Creditworthiness for a Loan

Step 1: Business and Data Understanding

What decisions need to be made?
Suddenly increasing the application for the loan created a demand to build a classification model to predict the applied application creditworthiness to give the loan.

What data is needed to inform those decisions?
Past application data with applicants' details.
Credit-data-training dataset helps build and validate the model
Customers-to-score is the new application to predict creditworthiness.

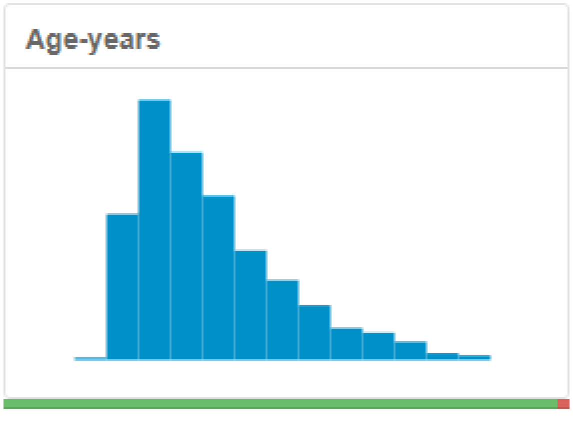
What kind of model (Continuous, Binary, Non-Binary, Time-Series) do we need to use to help make these decisions?
We are going to predict two values so Binary Model

Step 2: Explore and Cleanup the Data

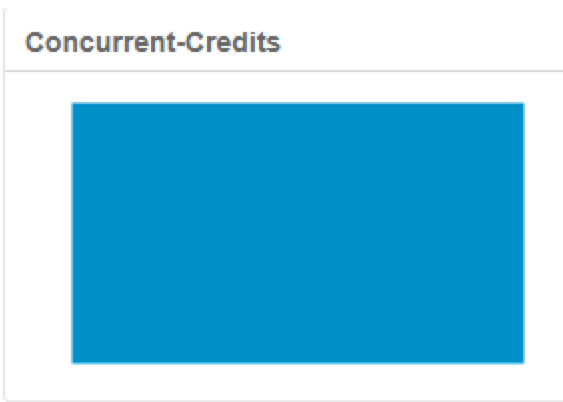
Build your training set given the data provided to you. The data has been cleaned up for you already so you shouldn’t need to convert any data fields to the appropriate data types.

In order to achieve the best predicting model, we have a great data set. When building the dataset I decide to remove some fields and impute one field. Check below.

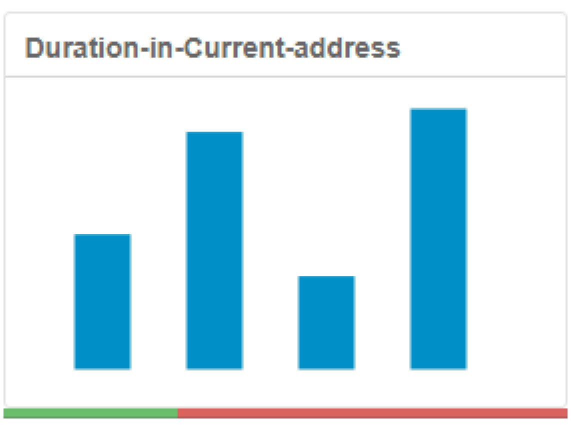
Age-years - This Field only has 2% null values. So I decided to impute data.



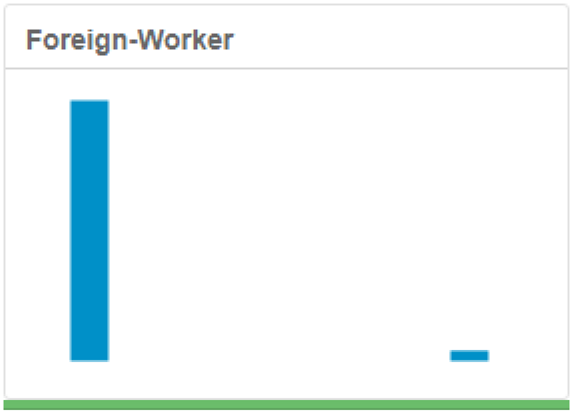
Concurrent-credits - This data field looks very uniform. If I choose to build the model it will skew the model.



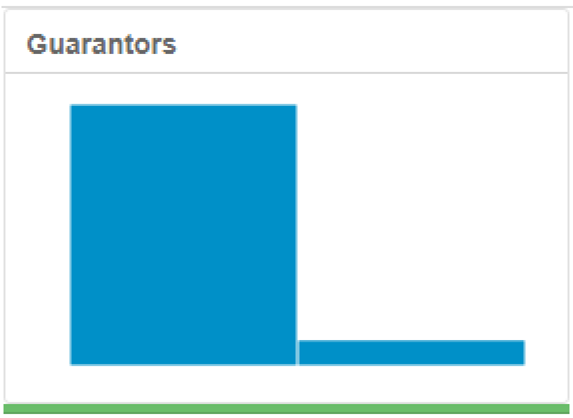
Duration-in-Current-Address - It has lots of null values. The best thing to do is remove it.



Foreign-Workers - This data field looks very uniform. If I choose to build the model it will skew the model.



Guarantors - This data field looks very uniform. If I choose to build the model it will skew the model.

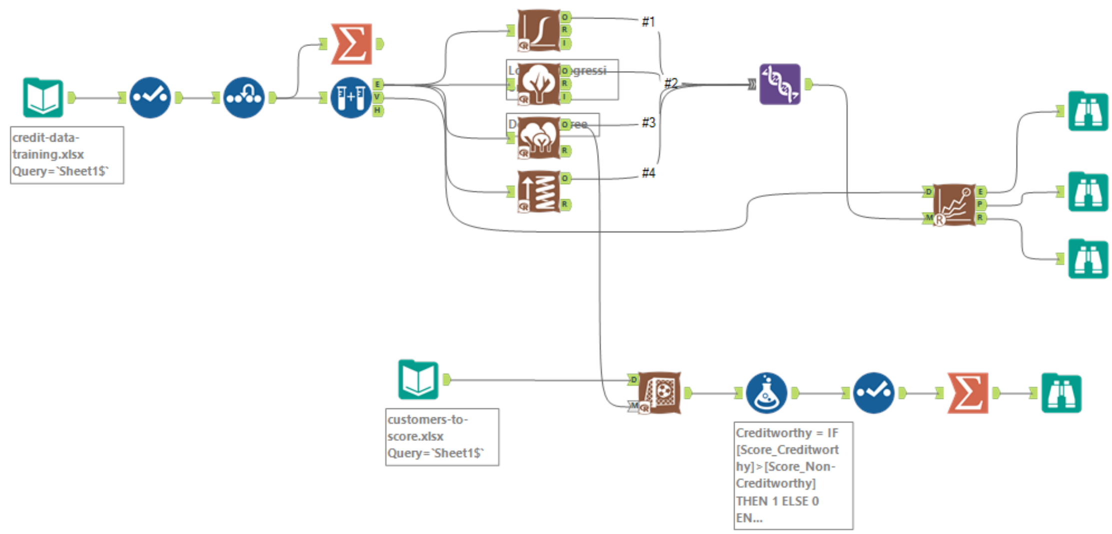


I got rounded up at age 36 and 13 columns after cleaning up the dataset.

Step 3. Train your Classification Models

Create all of the following models: Logistic Regression, Decision Tree, Forest Model, Boosted Model

Alteryx Workflow



Logistic regression Model

Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.118	-0.712	-0.428	0.721	2.618

Coefficients:

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-3.3658428	1.178e+00	-2.85816	0.00426 **
Account.BalanceSome Balance	-1.5769536	3.263e-01	-4.83298	1.34e-06 ***
Duration.of.Credit.Month	0.0078332	1.376e-02	0.56934	0.56912
Payment.Status.of.Previous.CreditPaid Up	0.4248873	3.857e-01	1.10151	0.27067
Payment.Status.of.Previous.CreditSome Problems	1.3099054	5.350e-01	2.44863	0.01434 *
PurposeNew car	-1.7413765	6.283e-01	-2.77171	0.00558 **
PurposeOther	-0.2316233	8.383e-01	-0.27629	0.78232
PurposeUsed car	-0.7987442	4.152e-01	-1.92361	0.0544 .
Credit.Amount	0.0001534	7.118e-05	2.15479	0.03118 *
Value.Savings.StocksNone	0.6267465	5.115e-01	1.22528	0.22047
Value.Savings.Stocks£100-£1000	0.1628575	5.673e-01	0.28710	0.77404
Length.of.current.employment4-7 yrs	0.5277541	4.918e-01	1.07303	0.28326
Length.of.current.employment< 1yr	0.8054343	3.953e-01	2.03752	0.0416 *
Instalment.per.cent	0.3005638	1.422e-01	2.11366	0.03454 *
Most.valuable.available.asset	0.3057238	1.568e-01	1.94952	0.05123 .
Age.years	-0.0159822	1.555e-02	-1.02768	0.3041
Type.of.apartment	-0.2505608	2.954e-01	-0.84827	0.39629
No.of.Credits.at.this.BankMore than 1	0.3607280	3.830e-01	0.94192	0.34623
No.of.dependents	-0.0247027	4.346e-01	-0.05684	0.95467
Telephone	0.3722599	3.151e-01	1.18131	0.23748

There are 2 variables that are Statistically Significant. There are ***Account.Balancesome Balance, Purposenew Car***.

Decision Tree Model

```
Call:
rpart(formula = Credit.Application.Result ~ Account.Balance +     Duration.of.Credit.Month + Payment.Status.of.Previous.Credit +     Purpose + Credit.Amount + Value.Savings.Stocks + Length.of.current.employment +     Instalment.per.cent + Most.valuable.available.asset + Age.years +     Type.of.apartment + No.of.Credits.at.this.Bank + No.of.dependents +     Telephone, data = the.data, minsplit = 20, minbucket = 7,     xval = 10, maxdepth = 20, cp = 1e-05, usesurrogate = 0, surrogatestyle = 0)
```

Model Summary

```
Variables actually used in tree construction:
[1] Account.Balance Duration.of.Credit.Month Purpose Value.Savings.Stocks
Root node error: 97/350 = 0.27714
n= 350
```

Pruning Table

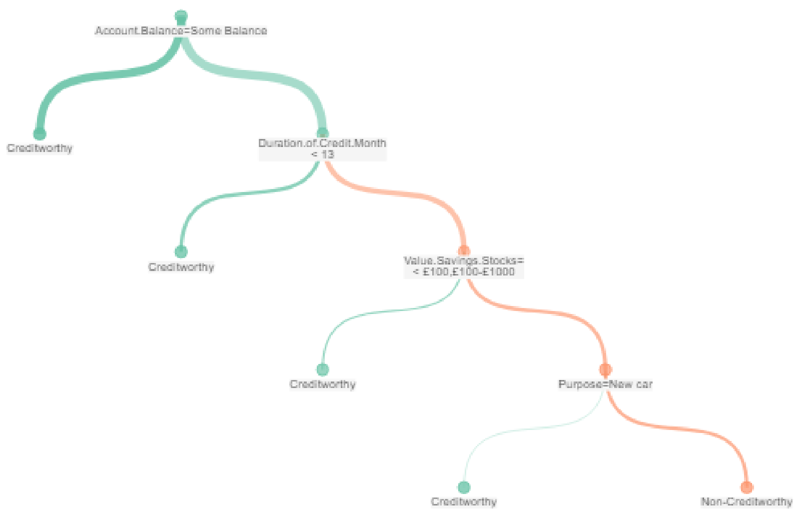
Level	CP	Num Splits	Rel Error	X Error	X Std Dev
1	0.068729	0	1.00000	1.00000	0.086326
2	0.041237	3	0.79381	0.94845	0.084898
3	0.025773	4	0.75258	0.88660	0.083032

Leaf Summary

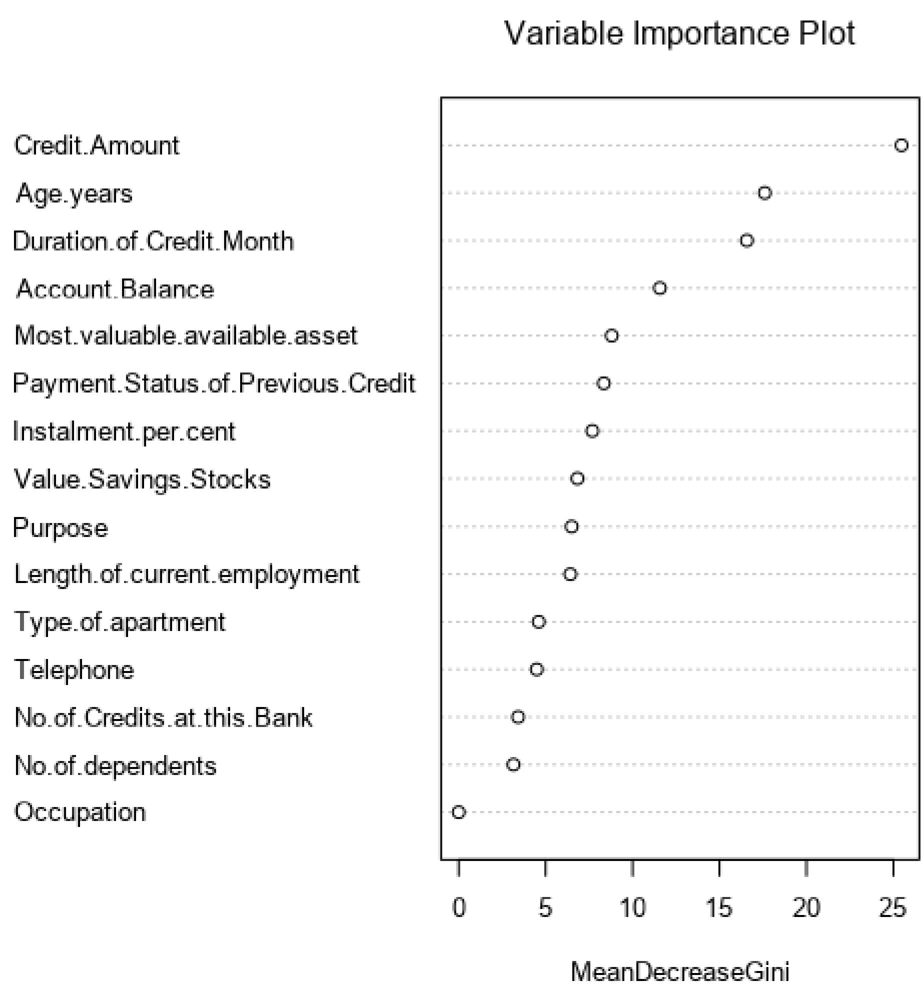
```
node), split, n, loss, yval, (yprob)
      * denotes terminal node
 1) root 350 97 Creditworthy (0.7228571 0.2771429)
   2) Account.Balance=Some Balance 166 20 Creditworthy (0.8795181 0.1204819) *
   3) Account.Balance=No Account 184 77 Creditworthy (0.5815217 0.4184783)
     6) Duration.of.Credit.Month< 13 74 18 Creditworthy (0.7567568 0.2432432) *
     7) Duration.of.Credit.Month>=13 110 51 Non-Creditworthy (0.4636364 0.5363636)
      14) Value.Savings.Stocks=< £100,£100-£1000 34 11 Creditworthy (0.6764706 0.3235294) *
      15) Value.Savings.Stocks=None 76 28 Non-Creditworthy (0.3684211 0.6315789)
        30) Purpose=New car 8  2 Creditworthy (0.7500000 0.2500000) *
        31) Purpose=Home Related,Other,Used car 68 22 Non-Creditworthy (0.3235294 0.6764706) *
```

The Important Predictor Variables are ***Account.Balancesome Balance, Value.Sainfs.tocks,Purposenew Car.*** This model has an overall 73% accuracy in predicting creditworthy applications.

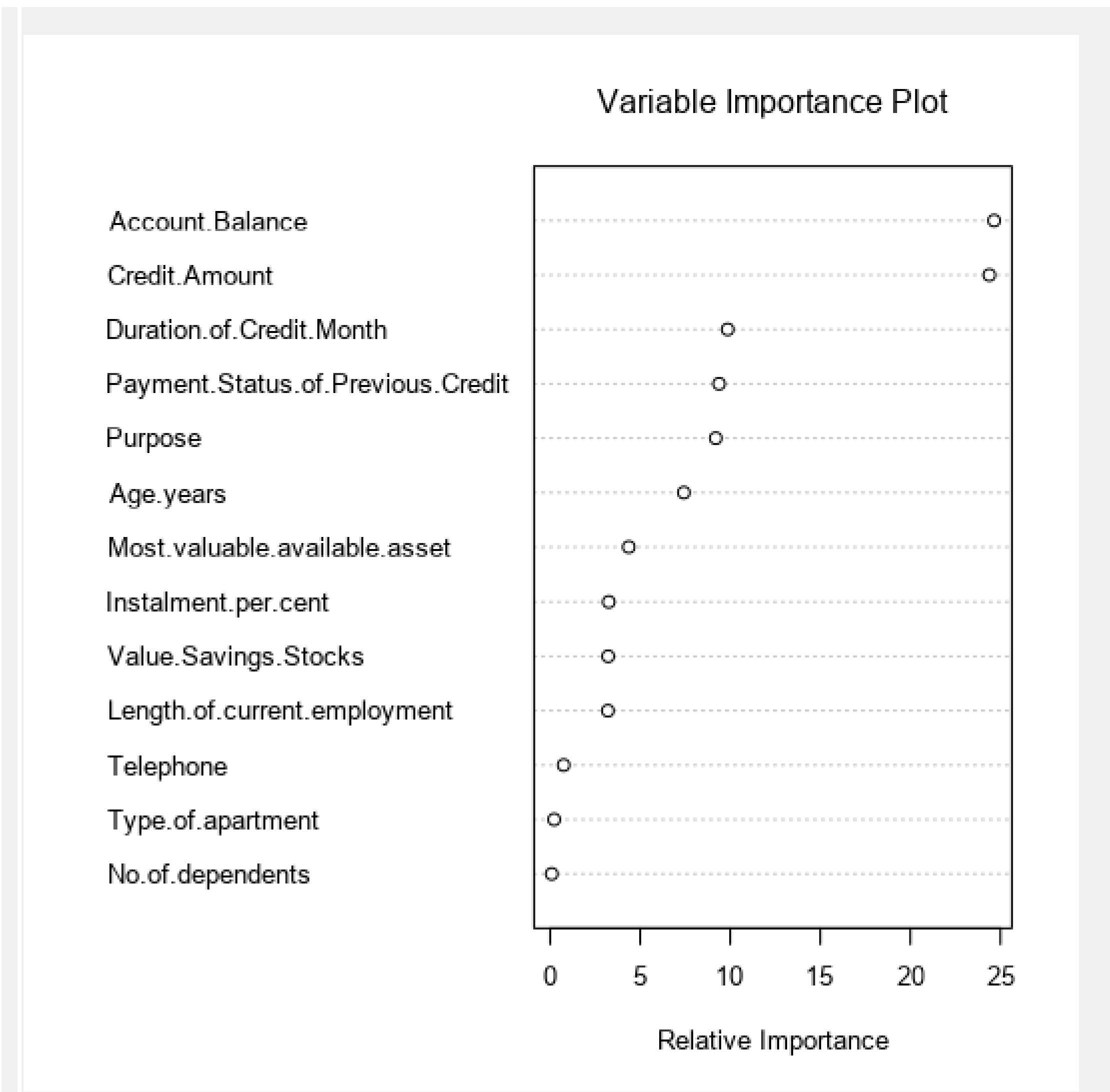
Decision Tree Graph



Forest model variable Importance Plot - It seems ***Credit.Amount, Age.years, Duration.of.Credit.Month*** *is a more important variable.*



Boosted Model Variable Importance Plot - It seems ***Account.Balance , Credit. Amount*** is an essential variable.



Step 4. Writeup

Write a brief report on how you came up with your classification model and write down how many of the new customers would qualify for a loan.

Compare all of the models' performances against each other. Decide on the best model and score your new customers.

Model Comparison Report

Model Comparison Report

Fit and error measures

Model	Accuracy	F1	AUC	Accuracy_Creditworthy	Accuracy_Non-Creditworthy
Logistic_Regression	0.7867	0.8559	0.7149	0.9048	0.5111
Decision_Tree	0.7467	0.8304	0.7035	0.8857	0.4222
Forest_Model	0.8067	0.8766	0.7576	0.9810	0.4000
Boosted_Model	0.7933	0.8670	0.7450	0.9619	0.4000

Model: model names in the current comparison.
Accuracy: overall accuracy, number of correct predictions of all classes divided by total sample number.
Accuracy_[class name]: accuracy of Class [class name] is defined as the number of cases that are **correctly** predicted to be Class [class name] divided by the total number of cases that actually belong to Class [class name], this measure is also known as *recall*.
AUC: area under the ROC curve, only available for two-class classification.
F1: F1 score, 2 * precision * recall / (precision + recall). The *precision* measure is the percentage of actual members of a class that were predicted to be in that class divided by the total number of cases predicted to be in that class. In situations where there are three or more classes, average precision and average recall values across classes are used to calculate the F1 score.

The Forest Model has greater accuracy in predicting creditworthy applications compared to other models. The second-best model is the Boosted model.

Confusion Matrix

Confusion matrix of Boosted_Model

	Actual_Creditworthy	Actual_Non-Creditworthy
Predicted_Creditworthy	101	27
Predicted_Non-Creditworthy	4	18

Confusion matrix of Decision_Tree

	Actual_Creditworthy	Actual_Non-Creditworthy
Predicted_Creditworthy	93	26
Predicted_Non-Creditworthy	12	19

Confusion matrix of Forest_Model

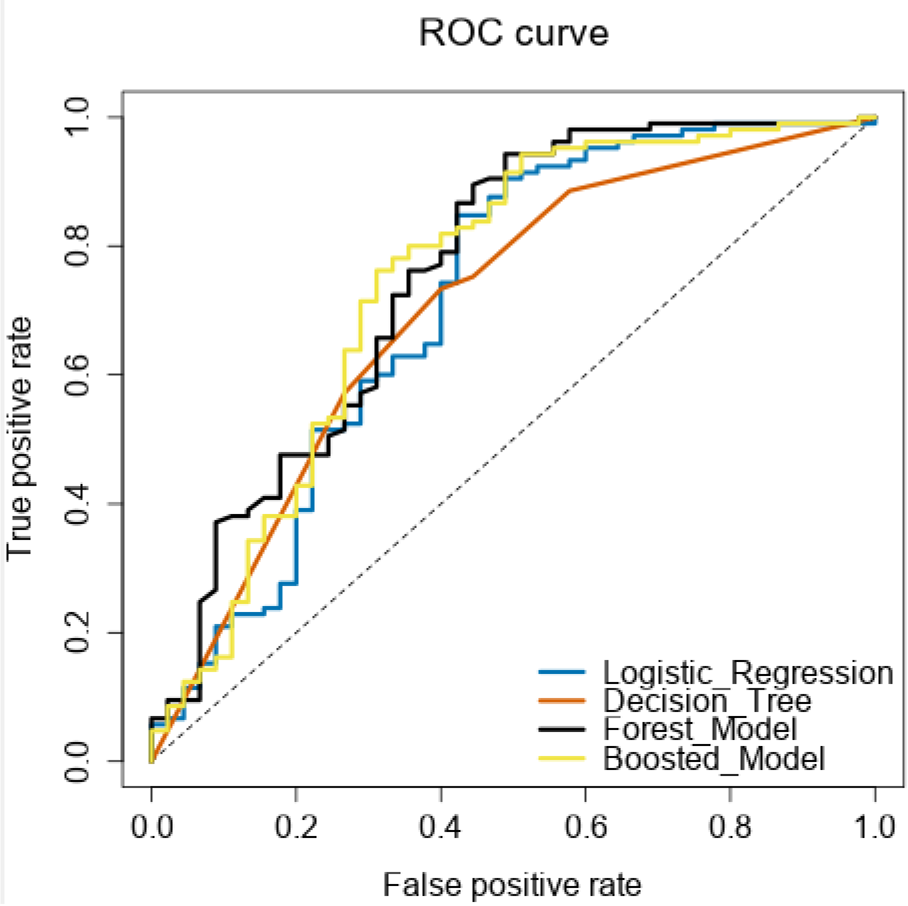
	Actual_Creditworthy	Actual_Non-Creditworthy
Predicted_Creditworthy	103	27
Predicted_Non-Creditworthy	2	18

Confusion matrix of Logistic_Regression

	Actual_Creditworthy	Actual_Non-Creditworthy
Predicted_Creditworthy	95	22
Predicted_Non-Creditworthy	10	23

Confusion Matrix shows clearly forest models perform better at validation samples.

ROC Curve



Logistic regression - It predicts more non-creditworthy applications but they are creditworthy and seem biased.

Decision Tree - It predicts more creditworthy applications but they are non-creditworthy and it seems biased.

The Forest model looks good at predicting creditworthy applications and it seems not biased.

The booster model is also good at predicting creditworthy applications but comparably it seems low with the forest model.

After comparing and seeing all the charts and numbers the forest model predicted many creditworthy applications without biases so I decided to use the forest model to predict the new applications.

After applying the forest model to the new application it predicted 419 applications for creditworthiness.

Alteryx Workflow