PREDICTING EMPLOYEE TURNOVER WITH MACHINE LEARNING

Contents

Note: All figures and tables are placed in the Appendix with working links throughout the document. Links in the appendix redirect back to the report.

Introduction

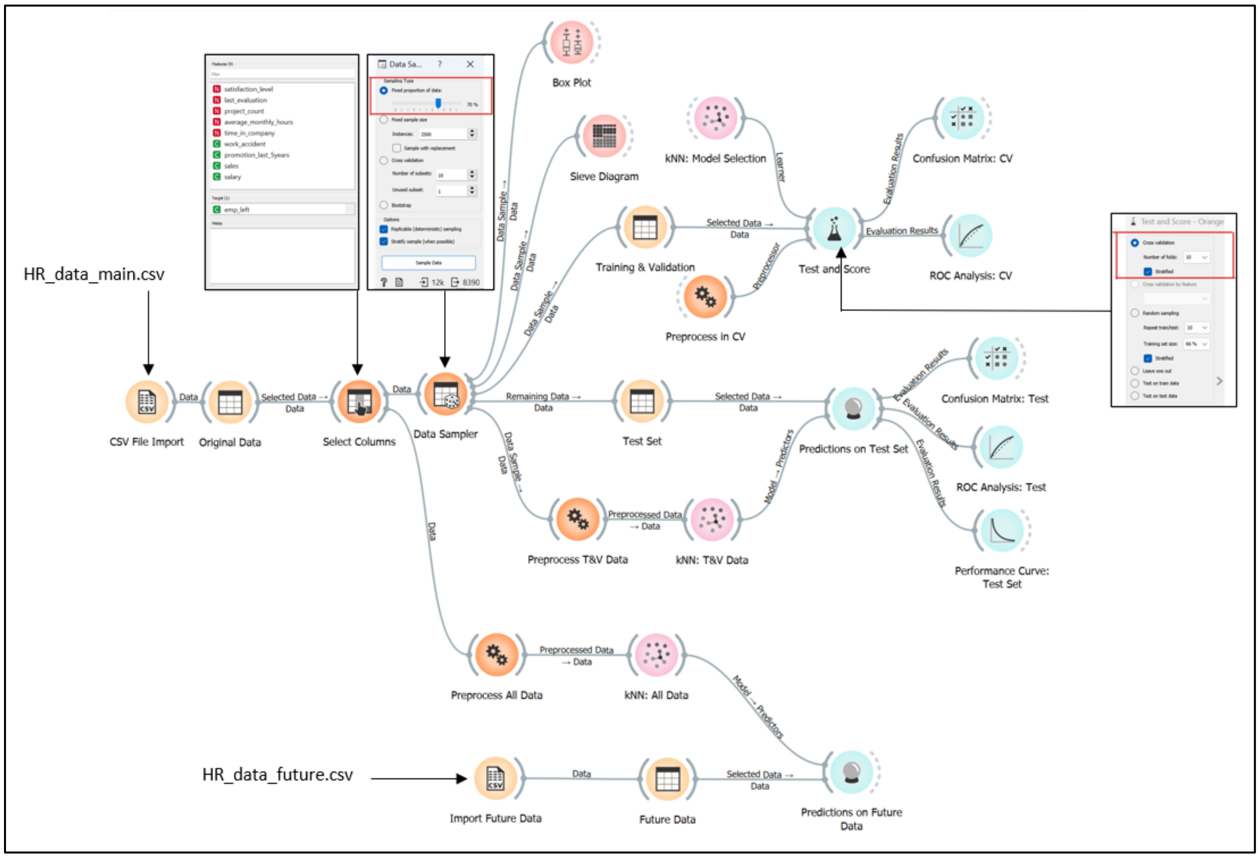
Employee turnover is a major challenge that organizations face, globally, as high turnover rates can result in increased costs, loss of institutional knowledge, and reduced morale among remaining employees. Predicting employee turnover can help organizations take proactive measures to retain employees and minimize the negative impact of employee turnover. Employee turnover refers to the rate at which employees leave a company and are replaced by new hires. In the United States, employee turnover has become a major managerial problem for businesses of all sizes and industries. High turnover rates can be costly and disruptive, affecting productivity, morale, and business performance.

Background information reveals that employee turnover rates in the US have been on the rise in recent years. According to Gartner, Inc. (2022) U.S. employee annual voluntary turnover is likely to jump 20% this year, from a pre-pandemic annual average of 31.9 million employees (about twice the population of New York) quitting their jobs to 37.4 million quitting in 2022. It is estimated that United States businesses spend one trillion dollars yearly on the costs of employee turnover (Zippa, 2023). There are numerous factors that contribute to employee turnover, including inadequate compensation and benefits, lack of opportunities for advancement, poor management and leadership, and job dissatisfaction. As such, companies must take a proactive approach to address these issues and develop effective strategies for retaining employees.

The significance of this project lies in the fact that high employee turnover can be extremely costly for businesses, as it leads to a loss of institutional knowledge, decreased productivity, and increased recruitment and training costs. Moreover, turnover can impact employee morale and job satisfaction, which can lead to a negative company culture and make it difficult to attract and retain top talent. Therefore, reducing employee turnover rates is critical to the long-term success of any business, and requires a strategic approach to talent management and employee engagement. In this project, we aim to predict employee turnover using a dataset available on Kaggle. We try out a variety of machine learning models and choose the best AUC to make predictions, form a profit curve, and ultimately inform business decisions. Our analysis helps business managers identify workers at high risk of quitting and recommends a practical strategy for preventing this and maximizing profit.

Data Collection and Preprocessing

Our dataset, containing 14,999 observations and 8 features, was fetched from Kaggle. We randomly separated 20% for future predictions and kept 80% for training the Random Forest model (which was determined to be the best model after testing several options). This was all done in Excel using the =RAND( ) function, which created a distribution of observations that followed the normal distribution. Observations assigned a random value ≥ 0.8 constituted the holdout ("future data”). Ultimately, we ended up with two separate Excel files “HR_data_main.csv” containing 11,985 observations for training, and “HR_data_future.csv” containing 3,014 observations for making “future” predictions on data the model hasn’t seen before. As per the following example, we uploaded “HR_data_main.csv” to the “CSV File Import” widget and “HR_data_future.csv” to the “Import Future Data” widget in Orange and did this for each available classification workflow retrieved from our ICON course “models” tab:



Features (9)
satisfaction_level
last_evaluation
project_count
average_monthly_hours
time_in_company
work_accident
promotion_last_5years
sales
salary
Target (1)
emp_left
Metas

Sampling Type
[✓ Fixed proportion of data:] 70 %
Fixed sample size
Instances: 2500
Sample with replacement
Cross validation
Number of subsets: 10
Unused subset: 1
Bootstrap
Options
[✓ Replicable (deterministic) sampling]
[✓ Stratify sample (when possible)]
12k 8390

[✓ Cross validation]
Number of folds: 10
[✓ Stratified]
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 66 %
[✓ Stratified]
Leave one out
Test on train data
Test on test data

Orange Workflow – General Layout (Training & Validation: CV-10, Test Set, Future Test Set)

Feature Engineering

The list of features available in our dataset:

- Satisfaction_level = self-reported employee satisfaction.
- Last_evaluation = measures the result of an employee’s latest performance review.
- Project_count = how many projects the employee is assigned.
- Average_monthly_hours = average time spent at work each month.
- Time_in_company = the employee’s tenure (in years).
- Work_accident = 1 if employee has experienced a work-related injury, 0 if not.
- Emp_left = 1 if the employee left his/her job, 0 if stayed with the company.

- Promotion_last_5years = 1 if offered a promotion in the last 5 years, 0 if not.
- Sales = the department name in which the employee works (IT, Sales, HR, etc.).
- Salary = low, medium, or high (arbitrarily assumed by us to be $50,000, $75,000, and $100,000, respectively)

Intuitively, our target variable is emp_left because it sets up a churn problem. So, emp_left serves as our target variable and the rest serve as predictors. Certainly, our predictors are correlated with each other, and multicollinearity exists. We are unconcerned with multicollinearity because our sole objective is to make predictions (not establish any notion of causality). So, none of these other variables are excluded during pre-processing. None of our data was mathematically altered or adjusted in any way as Kaggle conveniently represents each feature in the proper units of measurement. Besides that, we are uninterested in deriving marginal effects or ranking variable significance in the first place, so not much feature engineering is required to begin with.

Methods

To determine the best model for making future prediction, we examined logistic regression, KNN, single decision tree, random forest, gradient boosting, neural networks, and SVM based on AUC. We compare the AUC number reported by the "Test and Score" widget in Orange across models, and in some cases, we also reference the model's AUC from "Predictions on Test Set" to determine if the model was overfit (none were). For clarity: we did NOT use the performance criteria reported in "Predictions on Test Set" to select Random Forest. Random Forest was determined to the be the best option because it's AUC outperformed all other models in "Test and Score."

Moreover, our models were trained according to a 10-folds cross-validation process that utilized 70% of "HR_data_main.csv" or 8,390 random observations retrieved from the "Data Sampler" widget. Similarly, our test sets contained 30% of "HR_data_main.csv" or 3,595 random observations that were also retrieved from the "Data Sampler" widget:

Random Forest was the clear winner of our competition. We elected Random Forest because it returned the maximum AUC in "Test and Score" compared to Logistical Regress, KNN, Single Decision Tree, Gradient Boosting, Neural Networks, and Support Vector Machine.

Logistic Regression

Both Ridge and Lasso logistic regression gave an AUC of 0.820. None of the features were taken out of the model as they made the AUC decrease, resulting in a suboptimal output. A C=5 was used because it resulted in the highest AUC, being 0.820. Because logistic regression is the simplest model that we tested, we used it as a baseline for which to compare all the following models to.

KNN

The optimal KNN model based on AUC uses Manhattan distance and 23 neighbors (heuristically determined by adjusting the parameter settings in the Model Selection widget in 'knnclassification.ows'). Using stratified sampling and a 10-folds cross-validation process, we obtain an AUC of 0.985 from the 8390 observations sent to our training/validation workflow in Orange. We obtain a comparable result in the test set with 3595 observations or a 30% holdout: AUC=0.986. Therefore, the KNN model is not overfit and is a good choice for making future predictions because there is a strong probability that the model will assign larger probabilities to a random positive (emp_left=1) than a random negative (emp_left=0). Compared to the logistic regression model, KNN performed better and was therefore kept as an option for our final model.

Single Decision Tree

Based on the performance metrics obtained from running a single decision tree, we can conclude that the model performed remarkably well in predicting the target class 1, which denotes that an employee will leave. With an AUC of 0.980, the model achieved a near-perfect measure of how well it can distinguish between positive and negative instances. Additionally, the model's classification accuracy (CA) was found to be 0.973, which is a high level of overall correctness. The F1 score of 0.941 indicates that the model achieved a balance between precision and recall, with a precision of 0.977 and a recall of 0.908. Overall, these metrics suggest that the model is reliable and effective in predicting whether an employee is likely to leave. Compared to the logistic regression model, the single decision tree performed the same and was therefore removed as an option for our final model.

Random Forest

The first set of results (below) were achieved by using trial and error methods to manipulate the values under Random Forest widget. Setting the Number of trees to 200, the Number of attributes considered at each split to 7, limiting depth of individual trees to 10 and not splitting subsets smaller than 10 resulted in the highest AUC of 0.990. Compared to the logistic regression model, the random forest performed better and was therefore kept as an option for our final model.

Gradient Boosting

The gradient boosted ensemble reported below was realized by changing the setting parameters for number of trees and learning rate to get the best AUC. The Gradient Boosting(scikit-learn) had the highest AUC of 0.988. This result was obtained by setting the value for Number of Trees to 200, the Learning rate to 0.049, limiting the depth of individual trees to 3, and not splitting subsets smaller than 2. Compared to the logistic regression model, gradient boosting performed better and was therefore kept as an option for our final model.

Neural Networks

The optimal Neural Network model based on AUC establishes two hidden layers with an ensemble of 30 learners each (heuristically determined by adjusting the parameter settings in the Model Selection widget in Orange – a modelling software based on Python). Our data is passed through two hidden layers subject to a ReLu activation function and the algorithm is resolved by an L-BFGS-B solver function. This process yields an AUC equal to 0.98 from 10-folds cross-validation (Test and Score widget) and 0.981 from a 30% holdout (Predictions on Test Set widget), so the model is not overfit. Neural Networks are a good candidate for making predictions on future data *if not for underperforming more simplistic models like KNN and Random Forrest*. According to the Principle of Parsimony, Neural Networks are eliminated from consideration because better and more simplistic machine learning options exist at our disposal.

Support Vector Machine (SVM)

We ran a Support Vector Machine (SVM) to predict whether an employee would leave or not. The area under the curve (AUC) for the SVM was found to be 0.824, which indicates that the model is good at distinguishing between positive and negative classes. The classification accuracy (CA) was 0.798, which means that the model was able to classify 79.8% of the employees correctly. The F1 score was 0.497, indicating that the model has a below-average balance between precision and recall. The precision of the model was 0.617, which means that out of all the employees predicted to leave, 61.7% actually left. Finally, the recall was found to be 0.417, which means that out of all the employees who left, the model was able to correctly identify 41.7% of them. Overall, these performance metrics suggest that the SVM model has a below-average performance in predicting whether an employee will leave or not.

Compared to our baseline model, logistic regression, SVM did not perform well and hence was not selected as a contender for the final model.

Final Model Selection

After comparing the AUC of all the models, we created, we selected random forest as our final model. It had the highest AUC and therefore was the model that best fit and explained our dataset. We then used the same preprocessing rules as we did for the original model on the training and validation model. The fandom forest prediction on the test set, given an outcome of 1, resulted in an AUC value of 0.992.

Confusion Matrix

Using confusion Matrix, we corroborated the performance measures computed by Orange as follows:

Classification Accuracy:

- (TN + TP)/ (TN + TP + FN + FP) = (2726 + 792)/ (2726 + 792 + 6 + 71) = 0.978581.
- This shows that 98% of the employees were classified correctly.

Sensitivity/Recall:

- TP/ (FN +TP) = 792/ (6 +792) = 0.99248.
- This implies that 99% of the churning employees were predicted to churn.

Specificity:

- TN/ (TN + TP) = 2726/ (2726 + 792) = 0.774872.
- This indicates that 77% of non-churning employees were classified correctly.

Precision:

- TP/ (TP + FP) = 792/ (792 + 71) = 0.917729.
- This shows that 92% of employees predicted to churn do churn.

F1 Score:

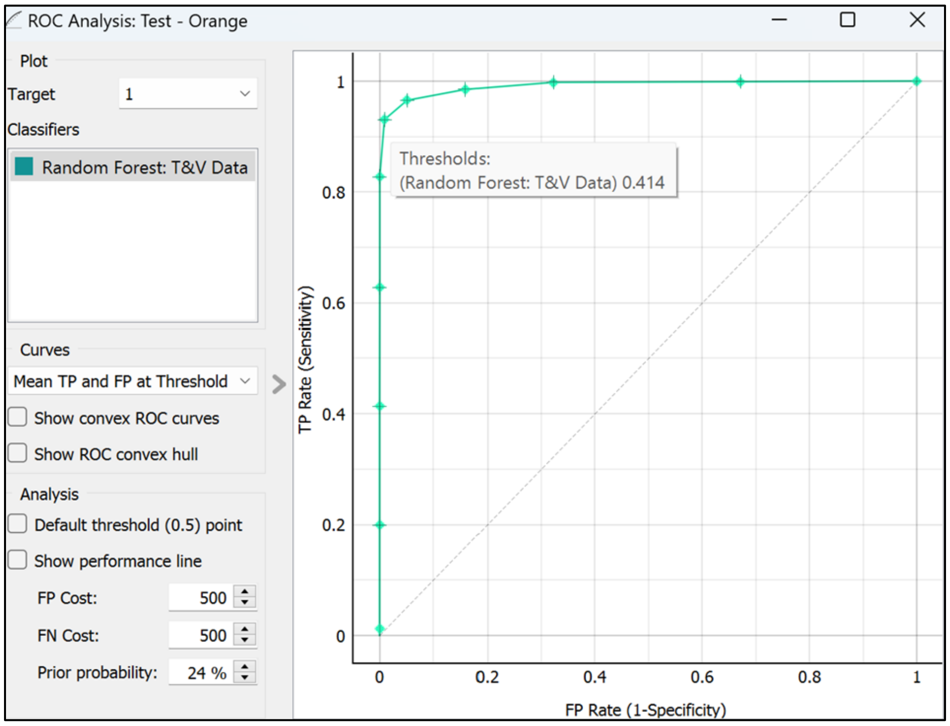
- TP + TP/ (FN +FP +TP + TP) = (792 + 792)/ (6 + 71 + 792 + 792) = 0.9536421.
- This means that the harmonic of the sensitivity and precision is 0.9536.
- F1 score informs the model balanced ability to identify churning employees by accurately predicting which employees are churning.

ROC Curve

The **AUC of 0.992** is the area under the ROC curve. This is generated by:

1. Computing a confusion matrix for each probability threshold between 1 and 0.
2. For each confusion matrix, computing the value of sensitivity(recall) and specificity.
3. Plotting the (x, y) point to (1-specificity, sensitivity) for each confusion matrix.

For instance, hovering over the point corresponding to (0.0021, 0.9460) shows that this combination was generated by using a probability threshold of 0.414. That is applying a probability threshold of 0.414 with the predictions random forest model, the sensitivity is 0.9460(94.6% of the churning employees are correctly identified) and the specificity is 1- 0.0021 = 0.9979(99.8% of the non-churning employees are correctly identified).



Target: 1

Classifiers: 🟩 Random Forest: T&V Data

Curves: Mean TP and FP at Threshold

☐ Show convex ROC curves

☐ Show ROC convex hull

☐ Default threshold (0.5) point

☐ Show performance line

FP Cost: 500

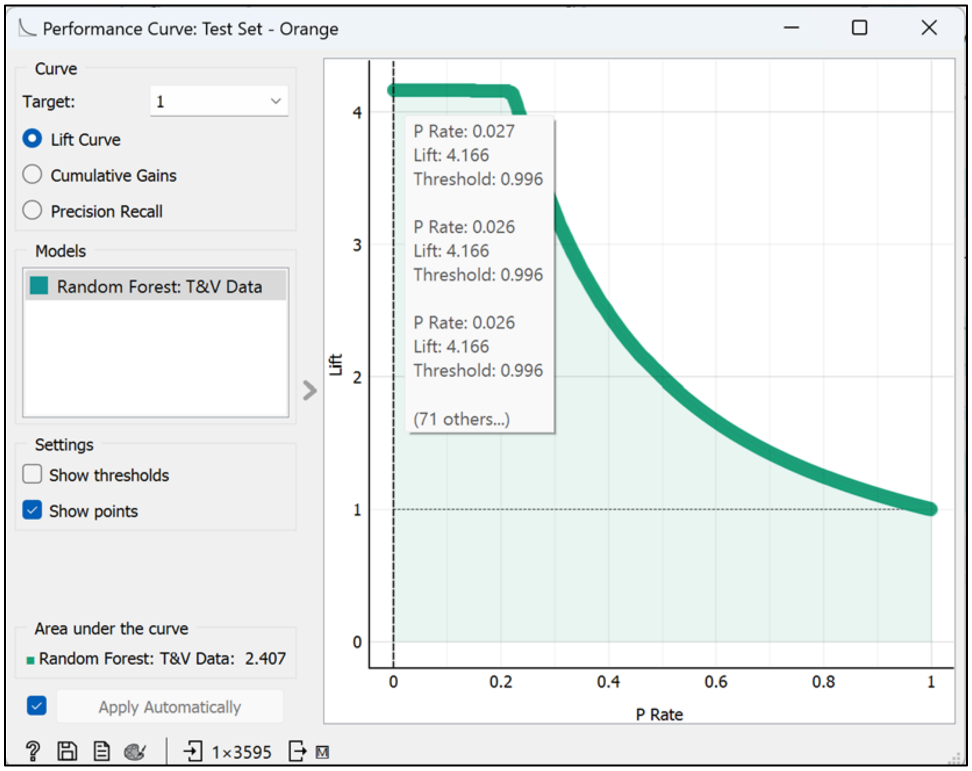
FN Cost: 500

Prior probability: 24 %

Lift Curve

The lift curve provided another way of comparing the random forest model to a random classifier. For instance, hovering over the curve reveals for the top 3% of the customers most likely to churn, the random forest model identified 4.166 times more churning employees than randomly selecting 3% of the 3595 individuals. Observing that there are 863 churning employees in the test set of 3595, we can conclude that:

$$\text{Lift} = 4.166 = 97/\ (0.027 * 3595)\ (863 \div 3595)$$



Profit/Cost Analysis

Our profit analysis attaches a $10,000 cost to offering a paid vacation to employees with a high probability of churning (emp_left=1). We assume that intervention always works i.e., a paid vacation will always retain an employee that may otherwise quit. To maximize profit, we use Random Forest to aim at only offering employees who have decided to quit a paid vacation, in order to persuade them otherwise and ultimately retain them.

Employees in our dataset earn "low" "medium" or "high" salaries. Kaggle does not provide their actual salary in dollars. We arbitrarily assume that low earners make $50,000 per year, medium earners make $75,000 per year, and high earners make $100,000 per year to formulate our profit curves. Furthermore, we assume that every employee generates the company an annual revenue equal to their salary such that:

- Low earners individually generate $50,000 in revenue per year for the firm
- Medium earners individually generate $75,000 in revenue per year for the firm
- High earners individually generate $100,000 in revenue per year for the firm

This is not a necessarily realistic assumption (since the company would expect to get more out of an employee than they pay in salary), but it works just fine for our purposes.

The cost of intervention ends up being $10,000 fixed, no matter what the employee earns i.e., we send every employee that is offered a paid vacation on the exact same trip. Therefore:

		Profit		Revenue		Cost
Profit true positive (low)	=	$40,000	=	$50,000	-	$10,000
Profit true positive (medium)	=	$65,000	=	$75,000	-	$10,000
Profit true positive (high)	=	$90,000	=	$100,000	-	$10,000
Profit false positive (all)	=	($10,000)	=	$0	-	$10,000
Profit true negative (all)	=	$0	=	$0	-	$0
Profit false negative	=	$0	=	$0	-	$0

Since the test set is a mixed population, we must first segment the data by "salary" and then generate 3 separate profit curves as opposed to just one (See Profit Curve – Low Salary, Profit Curve – Normal Salary, Profit Curve – High Salary)

Summarizing these results, we get:

Table 1. Profit Curves Summarized

Salary	n	Max Profit	nth employee	Random Forest p(1)
Low	1749	$19,350,000	550	0.236385
Medium	1555	$19,785,000	354	0.252054
High	291	$2,160,000	24	0.429304
Total	3595 (test set)	$41,295,000	928	

Translating predictions on future data into an action item: we recommend that the company sort observations by p(1) high-to-low and offer $10,000 trips to the first 31.45% of low earners (=550/1749) or any low-earning employee with an estimated probability of churning above or equal to 0.236385, the first 22.77% of medium earners (=354/1555) or any medium-earning employee with an estimated probability of churning above or equal to 0.252054, and the first 8.25% of high earners (=24/291) or any high-earning employee with an estimated probability of churning above or equal to 0.429304. Holding all else constant and according to our assumptions, this strategy will maximize expected profit.

Conclusion

Employee turnover is a growing problem in the United States, as well as the world. We came up with a solution to try to combat that problem. By offering employees a paid vacation as an incentive to stay with the company, we would be able to cut turnover costs as well as increase the profit coming from them. There would have to be some stipulations around the vacation incentive including the time they have been with the company, how much revenue they generate yearly, and their starting salary. Employers don't want to be giving vacations to 31% of the low salary employees and only 8% of the high salary employees, as was determined from out data sample.

Incentivizing your employees to stay with the company vs leave to look for new jobs elsewhere is a good strategy for retention.

Regarding the dataset that we looked at and the results we obtained, it was a little skewed in favor of the low salary earners and not even across the board for all salary levels. There are substantially more low and medium salary earners than high level salary earners, but in the real-world, the free vacations would be more evenly distributed percentage wise for a given number of employees. The breakdown will be different in each industry type, but this method can be used for any future data to determine how many people you can incentivize to stay.

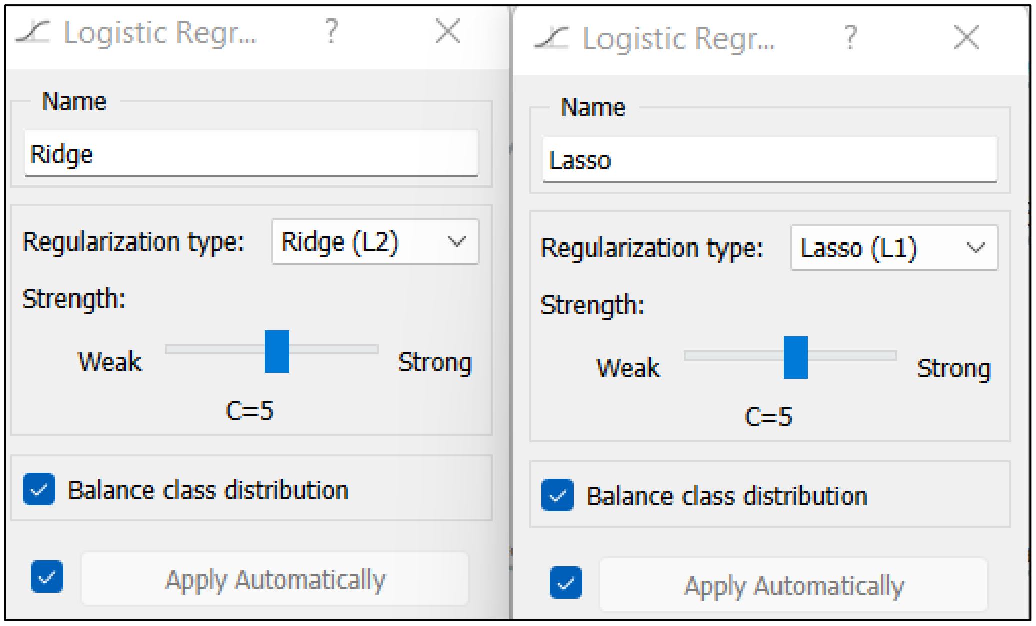
References

- Gartner, Inc (2022). Gartner Says U.S. Total Annual Employee Turnover Will Likely Jump by Nearly 20% From the Pre-pandemic Annual Average.https://www.gartner.com/en/newsroom/04-28-2022-gartner-says-us-total-annual-employee-turnover-will-likely-jump-by-nearly-twenty-percent-from-the-prepandemic-annual-average. Retrieved on 4/27/2023.

- Zippia."27 US Employee Turnover Statistics [2023]: Average Employee Turnover Rate, Industry Comparisons, And Trends" Zippia.com. Feb. 7, 2023.https://www.zippia.com/advice/employee-turnover-statistics/ . Retrieved on 4/27/2023.

Appendix

Figure 1: Logistic Regression (Lasso & Ridge) | CV-10 Experiment on Training & Validation Set



Logistic Regr...

Name

Ridge

Regularization type: Ridge (L2)

Strength:

Weak Strong

C=5

✓ Balance class distribution

✓ Apply Automatically

Logistic Regr...

Name

Lasso

Regularization type: Lasso (L1)

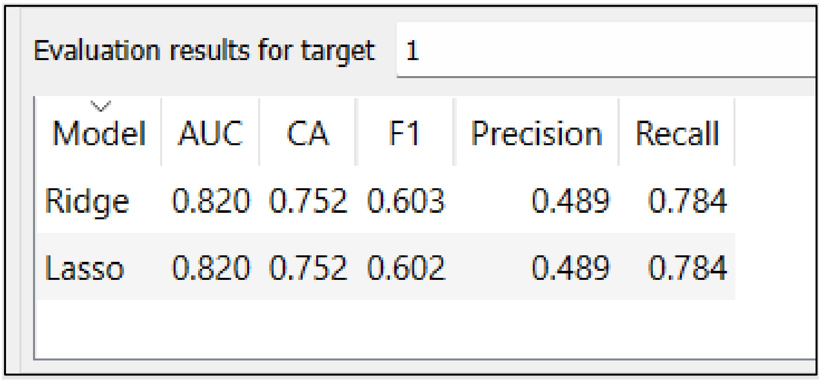
Strength:

Weak Strong

C=5

✓ Balance class distribution

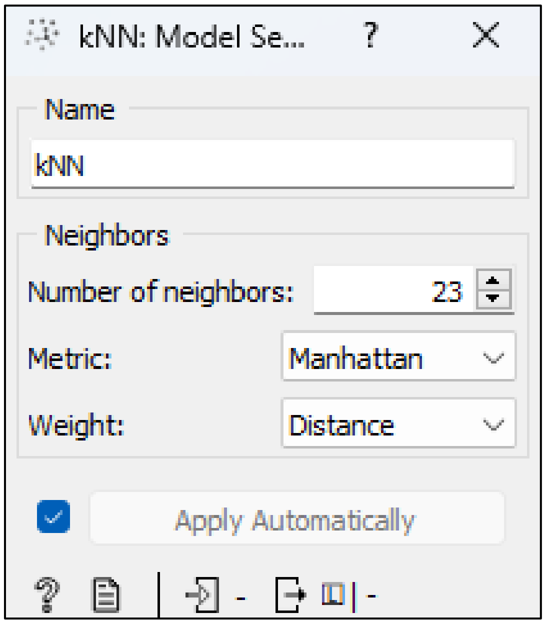
✓ Apply Automatically



Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
Ridge	0.820	0.752	0.603	0.489	0.784
Lasso	0.820	0.752	0.602	0.489	0.784

Figure 2: K-Nearest Neighbors (KNN) | CV-10 Experiment on Training & Validation Set


kNN: Model Se...

Name: kNN

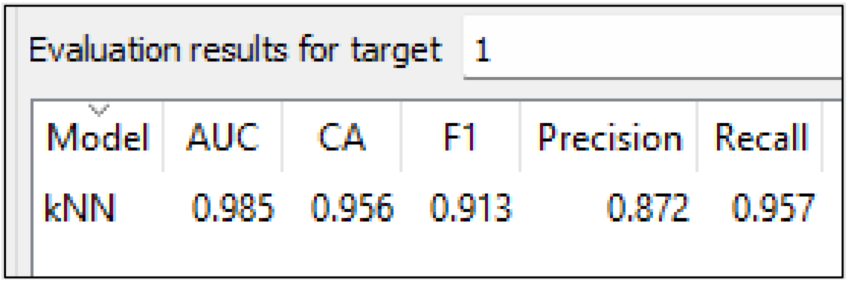
Neighbors

Number of neighbors: 23

Metric: Manhattan

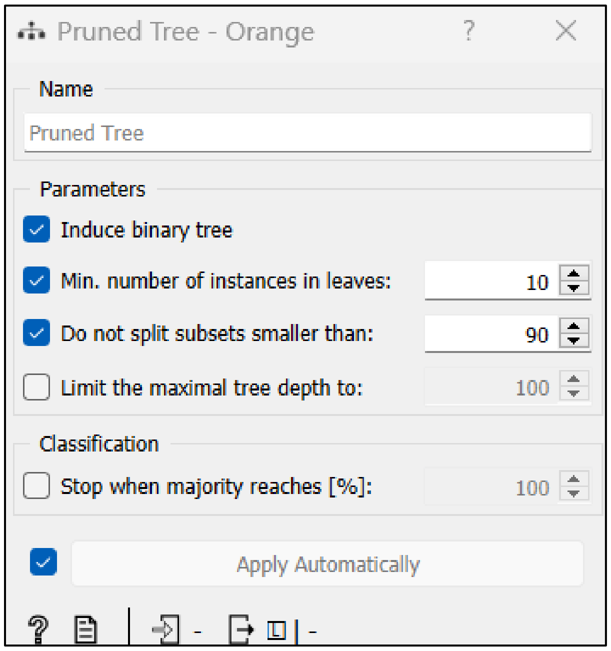
Weight: Distance

✓ Apply Automatically


Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
kNN	0.985	0.956	0.913	0.872	0.957

Figure 3: Single Decision Tree (Pruned) | CV-10 Experiment on Training & Validation Set


Pruned Tree - Orange

Name: Pruned Tree

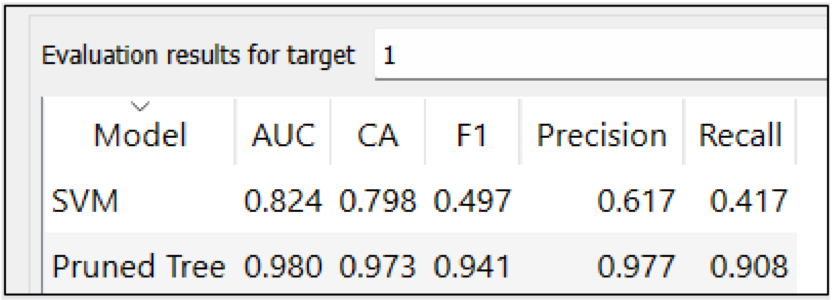
Parameters

- [✓] Induce binary tree
- [✓] Min. number of instances in leaves: 10
- [✓] Do not split subsets smaller than: 90
- [ ] Limit the maximal tree depth to: 100

Classification

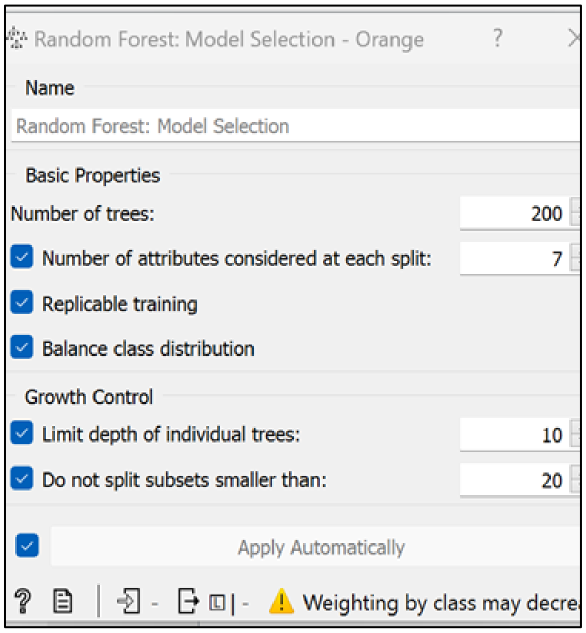
- [ ] Stop when majority reaches [%]: 100

[✓] Apply Automatically


Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
SVM	0.824	0.798	0.497	0.617	0.417
Pruned Tree	0.980	0.973	0.941	0.977	0.908

Figure 4: Random Forest | CV-10 Experiment on Training & Validation Set



Random Forest: Model Selection - Orange

Name

Random Forest: Model Selection

Basic Properties

Number of trees: 200

✓ Number of attributes considered at each split: 7

✓ Replicable training

✓ Balance class distribution

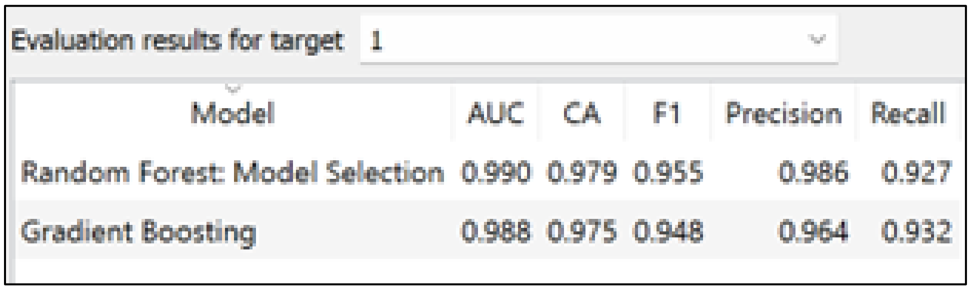
Growth Control

✓ Limit depth of individual trees: 10

✓ Do not split subsets smaller than: 20

✓ Apply Automatically

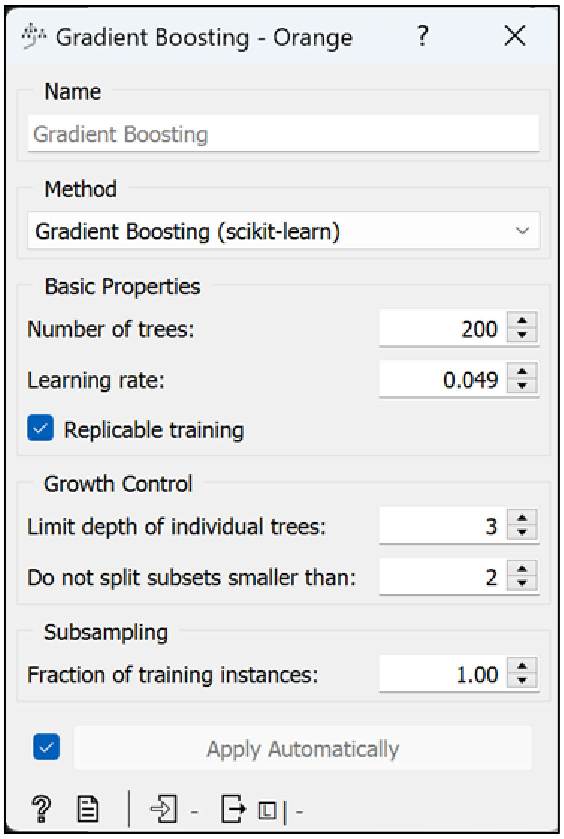
Weighting by class may decre



Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
Random Forest: Model Selection	0.990	0.979	0.955	0.986	0.927
Gradient Boosting	0.988	0.975	0.948	0.964	0.932

Figure 5: Gradient Boosting | CV-10 Experiment on Training & Validation Set



Gradient Boosting - Orange

Name

Gradient Boosting

Method

Gradient Boosting (scikit-learn)

Basic Properties

Number of trees: 200

Learning rate: 0.049

✓ Replicable training

Growth Control

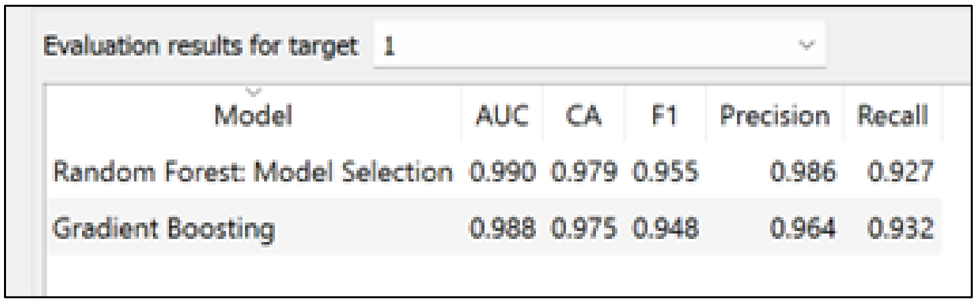
Limit depth of individual trees: 3

Do not split subsets smaller than: 2

Subsampling

Fraction of training instances: 1.00

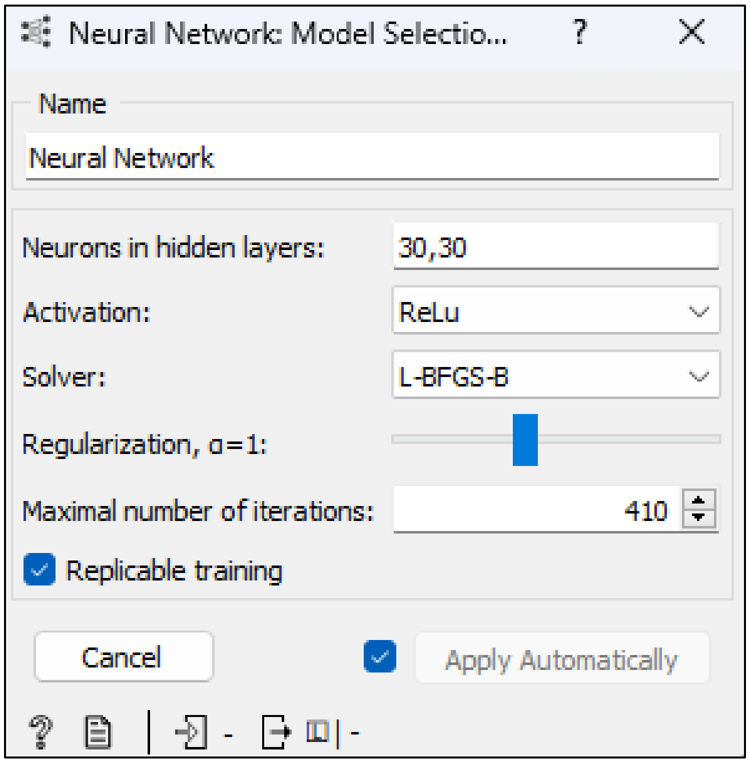
✓ Apply Automatically



Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
Random Forest: Model Selection	0.990	0.979	0.955	0.986	0.927
Gradient Boosting	0.988	0.975	0.948	0.964	0.932

Figure 6: Neural Networks | CV-10 Experiment on Training & Validation Set



Neural Network: Model Selectio...

Name

Neural Network

Neurons in hidden layers: 30,30

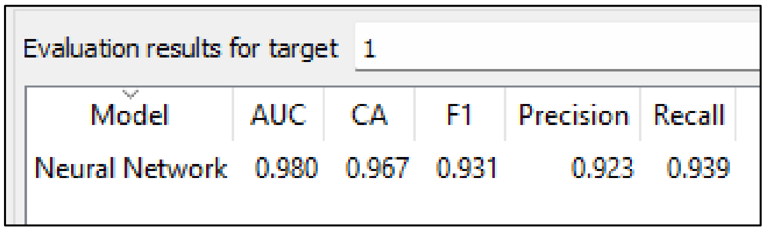
Activation: ReLu

Solver: L-BFGS-B

Regularization, α=1:

Maximal number of iterations: 410

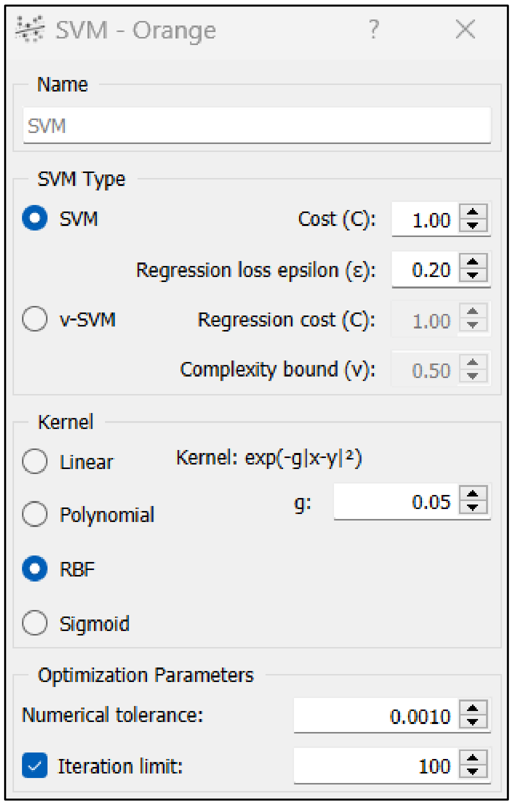
✓ Replicable training



Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
Neural Network	0.980	0.967	0.931	0.923	0.939

Figure 7: Support Vector Machine (SVM) | CV-10 Experiment on Training & Validation Set


SVM - Orange

Name: SVM

SVM Type

[✓ SVM] Cost (C): 1.00

Regression loss epsilon (ε): 0.20

v-SVM Regression cost (C): 1.00

Complexity bound (v): 0.50

Kernel

Linear

Polynomial

[✓ RBF]

Sigmoid

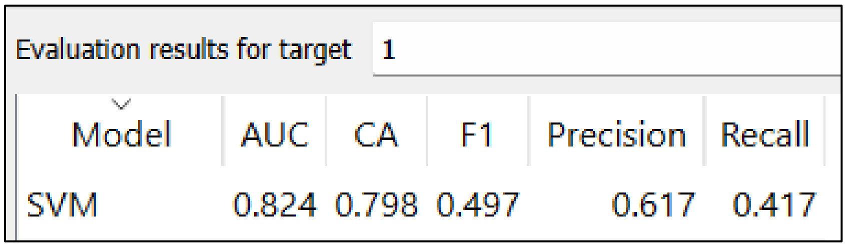
Kernel: exp(-g|x-y|²)

g: 0.05

Optimization Parameters

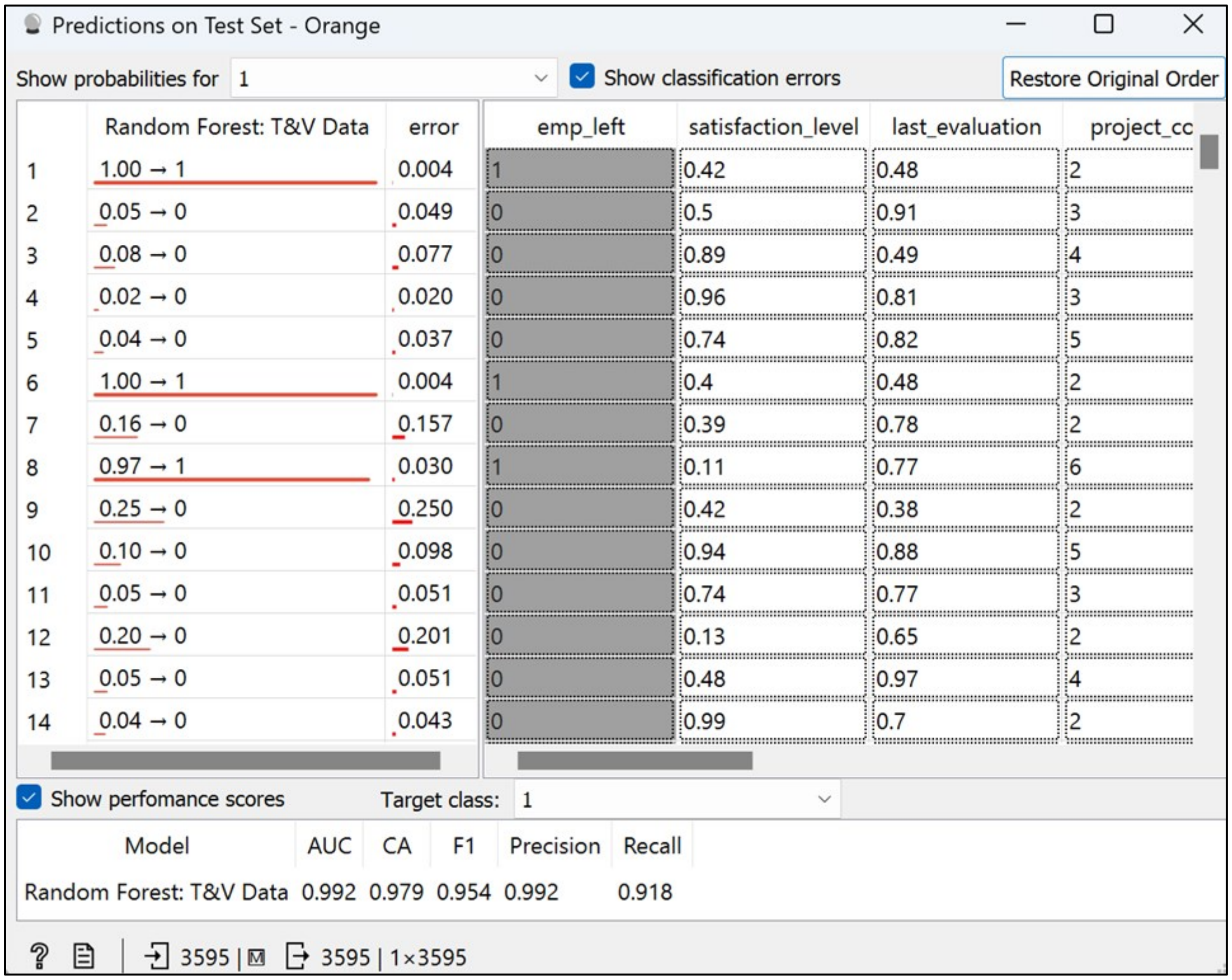
Numerical tolerance: 0.0010

[✓] Iteration limit: 100


Evaluation results for target 1

Model	AUC	CA	F1	Precision	Recall
SVM	0.824	0.798	0.497	0.617	0.417

Figure 8: Selected Model (Random Forest) Performance | Test Set Performance



Predictions on Test Set - Orange

Show probabilities for 1 | ✓ Show classification errors | Restore Original Order

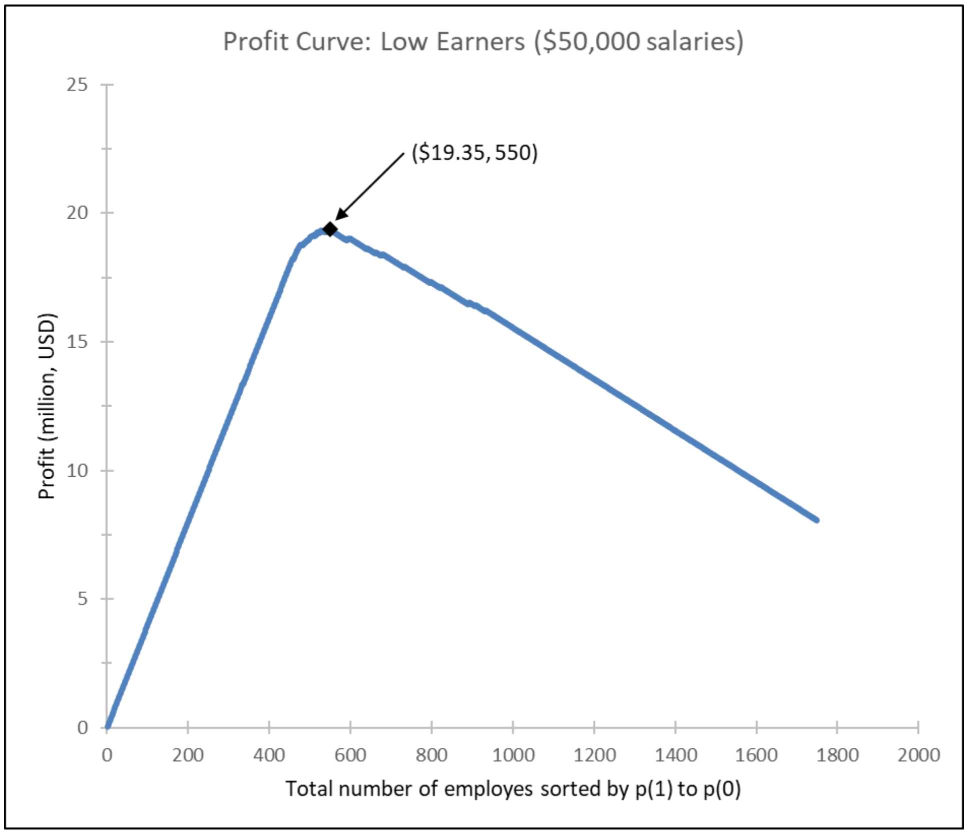
	Random Forest: T&V Data	error	emp_left	satisfaction_level	last_evaluation	project_cc
1	1.00 → 1	0.004	1	0.42	0.48	2
2	0.05 → 0	0.049	0	0.5	0.91	3
3	0.08 → 0	0.077	0	0.89	0.49	4
4	0.02 → 0	0.020	0	0.96	0.81	3
5	0.04 → 0	0.037	0	0.74	0.82	5
6	1.00 → 1	0.004	1	0.4	0.48	2
7	0.16 → 0	0.157	0	0.39	0.78	2
8	0.97 → 1	0.030	1	0.11	0.77	6
9	0.25 → 0	0.250	0	0.42	0.38	2
10	0.10 → 0	0.098	0	0.94	0.88	5
11	0.05 → 0	0.051	0	0.74	0.77	3
12	0.20 → 0	0.201	0	0.13	0.65	2
13	0.05 → 0	0.051	0	0.48	0.97	4
14	0.04 → 0	0.043	0	0.99	0.7	2

✓ Show perfomance scores | Target class: 1

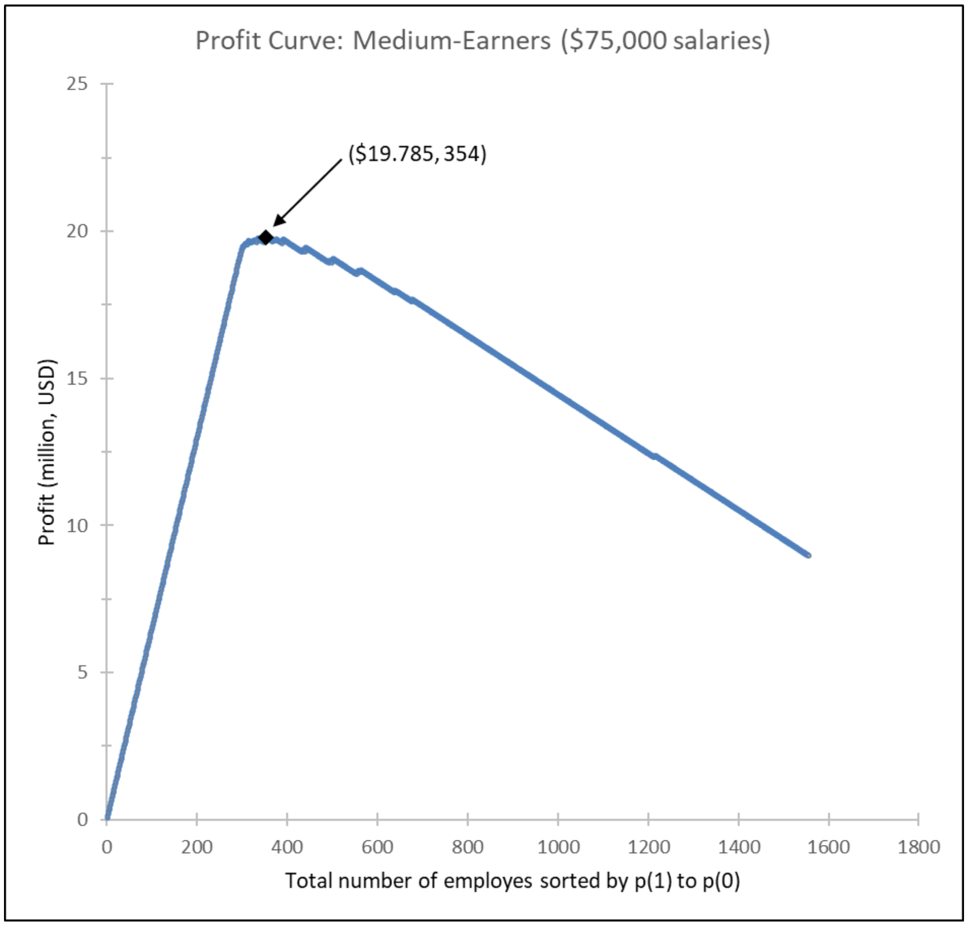
Model	AUC	CA	F1	Precision	Recall
Random Forest: T&V Data	0.992	0.979	0.954	0.992	0.918

3595 | 3595 | 1×3595

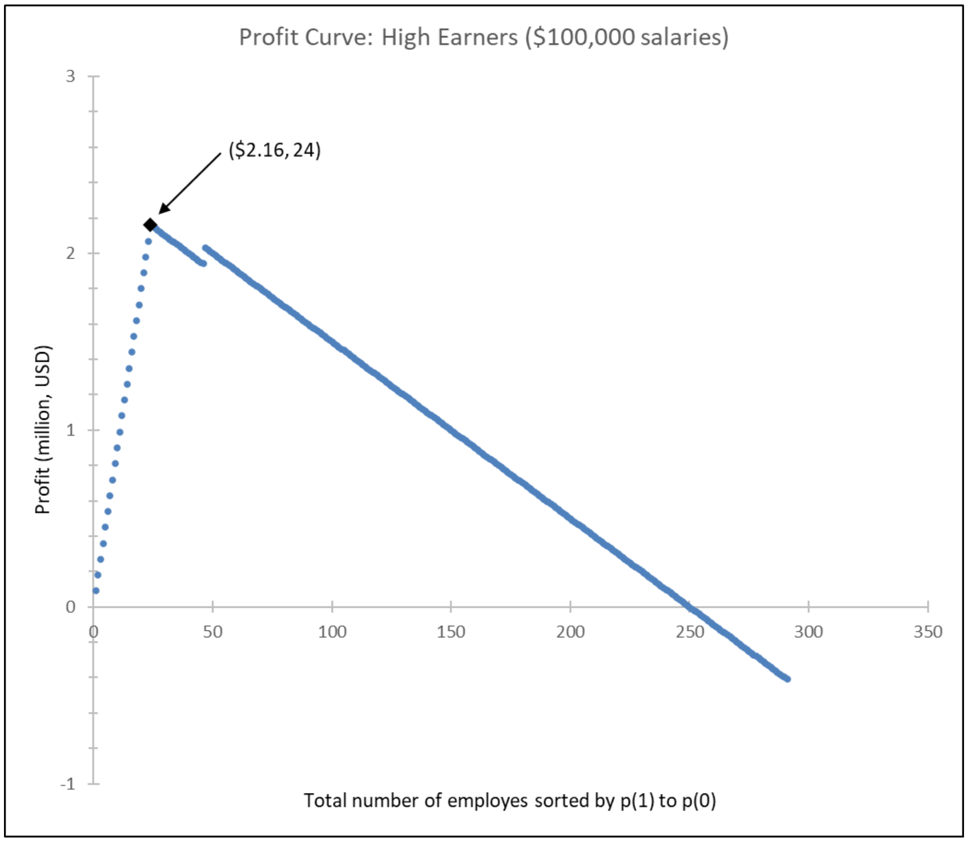
Profit Curve – Low



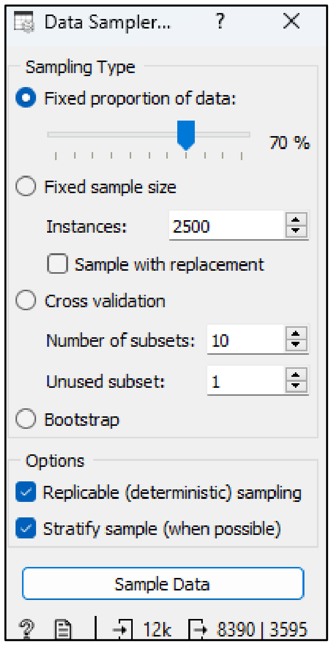
Profit Curve – Normal Salary



Profit Curve – High Salary



Data Sampler



Data Sampler...

Sampling Type

[✓ Fixed proportion of data:] 70 %

Fixed sample size

Instances: 2500

Sample with replacement

Cross validation

Number of subsets: 10

Unused subset: 1

Bootstrap

Options

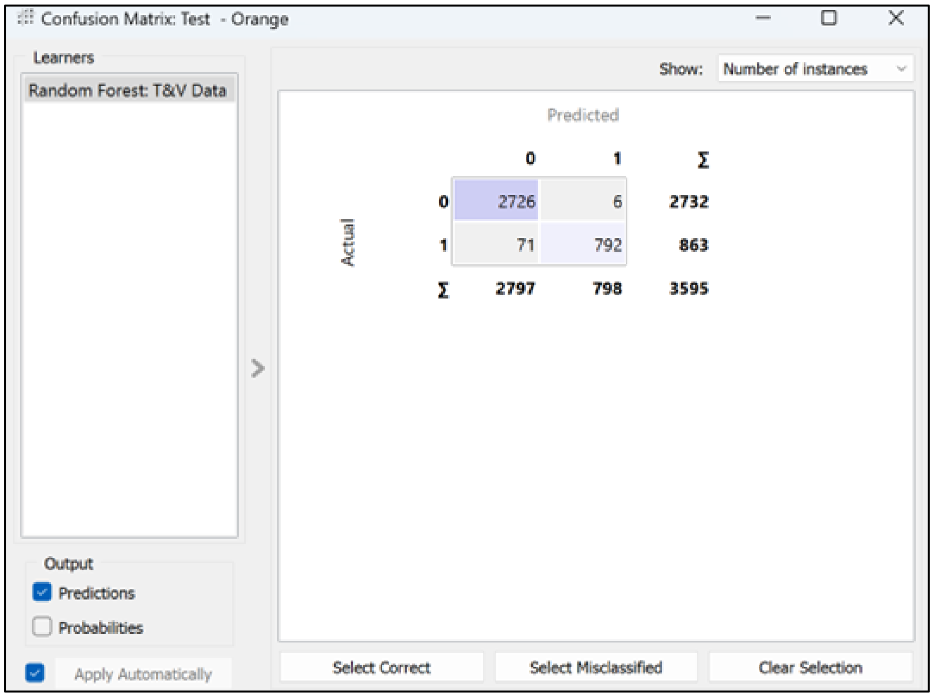
[✓ Replicable (deterministic) sampling]

[✓ Stratify sample (when possible)]

Sample Data

12k 8390 | 3595

Confusion Matrix



Confusion Matrix: Test - Orange

Learners

Random Forest: T&V Data

Show: Number of instances

		Predicted		
		0	1	Σ
Actual	0	2726	6	2732
	1	71	792	863
	Σ	2797	798	3595

Output

[✓ Predictions]

Probabilities

[✓ Apply Automatically]

Select Correct | Select Misclassified | Clear Selection