## **Data Robot Vs RapidMiner**

## **Introduction**

This report explores the comparative performance of manual and automated machine learning approaches on “Adult income dataset” (link mentioned in the Appendix) a dataset chosen from Kaggle repository. The main objective is to analyse the level of agreement between approaches such as DataRobot Autopilot Mode, RapidMiner AutoML, and a custom-built manual pipeline in RapidMiner, while exploring the factors behind any observed similarities or differences. The selected dataset involves predicting whether individuals earn more or less than $50,000 annually based on a range of demographic and financial variables. By Comparing theoretical knowledge with practical application, the findings aim to provide valuable insights into the strengths and limitations of different machine learning approaches which can contribute on solving complex real world business challenges.

## **Kaggle Python Notebook**

A Kaggle python notebook linked to the “Adult Income dataset“ effectively combines exploratory data analysis (EDA) with machine learning to build a predictive model. The EDA starts with data cleaning, handling the missing values in columns like ‘workclass’ and ‘occupation’ using python’s Panda's library. Categorical variables, such as ‘marital status’ and ‘education’, are encoded into numerical formats through tools like ‘LabelEncoder’ to prepare them for machine learning. Visualizations created with ‘Matplotlib’ and ‘Seaborn’ provide key insights, highlighting the trends like strong correlation between education and income levels, as well as the relationship between hours worked and income.

The notebook also implements preprocessing techniques, including scaling numeric features like ‘age’ and transforming the ‘income’ column into binary categories, essential for classification tasks. Finally, it applies and evaluates the Logistic Regression model using ‘Scikit-learn’, with metrics like accuracy and confusion metrics guiding performance assessment. This Kaggle notebook over all used and evaluated different models including Logistic Regression, Decision Tree, Random Forest, KNN and XGBoosting. However, after applying and evaluating different models with various metrics, the author concluded

The best two models as per Kaggle workbook are ‘Decision Tree’ and ‘XGBoosting’. And to pick one model as best is ‘XGBoosting’ as it has accuracy of **83.91%** and is more efficient and faster than Decision Tree model.

## **Data Robot Work**

The Adult Income dataset from Kaggle was analysed using DataRobot’s Autopilot Mode, by importing and choosing the column ‘Income’ as target variable. This variable classifies individuals into either >50k or <=50k income groups. And, as part of data quality assessment DataRobot evaluates all the input attributes and anomalies are handled automatically. However, it used all features in the dataset to ensure no critical information is excluded during training. In total, DataRobot analysed 58 various inbuilt models and recommended the “eXtreme Gradient Boosted Trees Classifier with Early Stopping” as the most effective model with as shown in *figure1* in appendix and the blueprint of it as shown in *figure2*. DataRobot output dashboard further shows a clear understanding of the model’s performance.

The **Lift chart** in *figure3* represents that the model could rank high- income cases by probability, as seen in its steep upward trend in the top bins, outperforming the random predictions. Likewise, the Prediction Distribution plot clearly separated into two income groups.

The **ROC curve** in the *figure4* evaluates the performance of a classification model that distinguishes between two groups (>50k and <=50k). It plots the True Positive Rate (Sensitivity) against the False Positive Rate (Fallout) across various classification thresholds. The curve rises steeply at the beginning indicates the model accurately identifies high-income cases(>50k) with minimal misclassification of low-income cases(<=50K). The closer the curve is to the top left corner, the better the model’s classification ability. The diagonal "Random" line serves as a baseline, showing the performance of a random guess classifier, the ROC curve being above this line highlights the model’s strong performance respectively. The **Confusion matrix** as in *figure5* presenting thedetails as below

* True Negatives (6,789): Correctly identified individuals earning <=50k
* False Positives (642): Cases mistakenly classified as >50K
* False Negatives (611): Cases mistakenly classified as <=50k
* True Positives (1,726): Correctly identified individuals earning >50k

The model achieved an impressive accuracy of **87.17%**, highlighting its consistent performance across the dataset. Other key metrics included a Precision of 72.89%, reflecting the proportion of correct predictions among all classifications, and a Recall of 73.86%, which measured the model’s ability to capture true high-income cases. The balanced F1 Score of 0.7337 demonstrated the model’s capability to manage trade-offs between Precision and Recall effectively, a critical requirement for imbalanced classification issues.

## **RapidMiner - Auto ML**

Using RapidMiner’s AutoML, we followed a similar approach to DataRobot by loading the dataset and executing the automated machine learning process by choosing target attribute as ‘Income’ for prediction. RapidMiner AutoML recommends models by default, however we chose few models manually for the further evaluation are Logistic Regression, Decision Tree, Random Forest and Gradient Boosting. The total number of models AutoML has applied on this dataset is 140, which is comparatively greater than DataRobot.

The output overview of the AutoML process is shown in *figure 6,* which demonstrates the performance of different models based on the evaluation of different metrics. It is observed that AutoML has chosen Logistic Regression as the best model for prediction as it shows the best result in performance, scoring time, Gain and time taken for model execution based on classification error metric. Unless Manual process, In AutoML it is easy to compare different models at a same time based on the evaluation metrics. In detail for each model, we can compare and analyze the outcomes of simulator, performance, predictions and lift chart. As per AutoML, The Logistic regression model has an accuracy of **85.1%** which is comparatively lesser than the DataRobot’s accuracy by “Xtreme Gradient Boosted Trees Classifier with Early Stopping” model and lesser accuracy than the Manually created pipelines.

## **RapidMiner - Manual**

Manual RapidMiner pipeline was developed for the same dataset followed by various data cleaning processes including Removed duplicates, handled outliers and replaced special characters and handled columns with predominant Zero’s with setting up binary flags (0,1) using Generate attributes operator for better data classification. Also, only one column named ‘fnlwgt’ from the dataset is dropped as this attribute does not have much scope to train the model and for prediction. After filtering the necessary attributes. The data was scaled properly using the Normalize operator and Compare ROC operator was used to analyse and choose the best models are better to choose for the selected dataset. In pipeline we have used Gradient Boosted Tree, Random Forest, Logistic Regression and Decision Tree. All the models were carried out by using Optimize parameters. In terms of AUC, ‘Random Forest’ and ‘Gradient Boosted Tree’ models are performing better as it ensures better classification performance compared to Logistic Regression and Decision Tree models, shown in *figure7.* Based on this outcome, the hyperparameter optimization chosen for each model and performed on 80% of training data to find the best set of parameters using ‘grid search’. These models with optimized parameters are then applied on test data (20%) to do the prediction and later performance was evaluated and compared.

Considering all appropriate models evaluated on manual pipeline, The Random Forest model provides the best Recall for the class ‘<=50k’ but might need some more improvements in its prediction for the other class >50k, shown in figure 8& 9*.* The model ‘Gradient Boosted Trees’ is the best overall model in terms of accuracy, precision, and recall, particularly for the >50K class. *[Figure: 10 & Figure:11]*. Parallelly ‘Logistic Regression’ gave better results, particularly for <=50K predictions, and offers a good balance between precision and recall. *[Figure: 12 & Figure 13]*. The Decision Tree model is doing well in classifying the <=50K class (high recall), but it is not performing as well on the >50K class (low recall). *[Figure: 14 & Figure: 15]*

* Best Accuracy: Gradient Boosted Trees (86.71%)
* Best Recall for <=50K: Random Forest (96.63%)
* Best Recall for >50K: Gradient Boosted Trees (75.91%)
* Best Precision for <=50K: Logistic Regression (91.29%)
* Best Precision for >50K: Gradient Boosted Trees (92.38%)

Overall, the manual pipeline implementation provides the better option for extensive data preprocessing and parameter optimization, but the main drawback is the execution time. Manual RapidMiner takes comparatively more processing time than Rapid miner AutoML and DataRobot Auto pilot process.

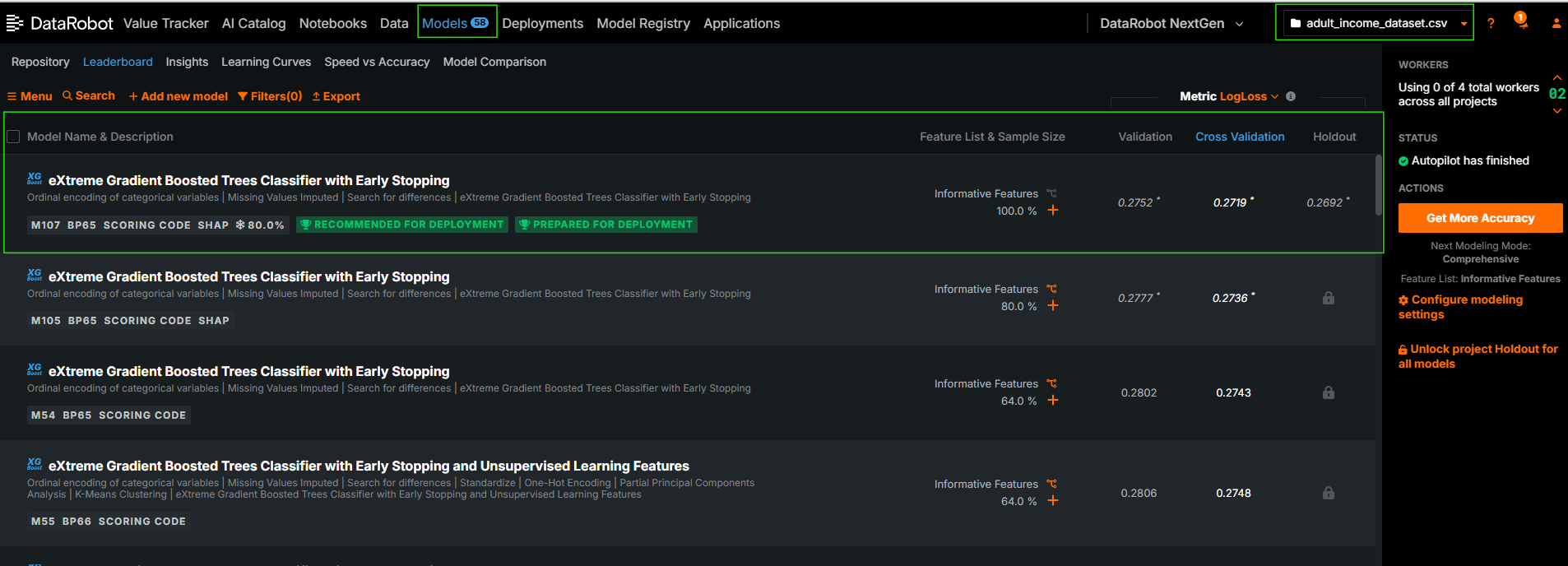
DataRobot serves as an excellent starting point by identifying potential target inconsistencies or issues within your dataset and offering a wide range of machine learning models for evaluation. Coding in Python is extremely effective for performing exploratory data analysis, enabling a deeper understanding of data patterns and relationships. RapidMiner AutoML provides a great platform to test and compare different machine learning models, making it easier to assess their accuracy. Additionally, building a custom pipeline using the Optimise Parameters operator is an efficient way to fine-tune and enhance model performance.

## **Appendix**

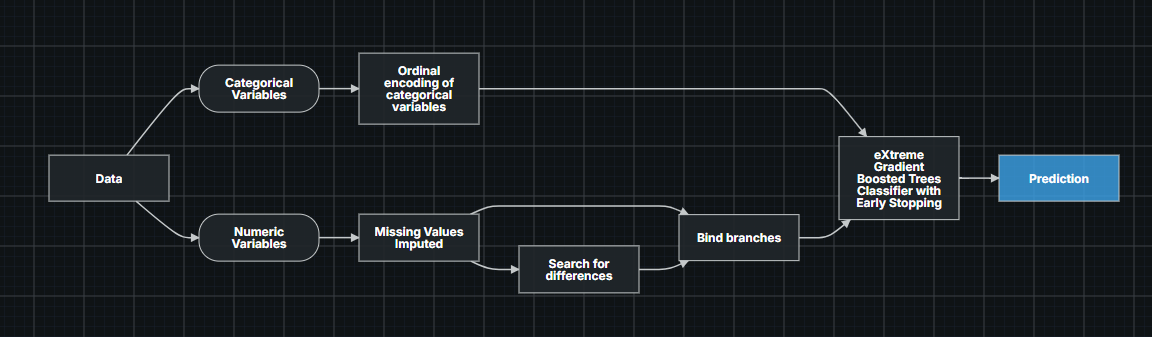
Kaggle Dataset: [Adult income dataset](https://www.kaggle.com/datasets/wenruliu/adult-income-dataset)

Kaggle Notebook: [Adult Income💵 | EDA📊 | ML🤖](https://www.kaggle.com/code/minaremon39/adult-income-eda-ml)

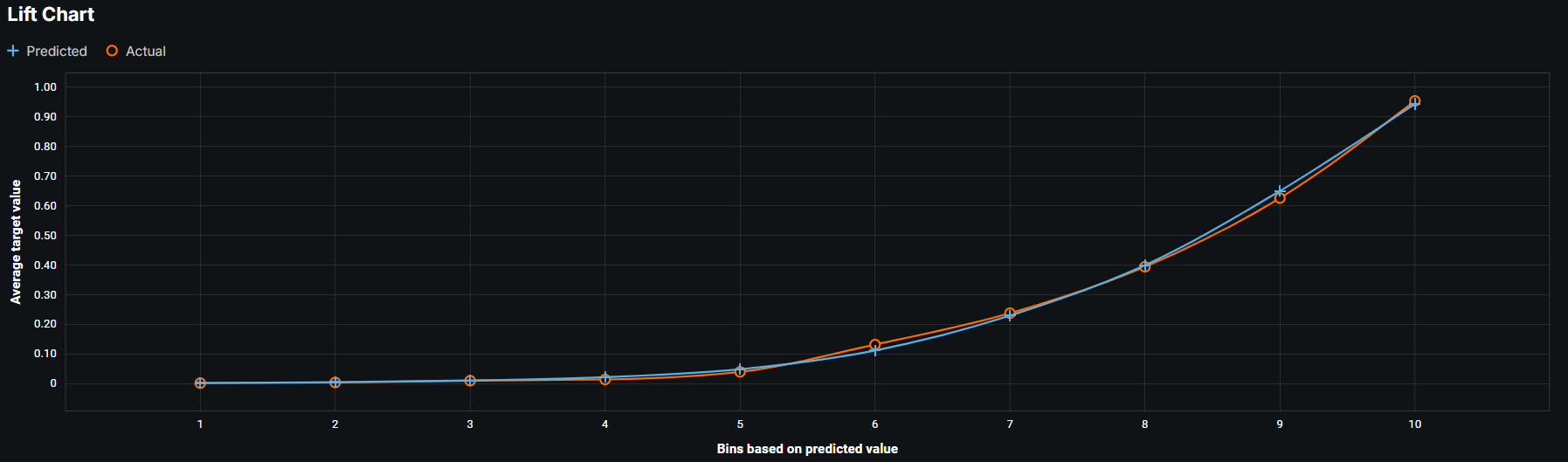
**DataRobot Figures:**



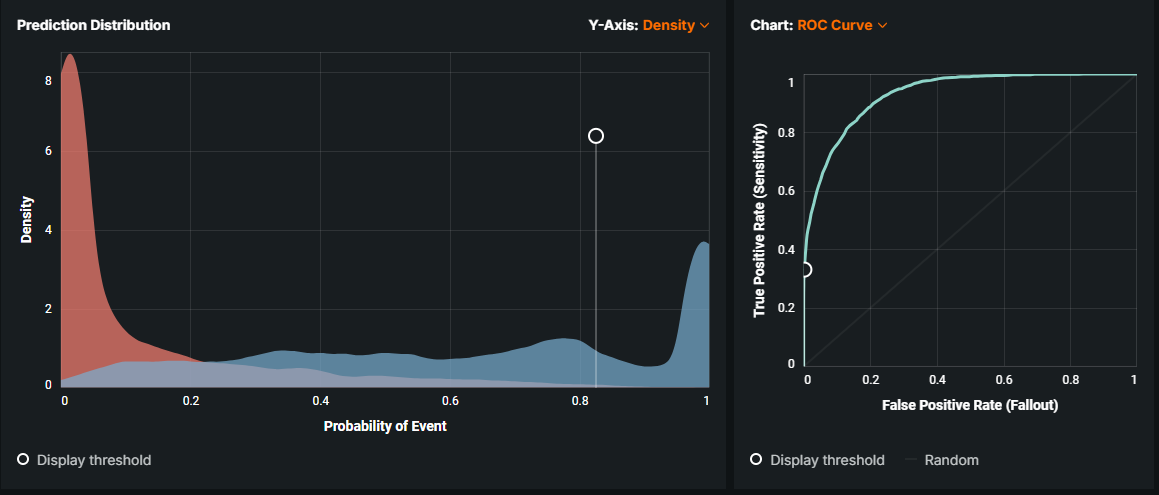
*Figure1: Leader Board*



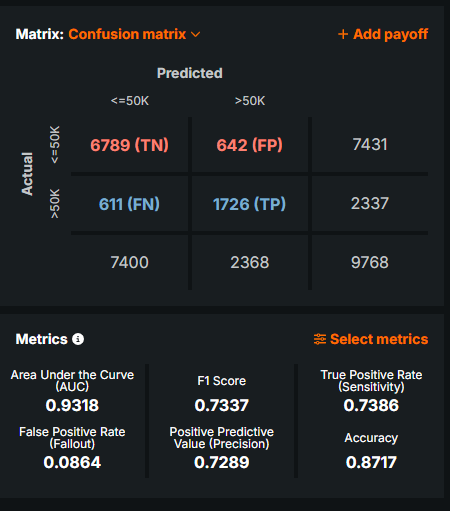
*Figure 2: Blueprint*



*Figure 3: Lift Chart*

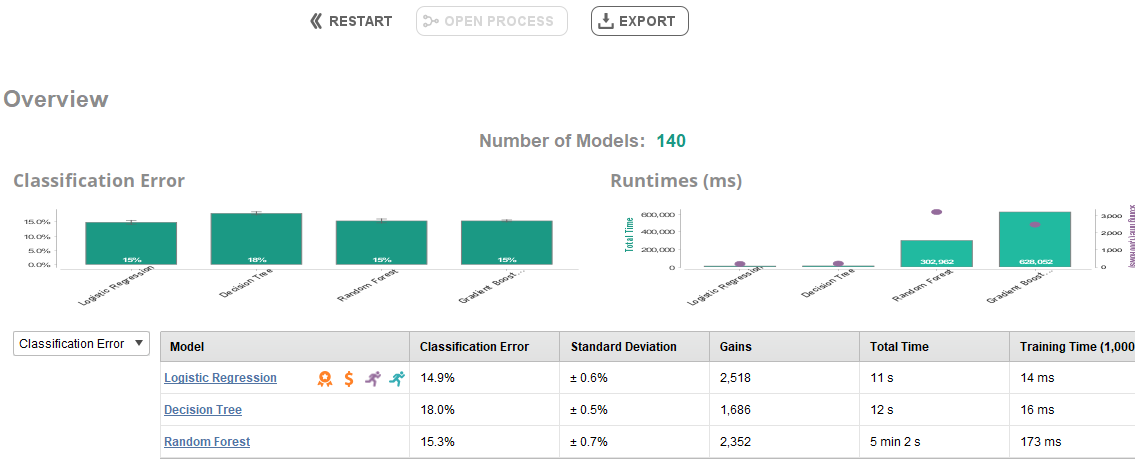


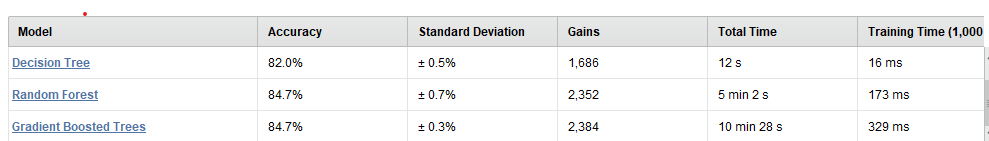
*Figure 4: Prediction Distribution & ROC Curve*



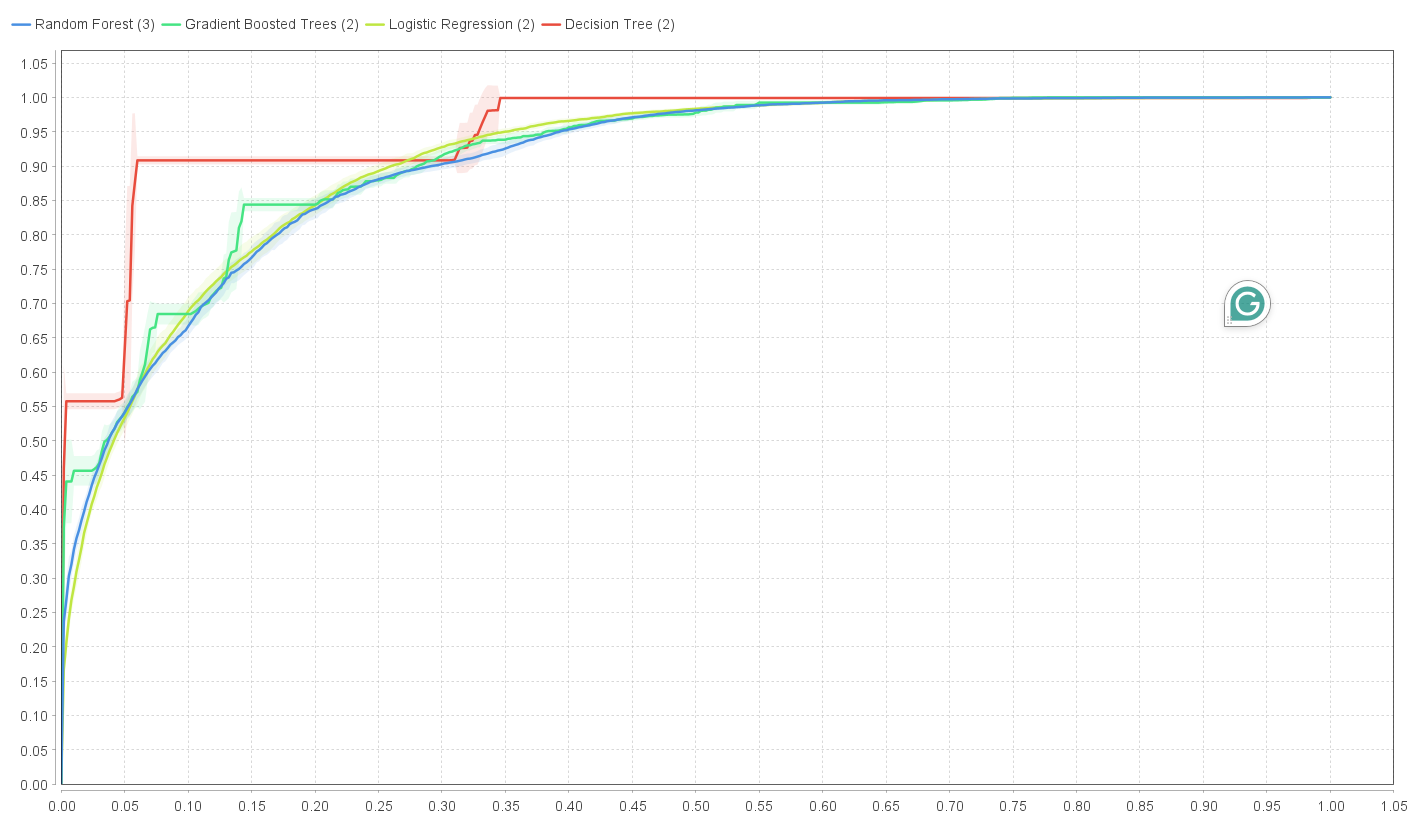
*Figure 5: Confusion Matrix*

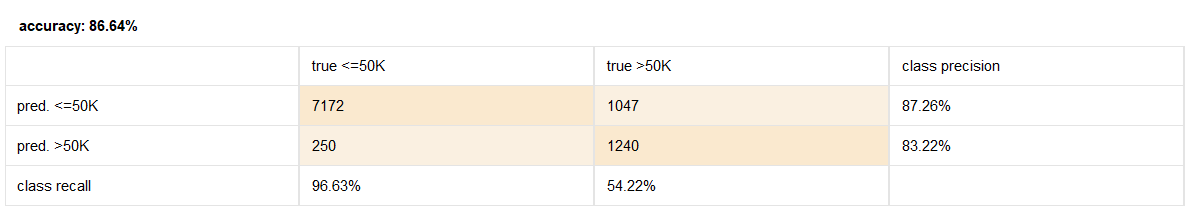
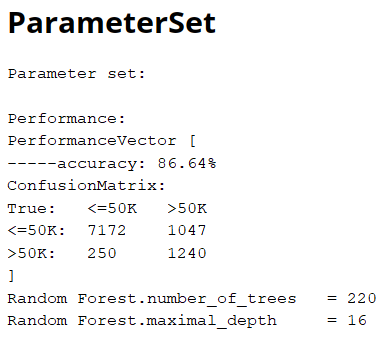
**AutoML Figures:**

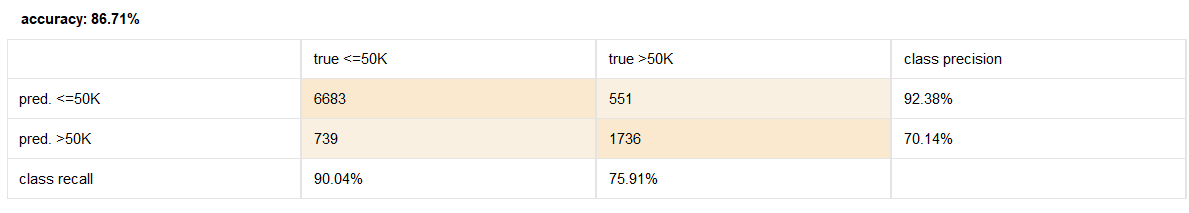


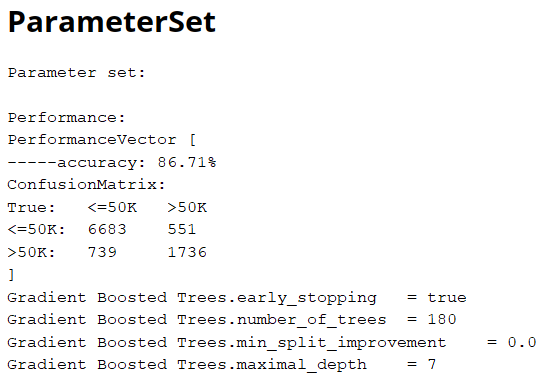
*Figure 6: Overview of RapidMiner AutoML Models*

**Manual RapidMiner Figures:**

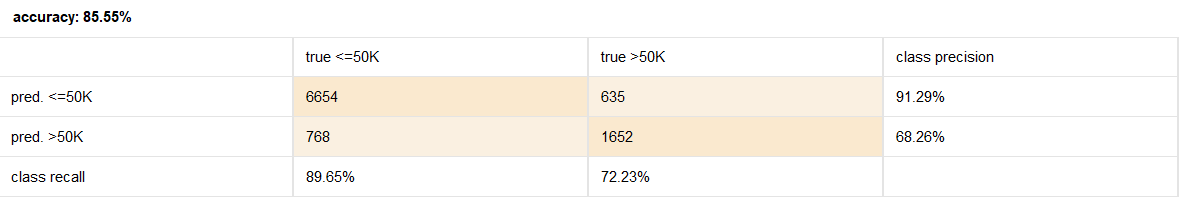
  
*Figure 7: ROC Curve*

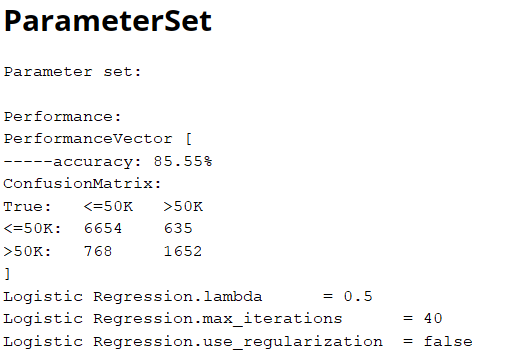
*Figure8:RandomForest*  
  
*Figure 9: Random Forest Parameter Set*

  
*Figure 10: Gradient boosted tree*

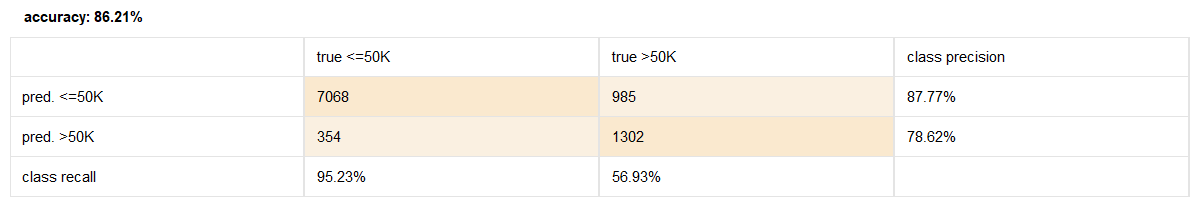


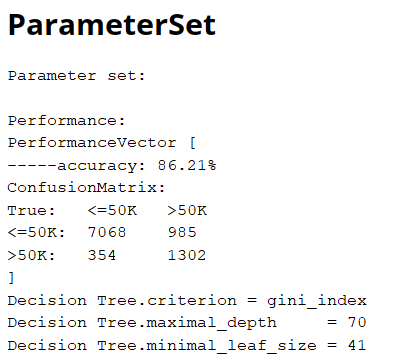
*Figure 11: Gradient boosted tree Parameter set*

  
*Figure 12: Logistic Regression*



*Figure 13: Logistic Regression Parameter set*

  
*Figure 14: Decision Tree*



*Figure 15: Decision Tree Parameter set*