Scenario 1: Human Development Index (HDI) Prediction

1.Type of Learning:

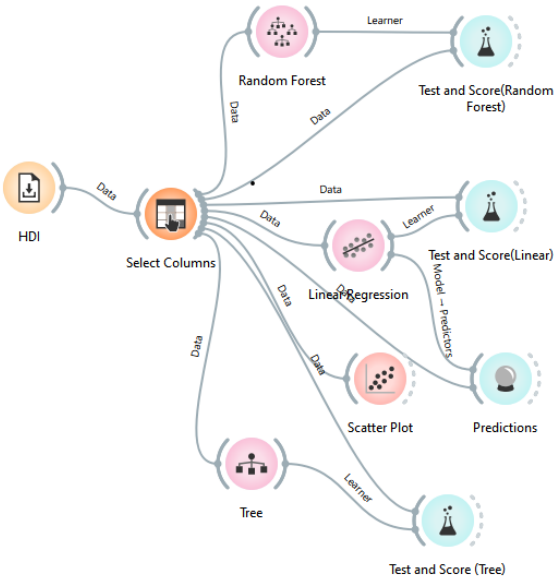
The Human Development Index (HDI) prediction is a Supervised Learning problem. In this case, the goal is to predict the HDI of different countries based on known input variables such as life expectancy, mean years of schooling, and gross national income (GNI).
Since HDI is a continuous numeric value, the task falls under Regression, where the model learns from existing data with known outputs to predict future values.

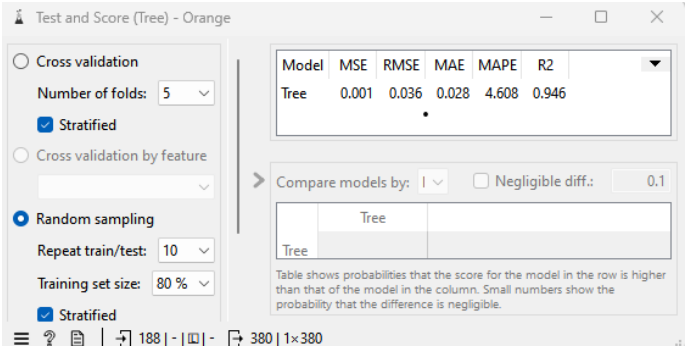
2.Selected Algorithm and Justification

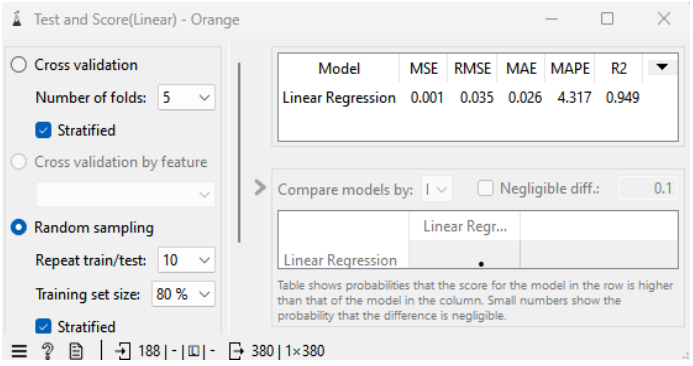
For this task, I tested three different regression algorithms which are Random Forest Regression, Linear Regression, and Tree Regression.
After comparing their results, I selected Random Forest Regression as the most suitable algorithm.

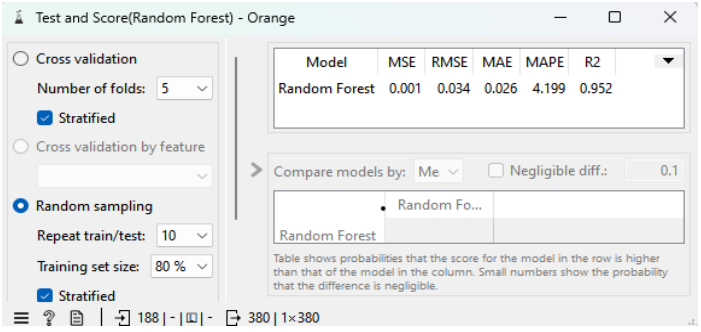
The main reason for choosing Random Forest is that it can handle complex, non-linear relationships among variables much better than a simple linear model. It uses multiple decision trees and averages their results, which makes the predictions more accurate and avoids overfitting.

From the results, it is clear that Random Forest Regression performed the best. It achieved the highest R^2 value (0.952) and the lowest RMSE (0.034), which indicates that the model explained about 95% of the variation in HDI and made very accurate predictions.
The Linear Regression model also performed well but was slightly less accurate, while the Tree Regression had the weakest results.


Random Forest
Learner
Test and Score(Random Forest)
Data
Data
HDI
Data
Select Columns
Data
Data
Learner
Test and Score(Linear)
Linear Regression
Model → Predictors
Data
Data
Data
Scatter Plot
Predictions
Tree
Learner
Test and Score (Tree)


Test and Score (Tree) - Orange
Cross validation
Number of folds: 5
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 80 %
Stratified
Model MSE RMSE MAE MAPE R2
Tree 0.001 0.036 0.028 4.608 0.946
Compare models by:
Negligible diff.: 0.1
Tree
Tree
Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.
188 | - | - 380 | 1×380


Test and Score(Linear) - Orange
Cross validation
Number of folds: 5
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 80 %
Stratified
Model MSE RMSE MAE MAPE R2
Linear Regression 0.001 0.035 0.026 4.317 0.949
Compare models by:
Negligible diff.: 0.1
Linear Regr...
Linear Regression
Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.
188 | - | - 380 | 1×380


Test and Score(Random Forest) - Orange
Cross validation
Number of folds: 5
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 80 %
Stratified
Model MSE RMSE MAE MAPE R2
Random Forest 0.001 0.034 0.026 4.199 0.952
Compare models by:
Negligible diff.: 0.1
Random Fo...
Random Forest
Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.
188 | - | - 380 | 1×380

Scenario 2: Employee Attrition Analysis

1. Type of Learning

This is a Supervised Learning problem. We are predicting a labeled outcome, whether an employee will leave the company (yes/no). Since the target variable is categorical, this is specifically a classification task.

2.Selected Algorithm and Justification

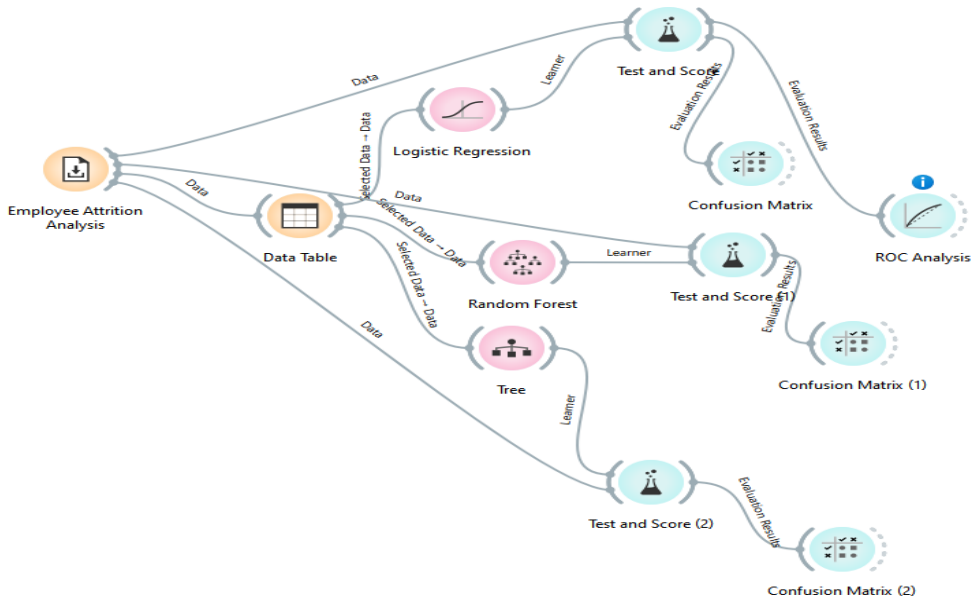
For this task as well I tested three different algorithms Logistic Regression, Random Forest and Decision Tree. After testing all three models, Logistic Regression was the best. It had the highest precision (0.866) and recall (0.878), meaning it correctly identifies most employees likely to leave while making few mistakes. Random Forest missed many employees who left, and Decision Tree did okay but was not as good as Logistic Regression.

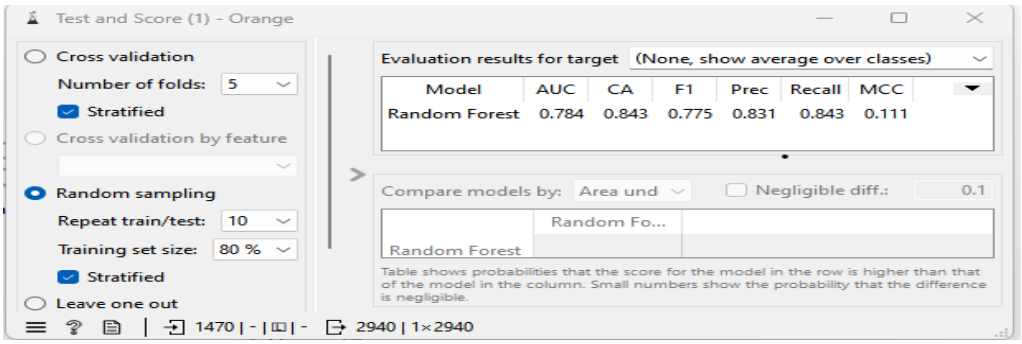
I also looked at the confusion matrix for all three models to see how well they predicted employees leaving or staying.

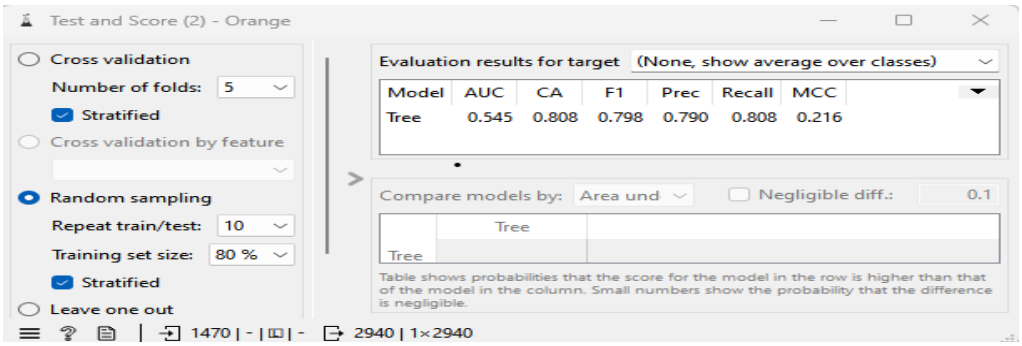
- Logistic Regression: Correctly predicted most employees who stayed (TN = 2,383) and a good number who left (TP = 199), with a few missed (FN = 271).
- Random Forest: Predicted almost everyone would stay (TP = 10), so it missed most employees who left.
- Decision Tree: Did better than Random Forest (TP = 135) but was still not as good as Logistic Regression.

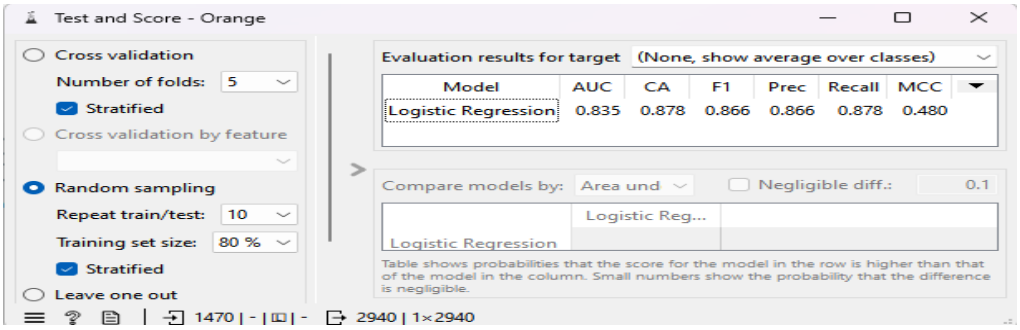
From this, it is clear that Logistic Regression is the best model, as it balances identifying employees at risk and minimizing false predictions.

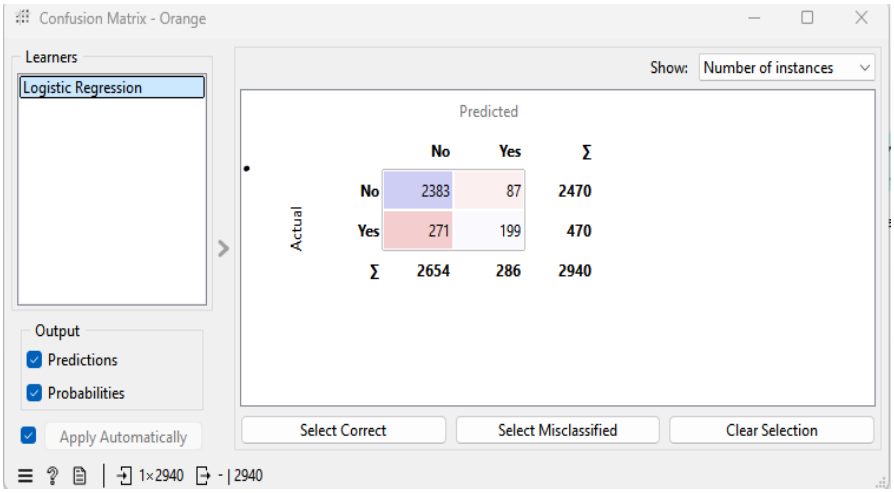
Please see my all attachments to get clear view.

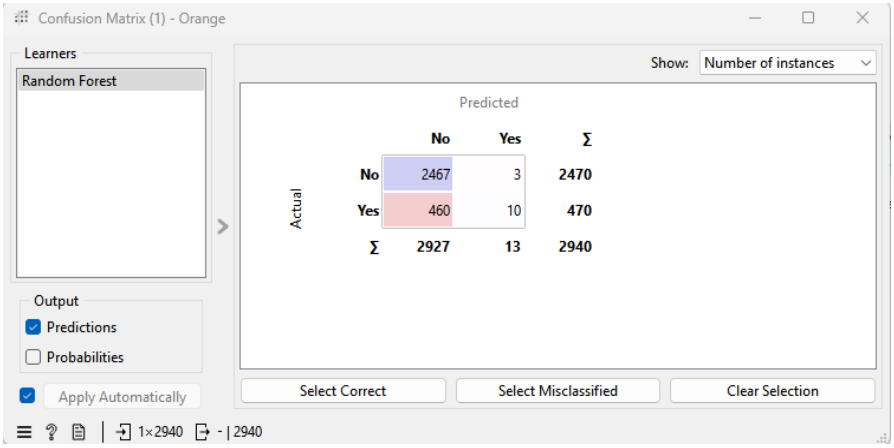

Employee Attrition Analysis
Data Table
Logistic Regression
Random Forest
Tree
Test and Score
Test and Score (1)
Test and Score (2)
Confusion Matrix
Confusion Matrix (1)
Confusion Matrix (2)
ROC Analysis
Data
Learner
Evaluation Results
Selected Data → Data

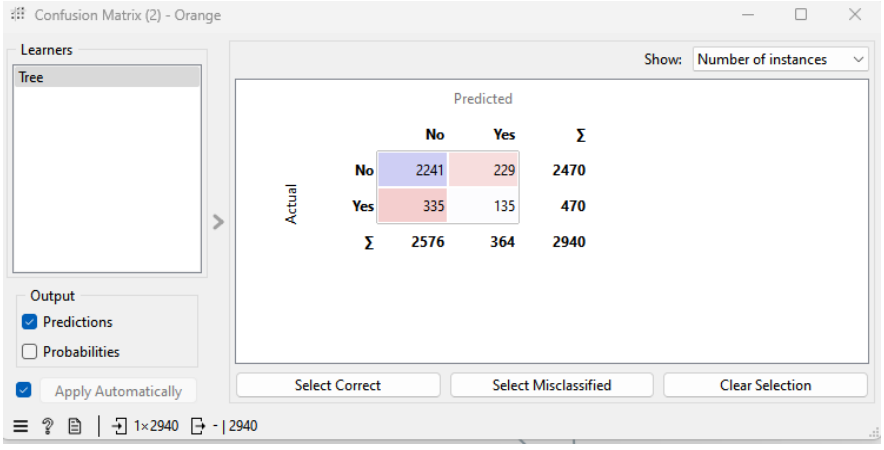

Test and Score (1) - Orange
Cross validation
Number of folds: 5
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 80 %
Stratified
Leave one out
Evaluation results for target (None, show average over classes)
Model AUC CA F1 Prec Recall MCC
Random Forest 0.784 0.843 0.775 0.831 0.843 0.111
Compare models by: Area und
Negligible diff.: 0.1
Random Fo...
Random Forest
Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.
1470 | - | - | 2940 | 1×2940


Test and Score (2) - Orange
Cross validation
Number of folds: 5
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 80 %
Stratified
Leave one out
Evaluation results for target (None, show average over classes)
Model AUC CA F1 Prec Recall MCC
Tree 0.545 0.808 0.798 0.790 0.808 0.216
Compare models by: Area und
Negligible diff.: 0.1
Tree
Tree
Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.
1470 | - | - | 2940 | 1×2940


Test and Score - Orange
Cross validation
Number of folds: 5
Stratified
Cross validation by feature
Random sampling
Repeat train/test: 10
Training set size: 80 %
Stratified
Leave one out
Evaluation results for target (None, show average over classes)
Model AUC CA F1 Prec Recall MCC
Logistic Regression 0.835 0.878 0.866 0.866 0.878 0.480
Compare models by: Area und
Negligible diff.: 0.1
Logistic Reg...
Logistic Regression
Table shows probabilities that the score for the model in the row is higher than that of the model in the column. Small numbers show the probability that the difference is negligible.
1470 | - | - | 2940 | 1×2940


Confusion Matrix - Orange
Learners
Logistic Regression
Show: Number of instances
Predicted
No Yes Σ
No 2383 87 2470
Yes 271 199 470
Σ 2654 286 2940
Actual
Output
Predictions
Probabilities
Apply Automatically
Select Correct
Select Misclassified
Clear Selection
1×2940
- | 2940


Confusion Matrix (1) - Orange
Learners
Random Forest
Show: Number of instances
Predicted
No Yes Σ
No 2467 3 2470
Yes 460 10 470
Σ 2927 13 2940
Actual
Output
Predictions
Probabilities
Apply Automatically
Select Correct
Select Misclassified
Clear Selection
1×2940
- | 2940


Confusion Matrix (2) - Orange
Learners
Tree
Show: Number of instances
Predicted
No Yes Σ
No 2241 229 2470
Yes 335 135 470
Σ 2576 364 2940
Actual
Output
Predictions
Probabilities
Apply Automatically
Select Correct
Select Misclassified
Clear Selection
1×2940
- | 2940

Scenario 3: Advertising

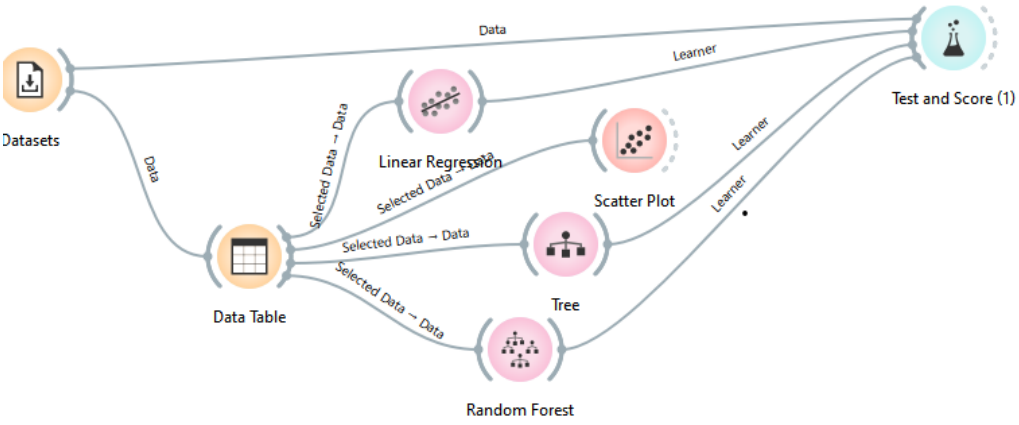
1.Type of learning

This task is a Supervised Learning problem because we already know the sales values and we are trying to train the model to predict sales based on inputs like TV, radio, and newspaper advertising.
Since the target (sales) is a number, this is a Regression type of learning. The model learns from the data where both inputs and outputs are known so it can make predictions for new cases.

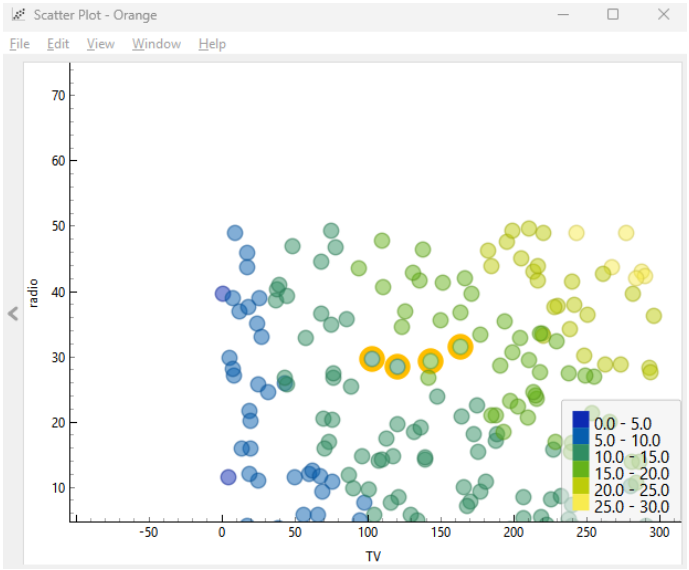
2. Selected Algorithm and Justification

For this scenario, I used Tree Regression, Random Forest Regression, and Linear Regression. After comparing the results, I found that **Tree Regression** gave the best outcome with an R^2 value of **0.955**, which means it predicted sales very accurately. Model explained about 95% of the variation in sales, which shows it predicted very accurately.

I chose Tree Regression because it works well with data that has non-linear relationships. In real life, increasing TV or radio spending doesn't always increase sales in a straight line, so the Tree model is better at showing these patterns. Random Forest also worked well but was a little less accurate, while Linear Regression didn't fit the data very well because the relationship was not purely linear. The scatters plot also showed that the Tree model's predictions were closest to the actual sales line.



Model	MSE	RMSE	MAE	MAPE	R2
Tree	1.245	1.116	0.861	7.504	0.955
Linear Regression	3.082	1.756	1.359	14.478	0.889
Random Forest	2.373	1.540	1.185	10.375	0.915



I used Orange for all my workflows. I added the datasets, connected the widgets for each task, and tested different models like Linear Regression, Tree Regression, Random Forest, and Logistic Regression. I checked how well the models worked using Test & Score, Confusion Matrix, and plots, which helped me pick the best model for each scenario