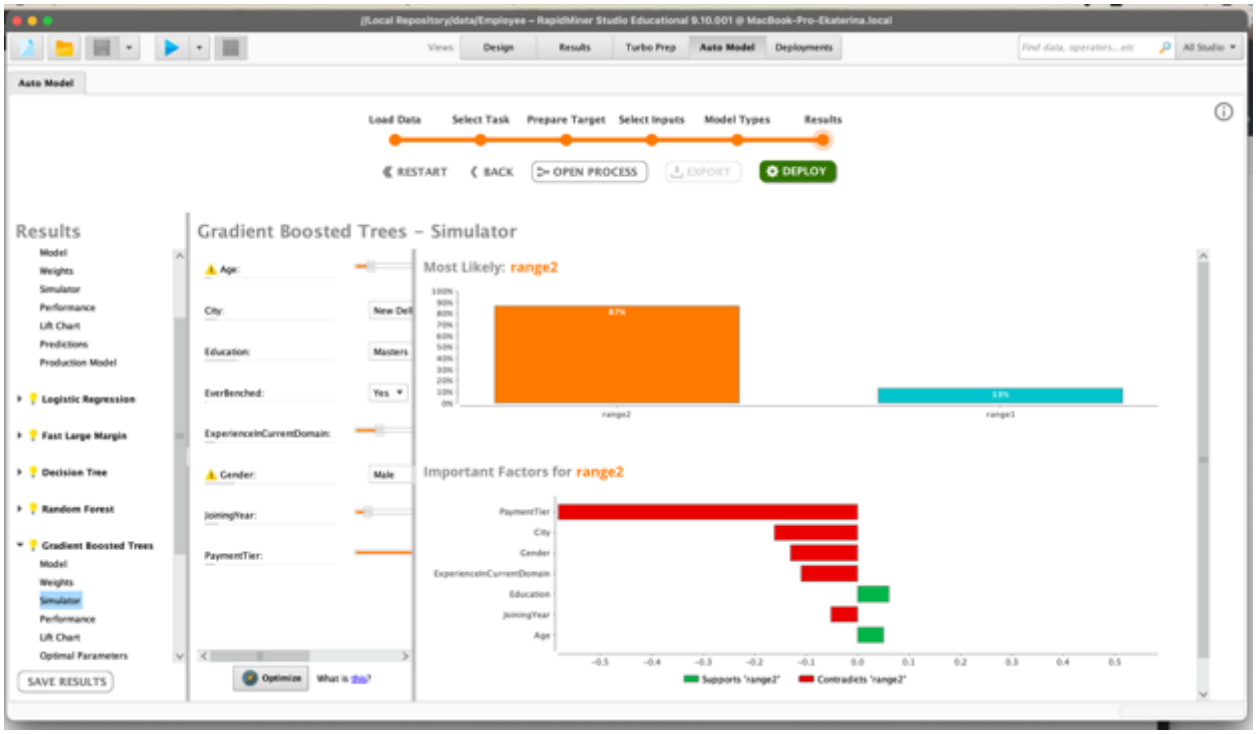
Let’s see what fields had the most significant impact on the solution when creating the model.

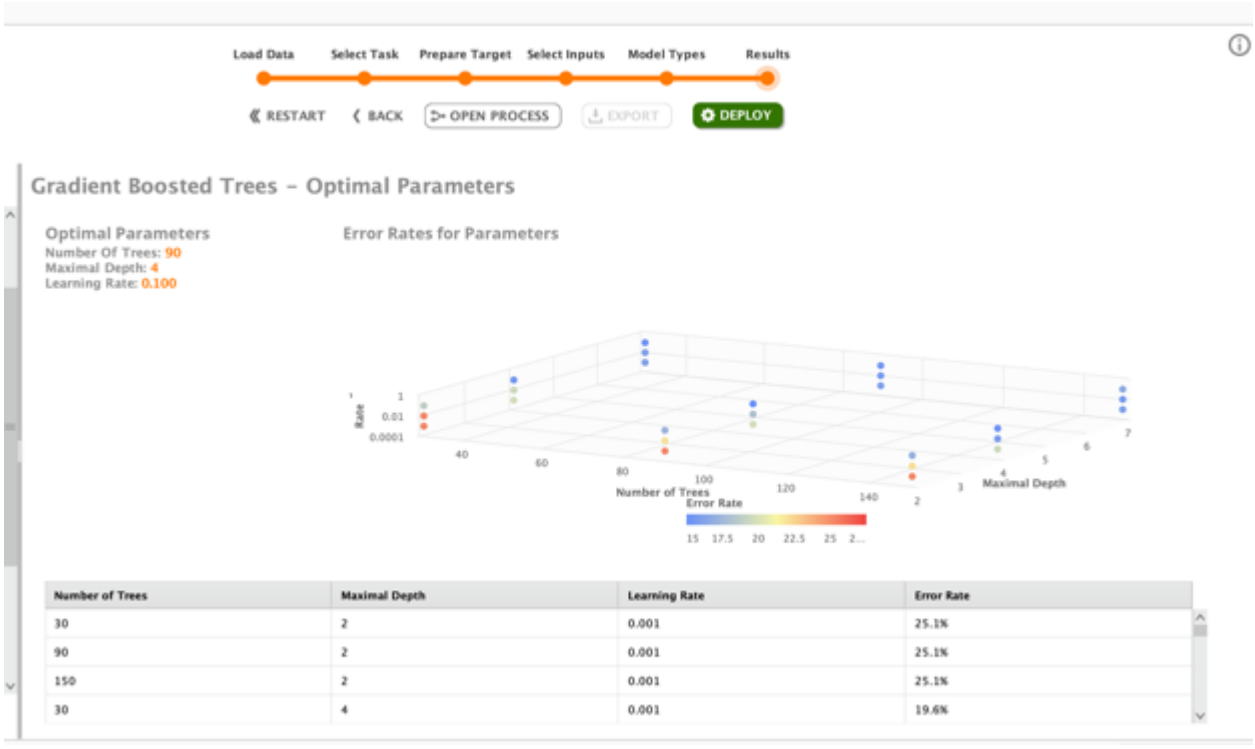


The highest weight in this model is shown for the following fields: **Education, Gender, JoiningYear**.
It means that **Education, Gender, JoiningYear** are the parameters that have the most impact on if the employee will leave the company or not.

Optimizing the model parameters

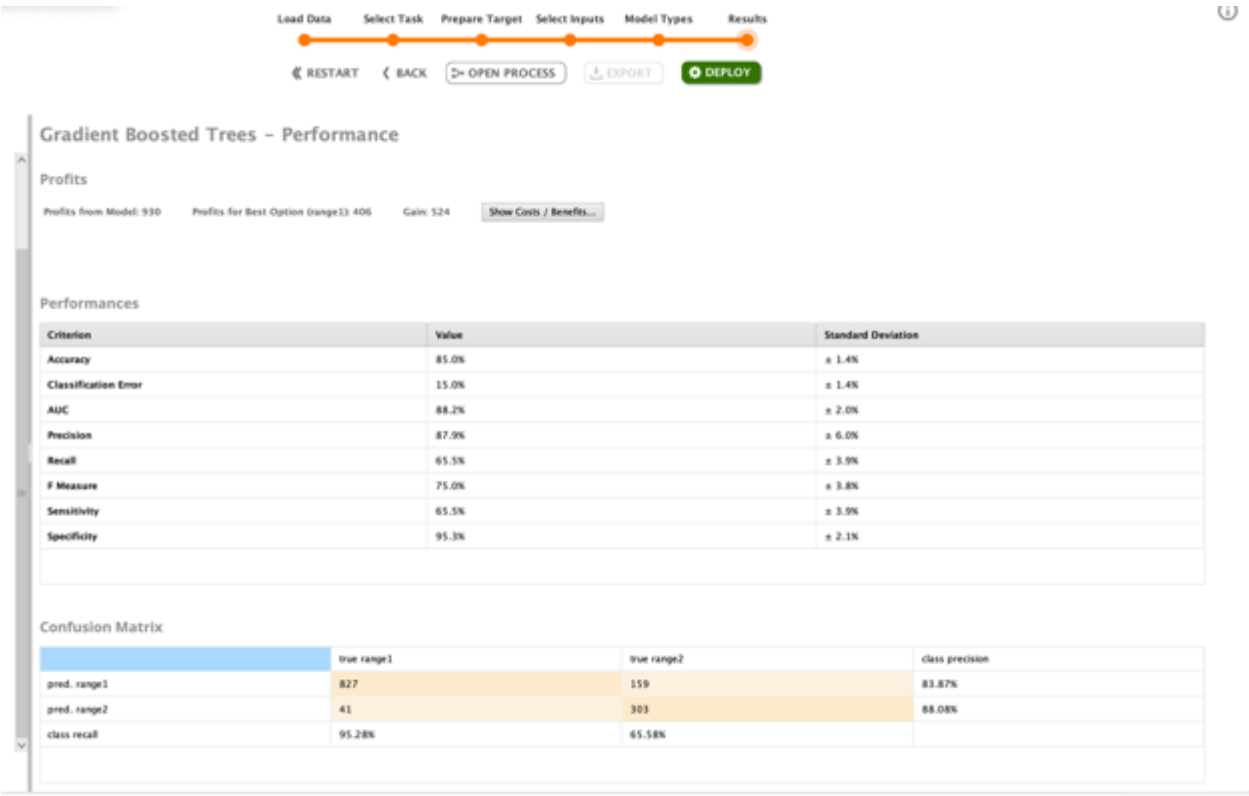


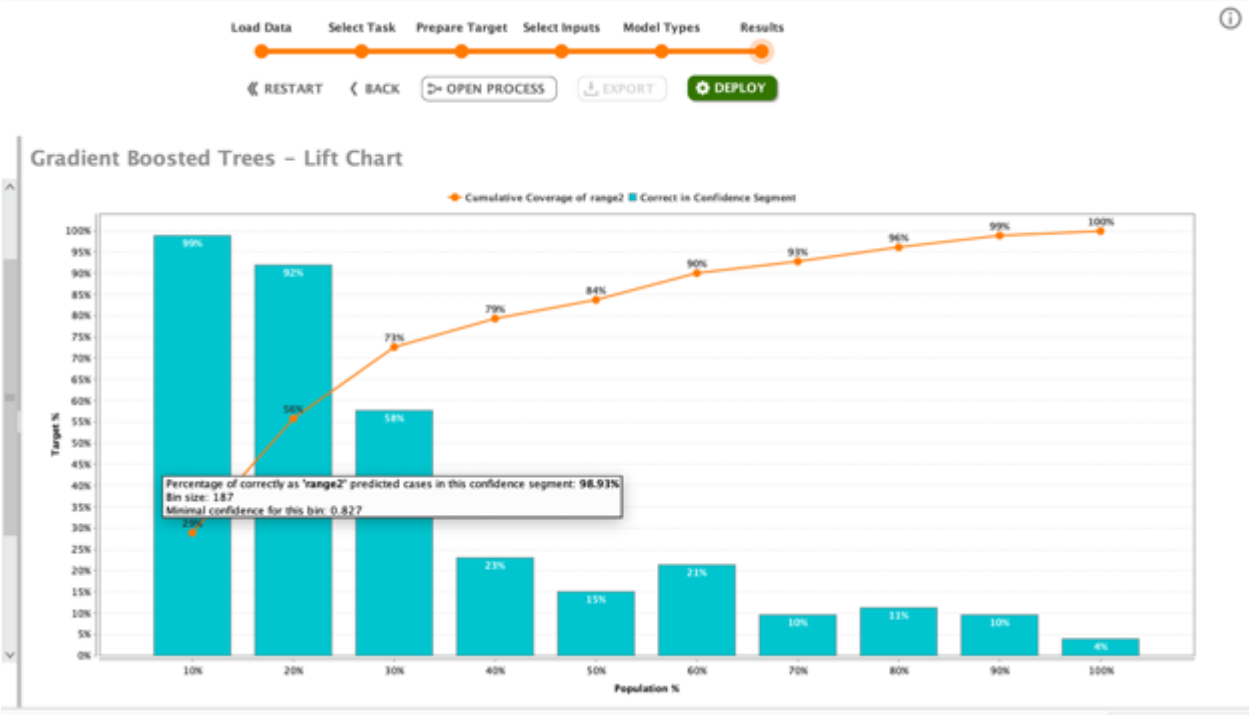




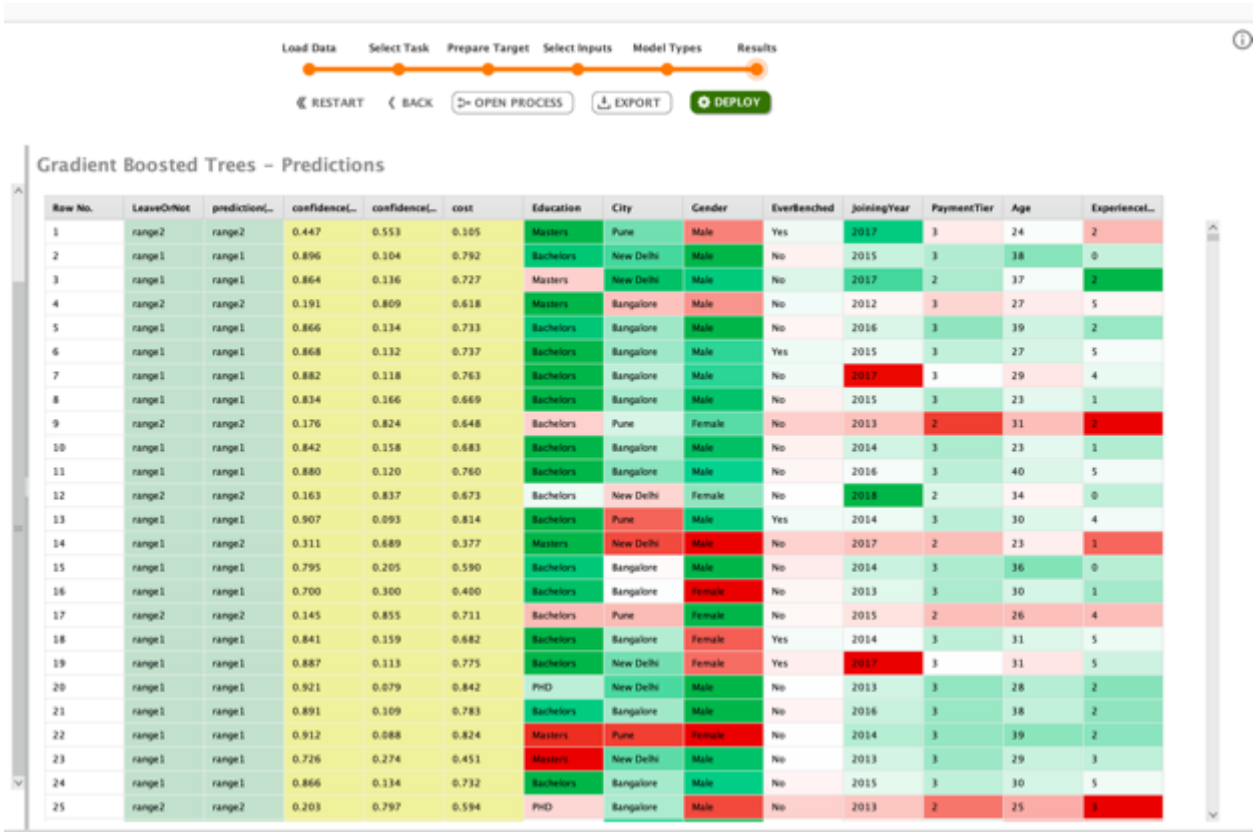
Number of Trees	Maximal Depth	Learning Rate	Error Rate
30	2	0.001	25.1%
90	2	0.001	25.1%
150	2	0.001	25.1%
30	4	0.001	19.6%

Let’s have a look at the rest of the metrics, they all are quite good.

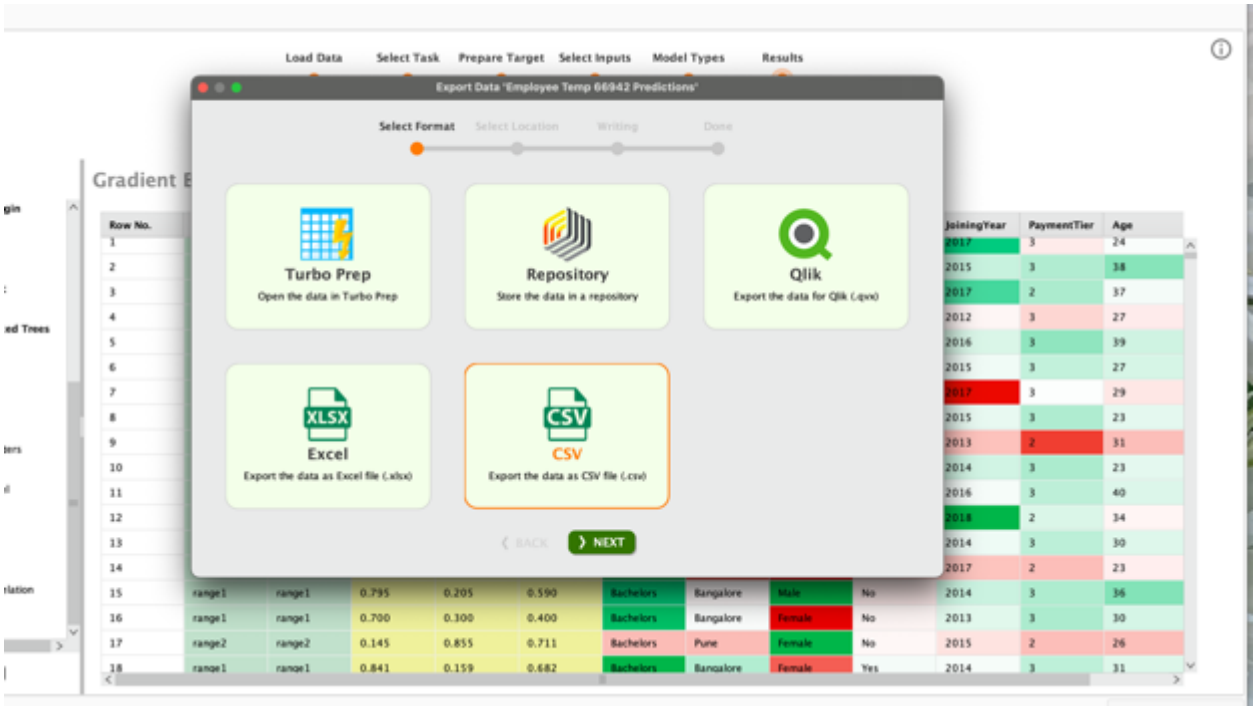




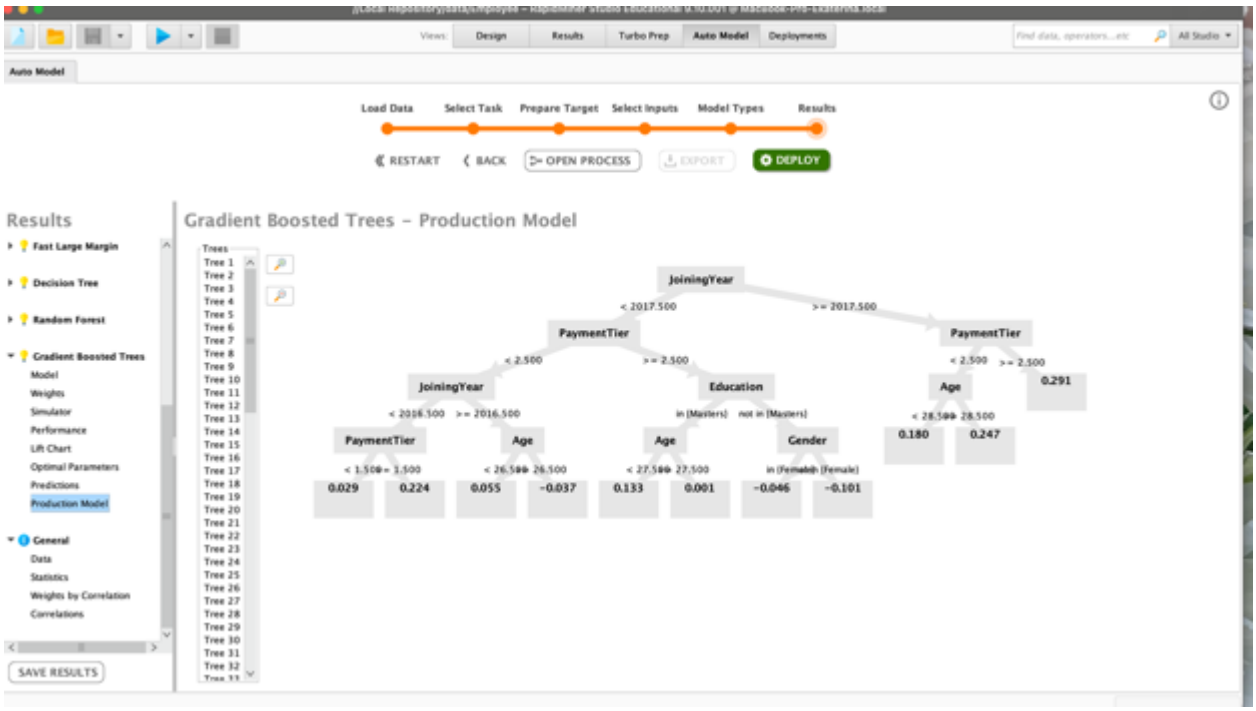
Now let’s compare the results of the prediction provided by the model and the results of the target field that we have in provided dataset.



We can export the results of prediction provided by the model if it is needed.

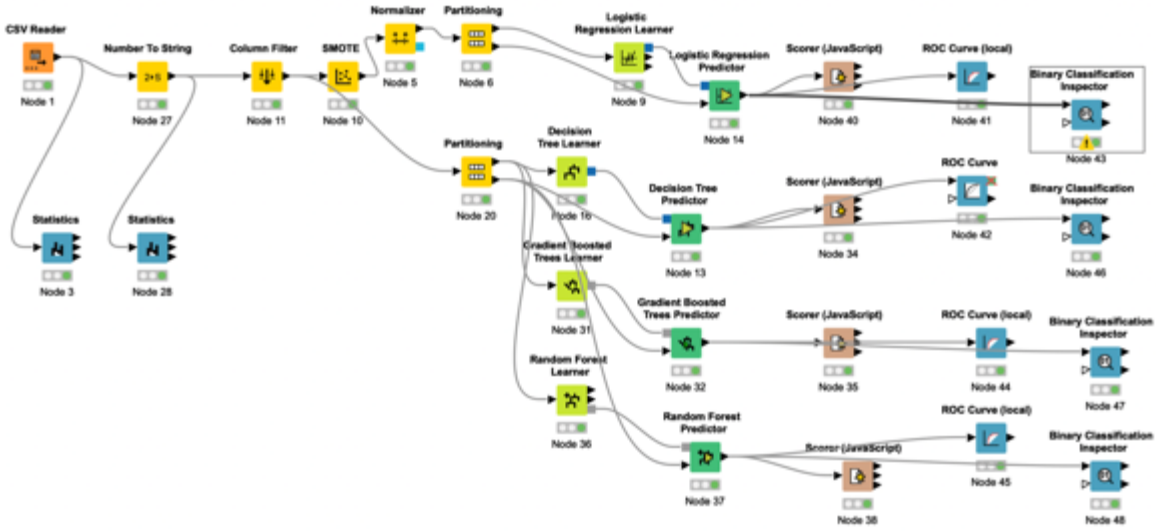


The structure of predictive model (Gradient Boosted Trees)



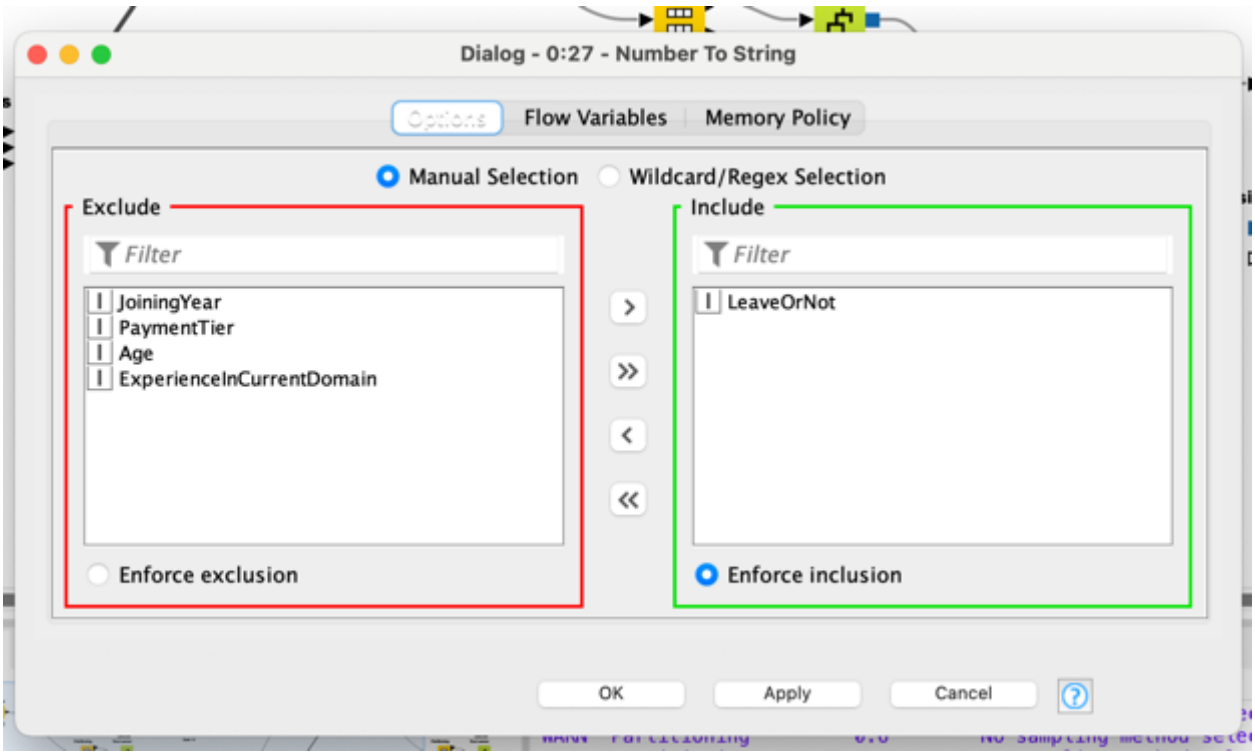
3. Knime Analytics Platform

Let's use **Knime Analytics Platform** and create and train 4 ML models: **Logistic Regression, Gradient Boosted Trees, Decision Tree, Random Forest**.

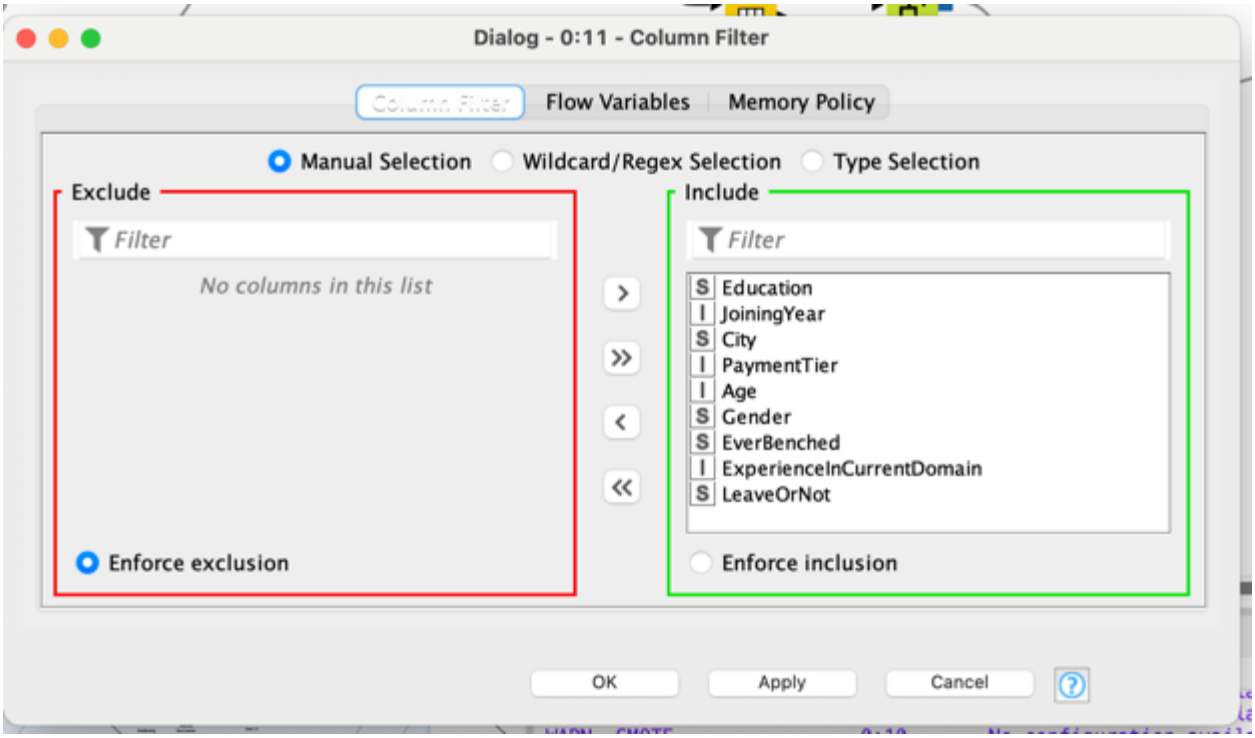


Converting the target filed to categorical type

The target field is numerical now, but in order to use it as a target field for classification task, we need to convert it to categorical type.



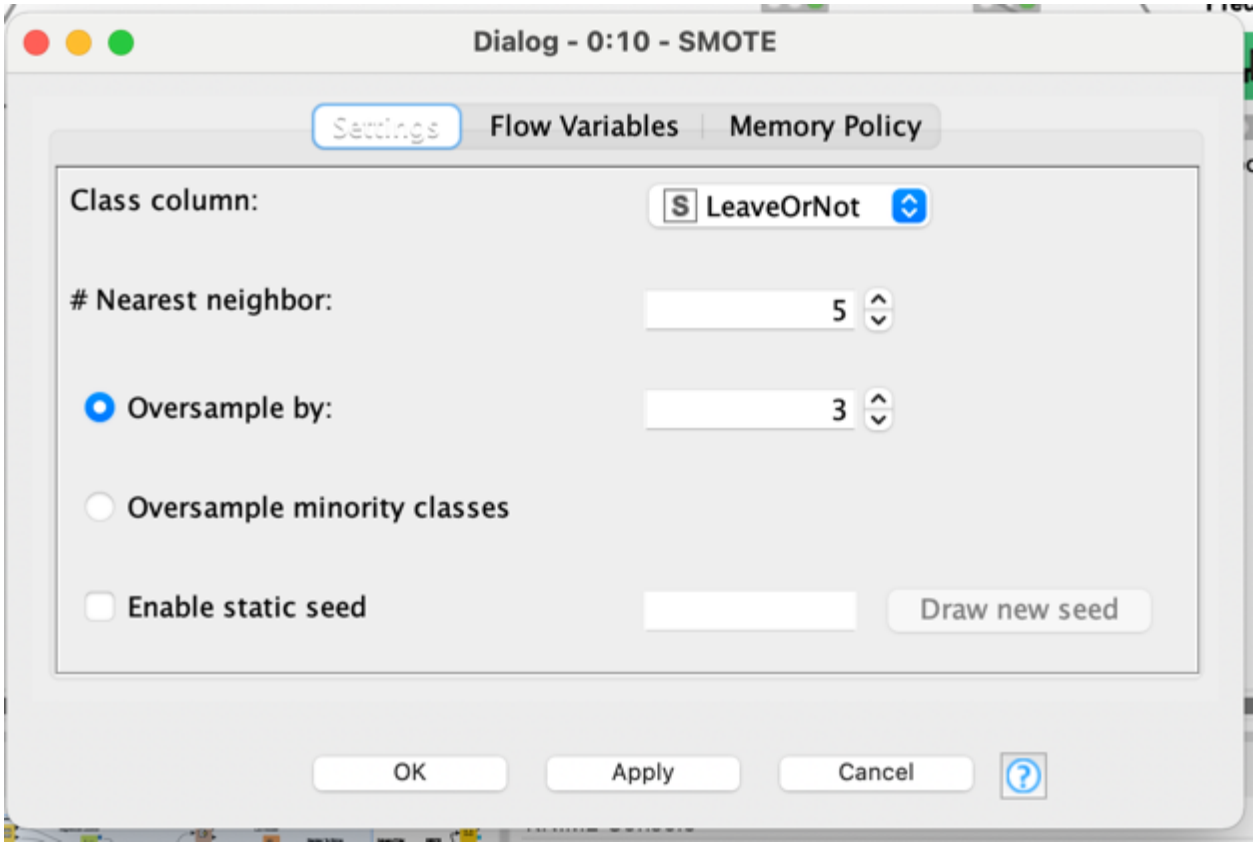
The dataset is clean, there are no outliners, missed values, all fields will be used for creating the model, no need to drop any fields.



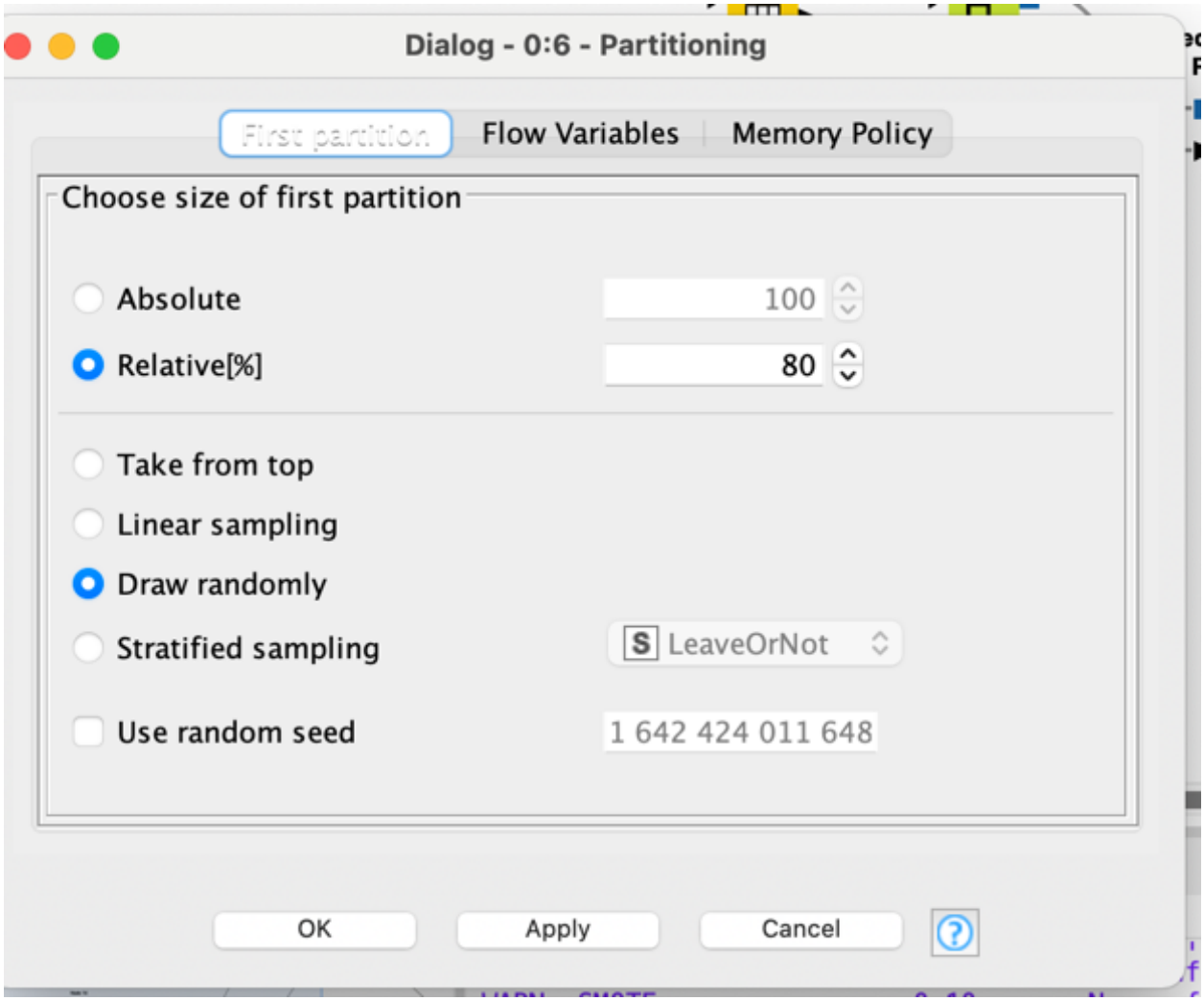
SMOTE module to fix the class imbalance

As we already know, there is a class imbalance in this dataset, that means that there is a severe skew in the class distribution and this class imbalance may affect the model that will be created. If we train the model on the imbalanced data, the predictions would be skewed as well.

To alleviate the imbalance, we will use SMOTE module to artificially increase the data to balance the dataset.

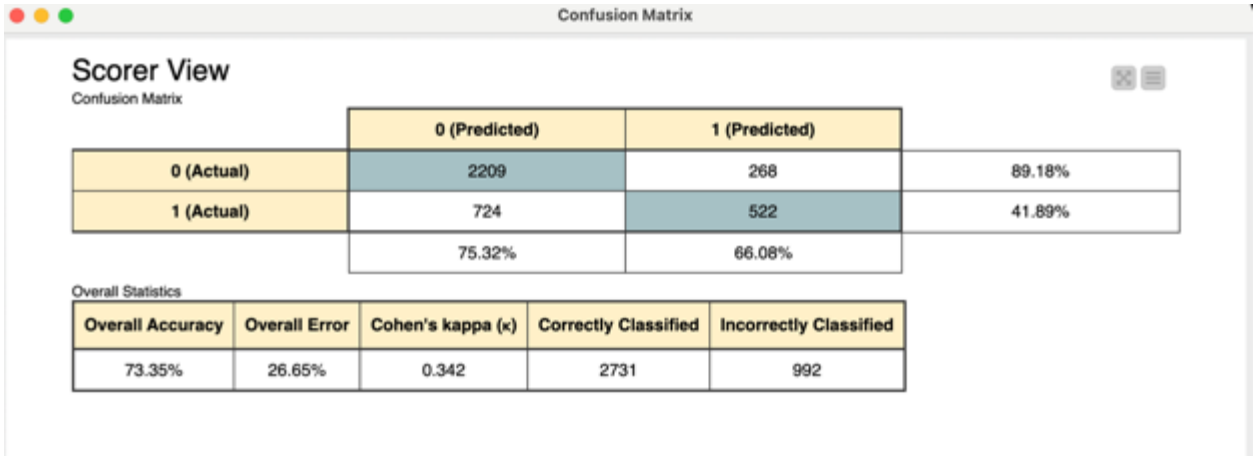


Let’s split the dataset into the training (80%) and test sets (20%).



Comparing the models

Logistic Regression



Confusion Matrix

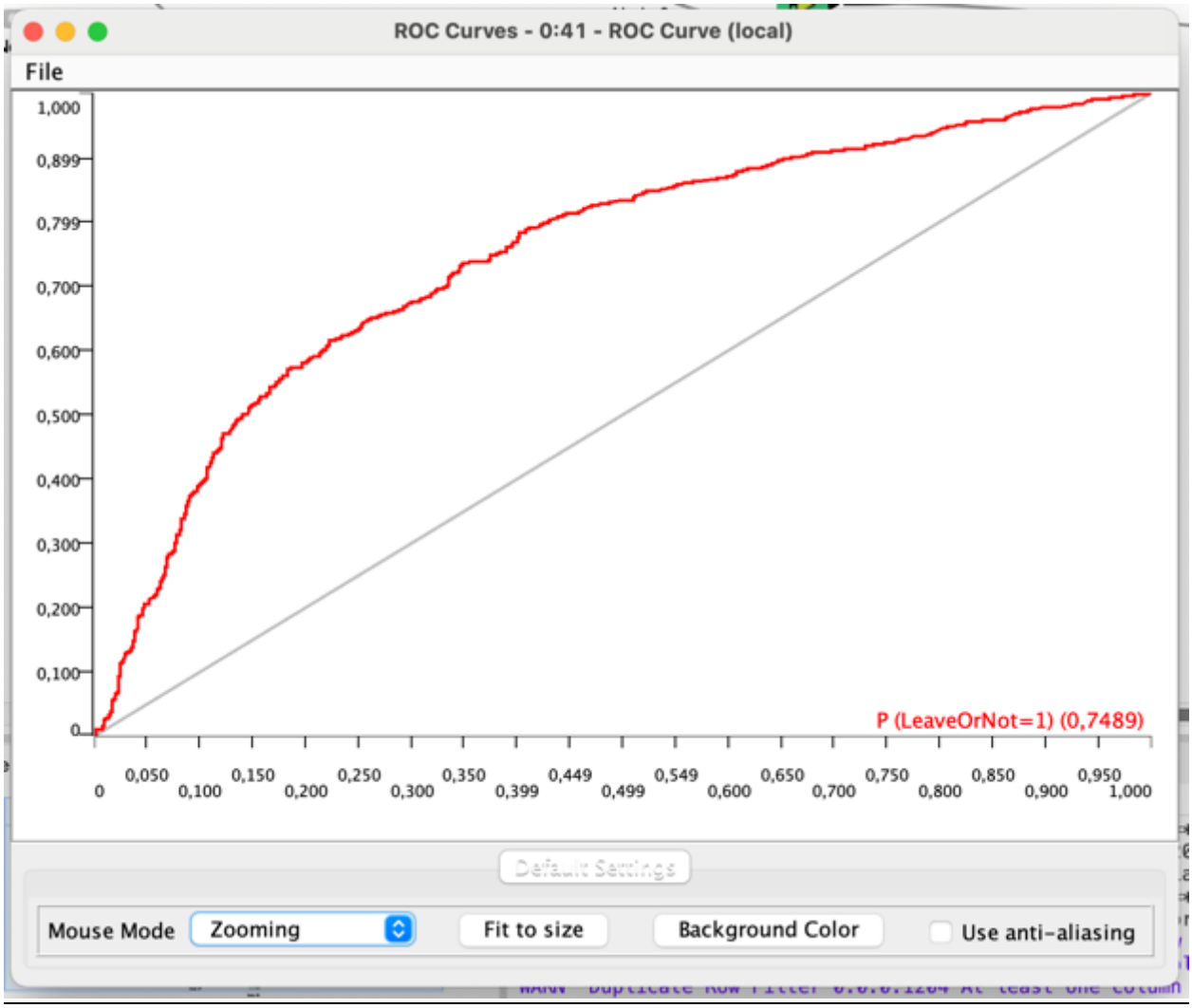
Scorer View

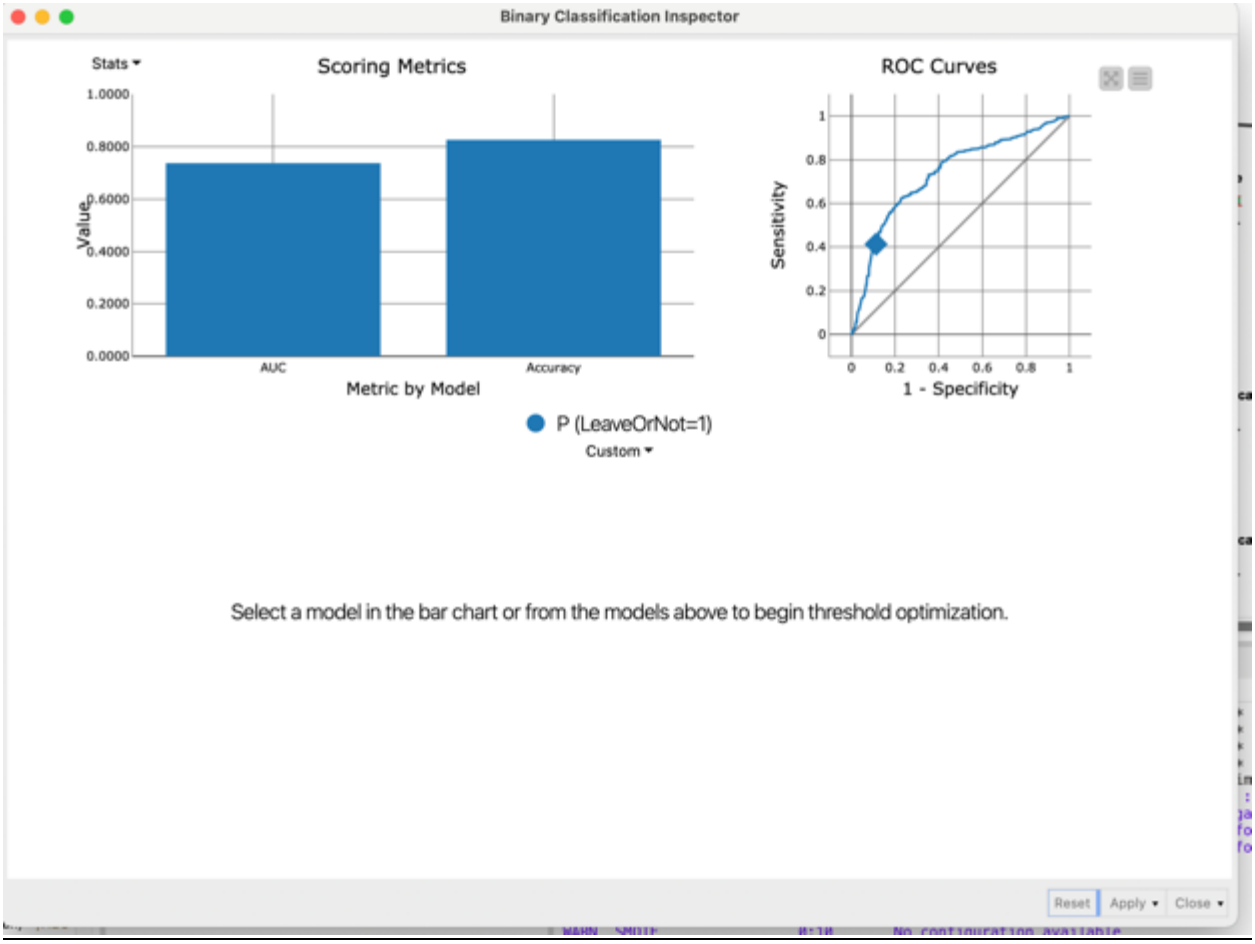
Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	2209	268	89.18%
1 (Actual)	724	522	41.89%
	75.32%	66.08%	

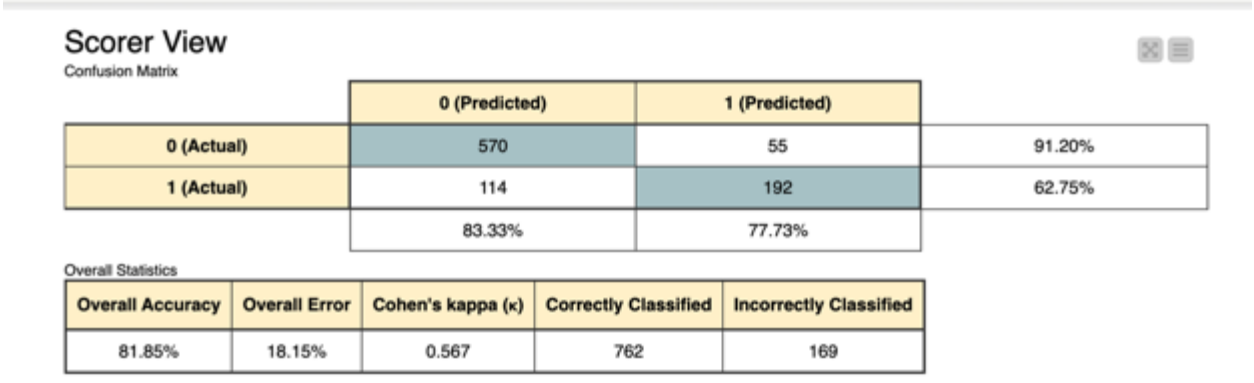
Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
73.35%	26.65%	0.342	2731	992


ROC Curves - 0:41 - ROC Curve (local)
File
1,000
0,899
0,799
0,700
0,600
0,500
0,400
0,300
0,200
0,100
0
P (LeaveOrNot=1) (0,7489)
0
0,050
0,100
0,150
0,200
0,250
0,300
0,350
0,399
0,449
0,499
0,549
0,600
0,650
0,700
0,750
0,800
0,850
0,900
0,950
1,000
Default Settings
Mouse Mode
Zooming
Fit to size
Background Color
Use anti-aliasing



Decision Tree



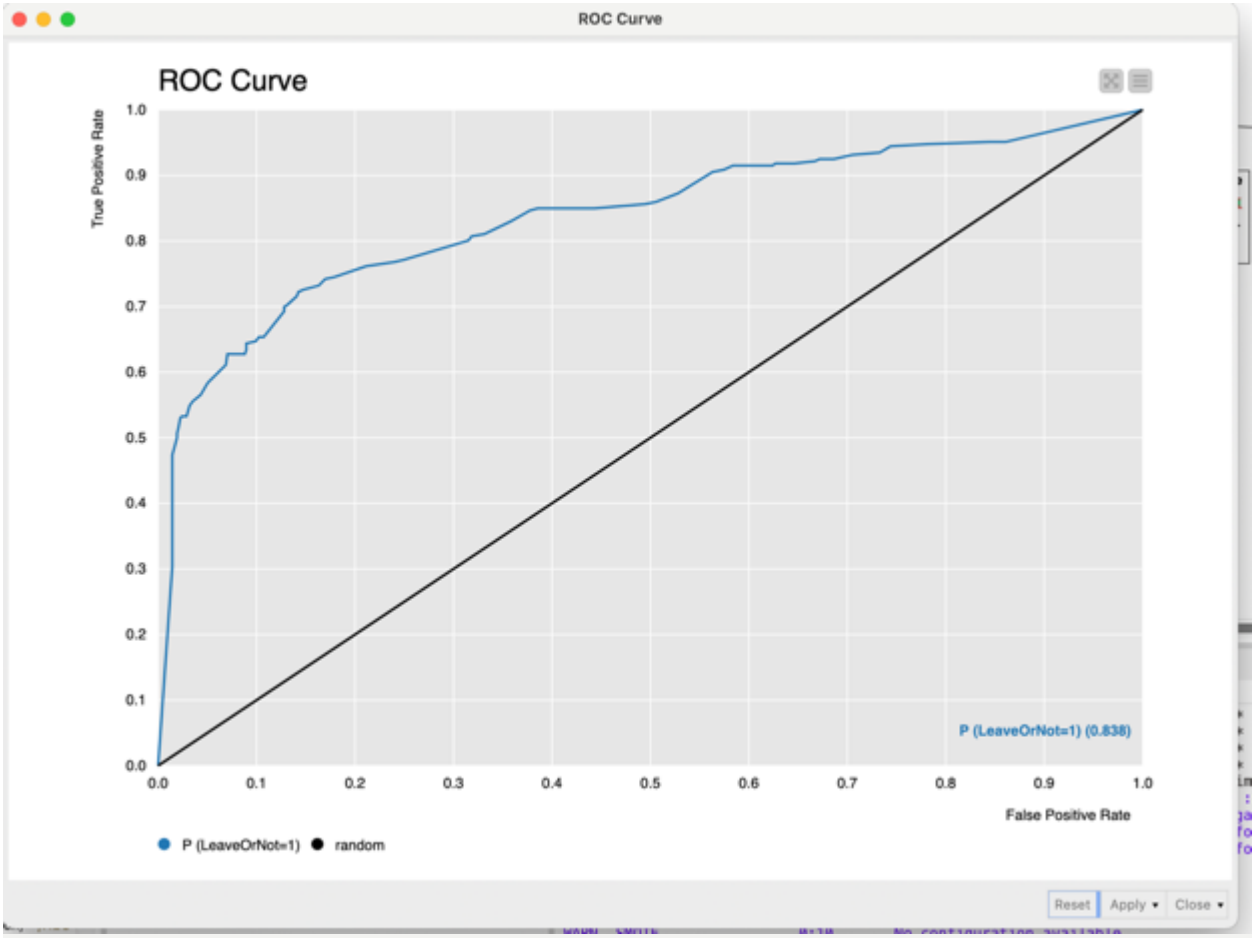
Scorer View

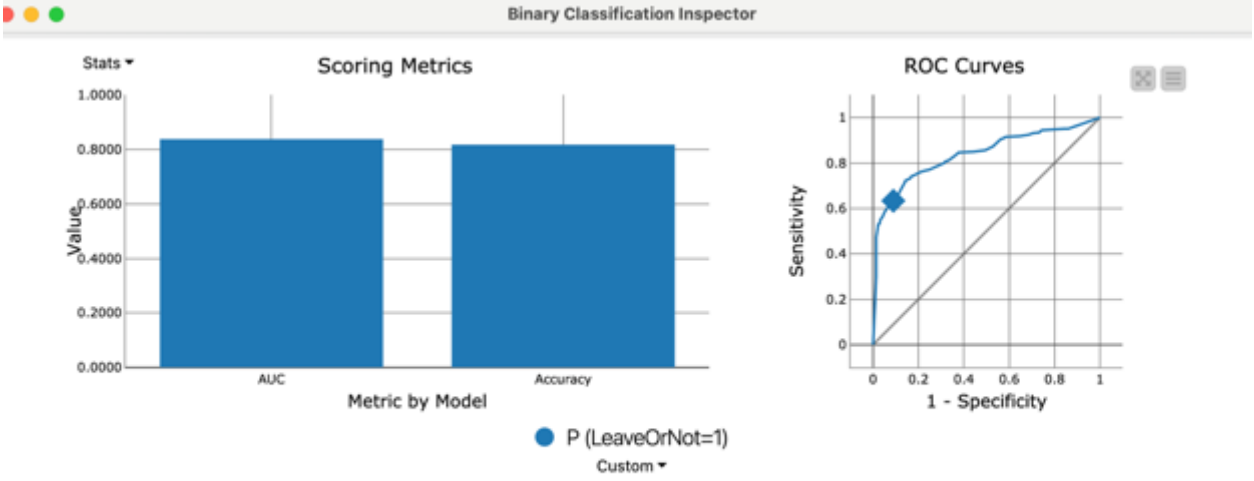
Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	570	55	91.20%
1 (Actual)	114	192	62.75%
	83.33%	77.73%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
81.85%	18.15%	0.567	762	169


ROC Curve
ROC Curve
True Positive Rate
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
False Positive Rate
P (LeaveOrNot=1) (0.838)
P (LeaveOrNot=1)
random
Reset
Apply
Close


Binary Classification Inspector
Stats
Scoring Metrics
1.0000
0.8000
0.6000
0.4000
0.2000
0.0000
Value
AUC
Accuracy
Metric by Model
ROC Curves
Sensitivity
1
0.8
0.6
0.4
0.2
0
0
0.2
0.4
0.6
0.8
1
1 - Specificity
P (LeaveOrNot=1)
Custom

Gradient Boosted Tree

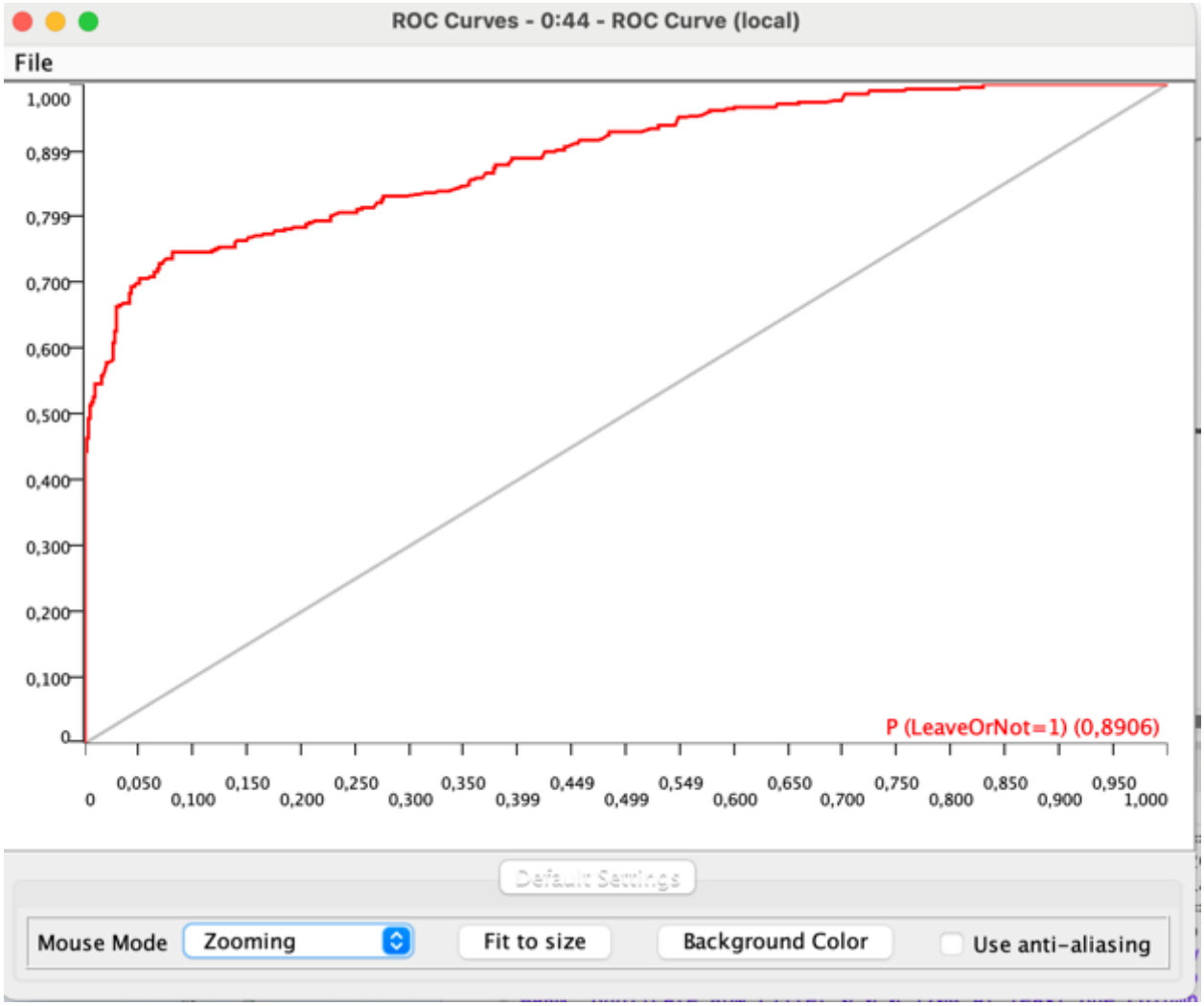
Scorer View

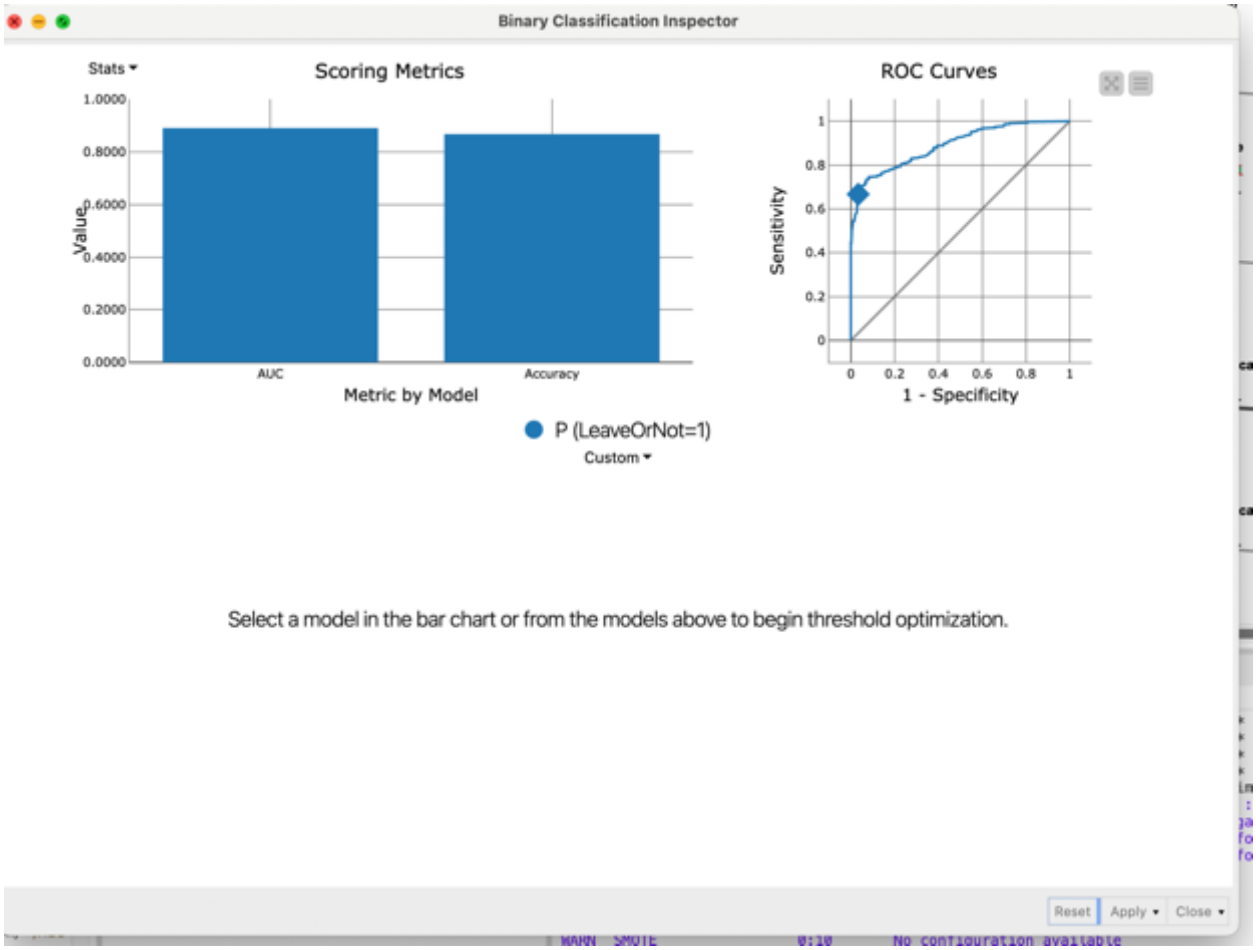
Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	605	20	96.80%
1 (Actual)	102	204	66.67%
	85.57%	91.07%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
86.90%	13.10%	0.681	809	122


ROC Curves - 0:44 - ROC Curve (local)
File
1,000
0,899
0,799
0,700
0,600
0,500
0,400
0,300
0,200
0,100
0
P (LeaveOrNot=1) (0,8906)
0
0,050
0,100
0,150
0,200
0,250
0,300
0,350
0,399
0,449
0,499
0,549
0,600
0,650
0,700
0,750
0,800
0,850
0,900
0,950
1,000
Default Settings
Mouse Mode
Zooming
Fit to size
Background Color
Use anti-aliasing



Random Forest

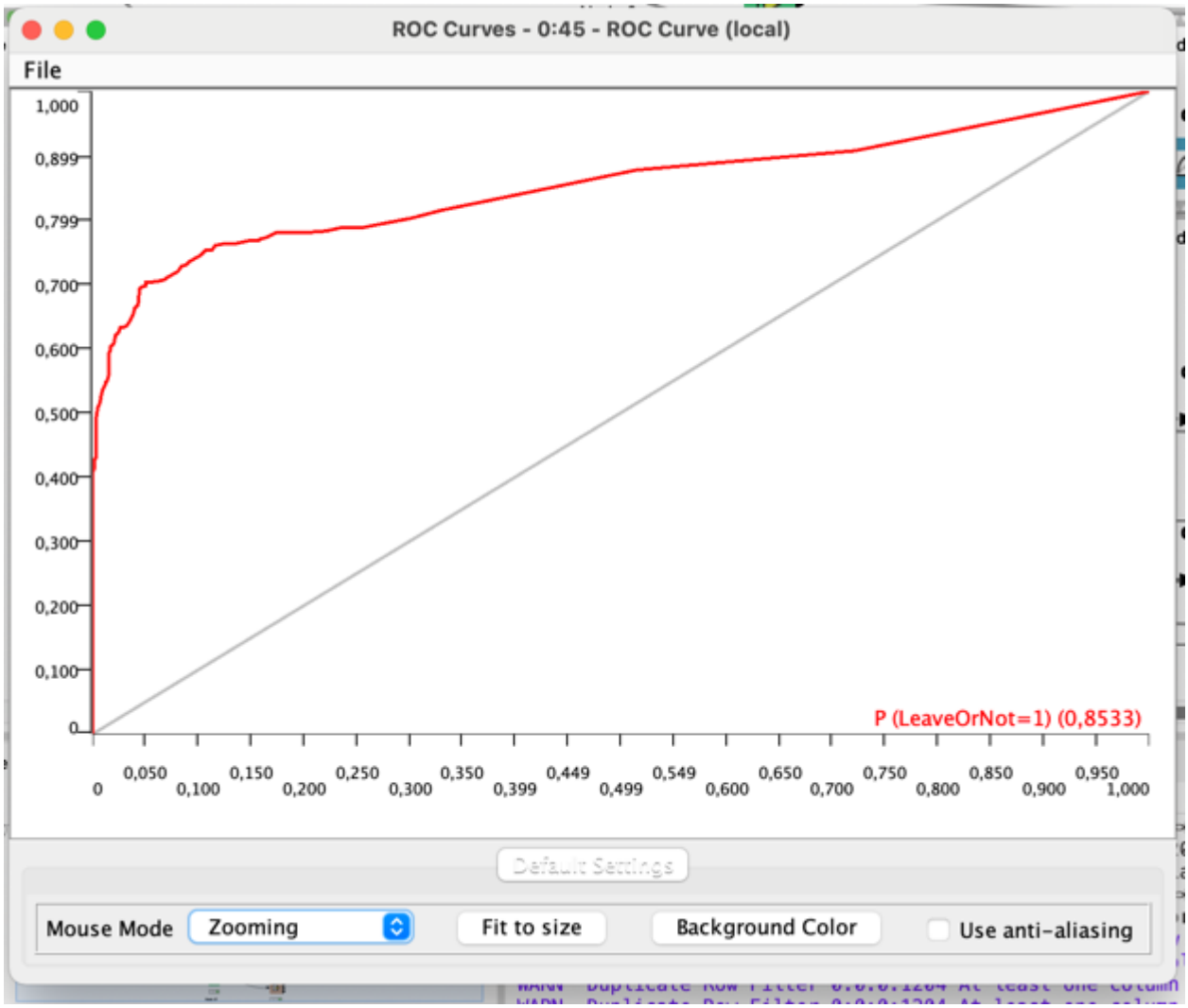
Scorer View

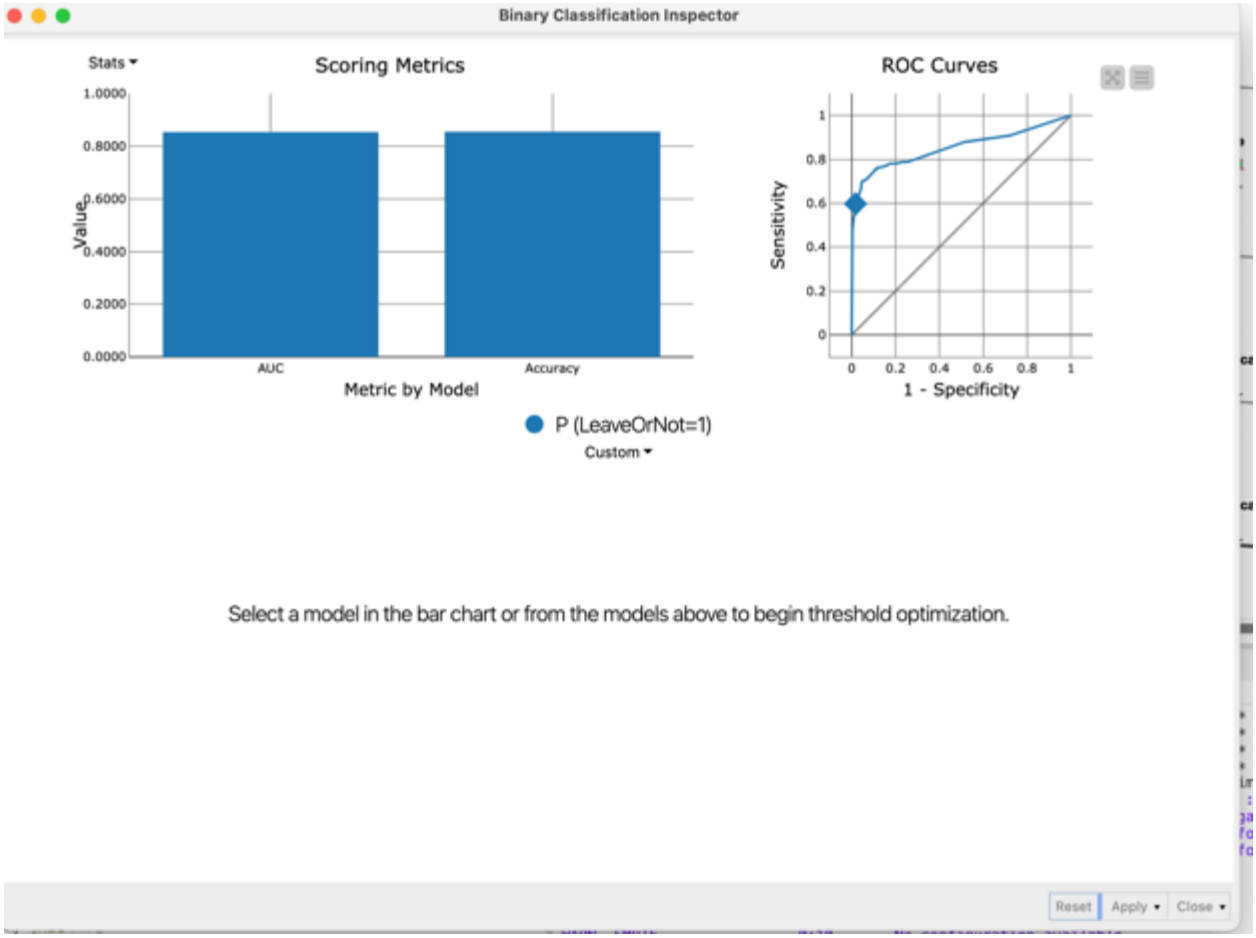
Confusion Matrix

	0 (Predicted)	1 (Predicted)	
0 (Actual)	615	10	98.40%
1 (Actual)	124	182	59.48%
	83.22%	94.79%	

Overall Statistics

Overall Accuracy	Overall Error	Cohen's kappa (κ)	Correctly Classified	Incorrectly Classified
85.61%	14.39%	0.640	797	134


ROC Curves - 0:45 - ROC Curve (local)
File
1,000
0,899
0,799
0,700
0,600
0,500
0,400
0,300
0,200
0,100
0
P (LeaveOrNot=1) (0,8533)
0
0,050
0,100
0,150
0,200
0,250
0,300
0,350
0,399
0,449
0,499
0,549
0,600
0,650
0,700
0,750
0,800
0,850
0,900
0,950
1,000
Default Settings
Mouse Mode
Zooming
Fit to size
Background Color
Use anti-aliasing



The best model

The best model is **Gradient Boosted Tree** with **AUC of 0.8906**.