BANK MARKETING PREDICTION

Pushkar Bharambe
Analytics: Prin & Appl
ANLY 500-91-O-2018/Late Summer

Project Summary

Marketing selling campaigns constitute a typical strategy to enhance business in the banking world. Banks use direct marketing when targeting segments of customers by contacting them to meet a specific goal. The objective of this project is to predict whether a client will subscribe as a result of marketing efforts. The data is related to direct marketing campaigns of a Portuguese banking institution. The data contains information like age of the client, job status – whether the client is employed or not, marital status, level of education, whether the client has ever defaulted on his loans, account balance, whether the client currently holds a loan or not, preferred method of contact and information related to previous marketing outreaches to the client.

To predict the client's decision, I intend to create a model using Alteryx. We are data rich, meaning that we can historical data to build a model and since the model output is expected to be 'yes' or 'no' we need to build a binary model. I intend to explore classification algorithms like logistic regression, decision trees, random forest model and boosted model to predict the outcome. After building four binary classification models, I will evaluate them based on their characteristics and choose the best fit. By the end of the project, I expect to have a binary classification model which can be used on the test data set to predict whether a client will subscribe.

Project Methodology

The entire project will be implemented in a tool called **Alteryx**. The data set will be loaded in the tool and explored for missing values, low variability, and association analysis will be done with respect to the target and amongst the variables themselves. If need be data will be appropriately formatted or cleansed before going forward. The next step will be to partition data into sample data (70% of total data) and validation data (30%). The sample data will be then used to build classification models (logistic regression, decision trees, random forest and boosted). Models will be compared considering accuracy of the entire models, accuracy of 'yes' prediction, accuracy of 'no' prediction and the ROC curve.

Project Timeline:

Week 1 – 2: Select data-set and decide tools to use.

Week 3 – 5: Explore data, find inconsistencies, do association analysis, filtering and splitting data

Week 6 – 8: Build at least one model (logistic regression) in Alteryx. Submit project proposal

Week 7 – 9: Build other models and fine tune them.

Week 10 – 12: Compare the models and select the best fit model. Submit project status sample presentation.

Week 13 – 15: Wrap up the documentation and submit the final project.

Data Set Information

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

Attribute Information:

Input variables:

bank client data:

1 - age (numeric)
2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown')
5 - default: has credit in default? (categorical: 'no','yes','unknown')
6 - housing: has housing loan? (categorical: 'no','yes','unknown')
7 - loan: has personal loan? (categorical: 'no','yes','unknown')

related with the last contact of the current campaign:

8 - contact: contact communication type (categorical: 'cellular','telephone')
9 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
11 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.
#other attributes: 12 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
13 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
14 - previous: number of contacts performed before this campaign and for this client (numeric)
15 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

social and economic context attributes

16 - emp.var.rate: employment variation rate - quarterly indicator (numeric)
17 - cons.price.idx: consumer price index - monthly indicator (numeric)
18 - cons.conf.idx: consumer confidence index - monthly indicator (numeric)
19 - euribor3m: euribor 3 month rate - daily indicator (numeric)
20 - nr.employed: number of employees - quarterly indicator (numeric)

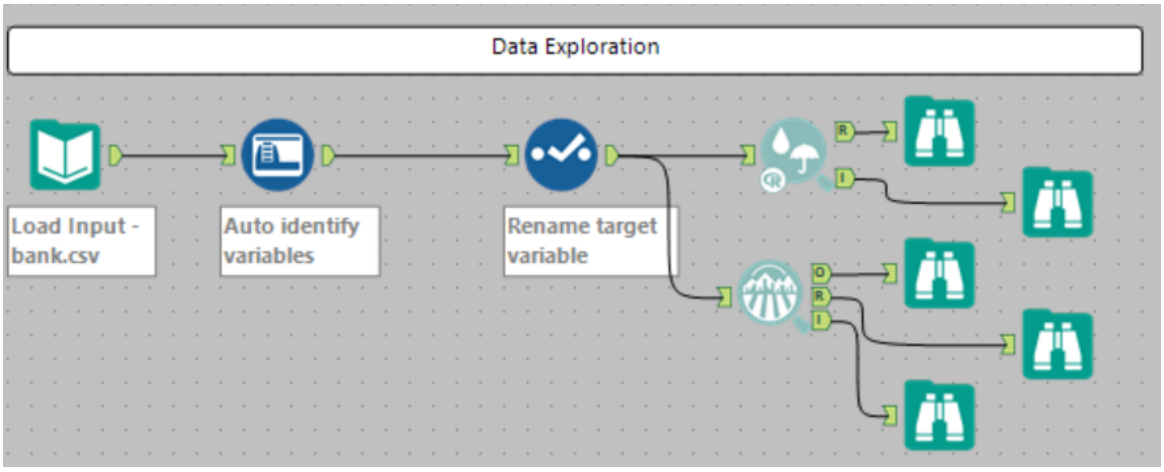
Output variable (desired target):
21 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Technical Approach

To analyze the dataset, we will do the following:

1. Load & explore data (improve readability, check structure, formatting, check for nulls, correlation plots, visualizing data)
2. Filtering data (split data into training & test data)
3. Building Model (Logistic Regression, Decision Trees, Random Forest, Boosted Model)
4. Model Comparison (choose the best model based on accuracy)
5. Model Validation (use test data for prediction and judge accuracy of the chosen model)
6. Scoring the model

Load & Explore Data



The data presented to us is in csv format. We use the 'Input tool' in Alteryx to load the data. Here is a view of the first five rows of data across all columns.

Record #	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration
1	30	unemployed	married	primary	no	1787	no	no	cellular	19	oct	79
2	33	services	married	secondary	no	4789	yes	yes	cellular	11	may	220
3	35	management	single	tertiary	no	1350	yes	no	cellular	16	apr	185
4	30	management	married	tertiary	no	1476	yes	yes	unknown	3	jun	199
5	59	blue-collar	married	secondary	no	0	yes	no	unknown	5	may	226

campaign	pdays	previous	poutcome	y
1	-1	0	unknown	no
1	339	4	failure	no
1	330	1	failure	no
4	-1	0	unknown	no
1	-1	0	unknown	no

The data loaded from a csv is all in string format. We will need to fix this. The 'Auto-field tool' identifies the correct data type for variables and changes them accordingly.

Auto Field (3)	The FieldType of "age" changed to: Byte
Auto Field (3)	The FieldType of "job" changed to: String(13)
Auto Field (3)	The FieldType of "marital" changed to: String(8)
Auto Field (3)	The FieldType of "education" changed to: String(9)
Auto Field (3)	The FieldType of "default" changed to: String(3)
Auto Field (3)	The FieldType of "balance" changed to: Int32
Auto Field (3)	The FieldType of "housing" changed to: String(3)
Auto Field (3)	The FieldType of "loan" changed to: String(3)

Auto Field (3)	The FieldType of "contact" changed to: String(9)
Auto Field (3)	The FieldType of "day" changed to: Byte
Auto Field (3)	The FieldType of "month" changed to: String(3)
Auto Field (3)	The FieldType of "duration" changed to: Int16
Auto Field (3)	The FieldType of "campaign" changed to: Byte
Auto Field (3)	The FieldType of "pdays" changed to: Int16
Auto Field (3)	The FieldType of "previous" changed to: Byte
Auto Field (3)	The FieldType of "poutcome" changed to: String(7)
Auto Field (3)	The FieldType of "y" changed to: String(3)

To improve readability, we change the name of our target variable 'y' to 'Target' using the 'Select Tool'.

	Field	Type	Size	Rename
☑	y	String	3	Target

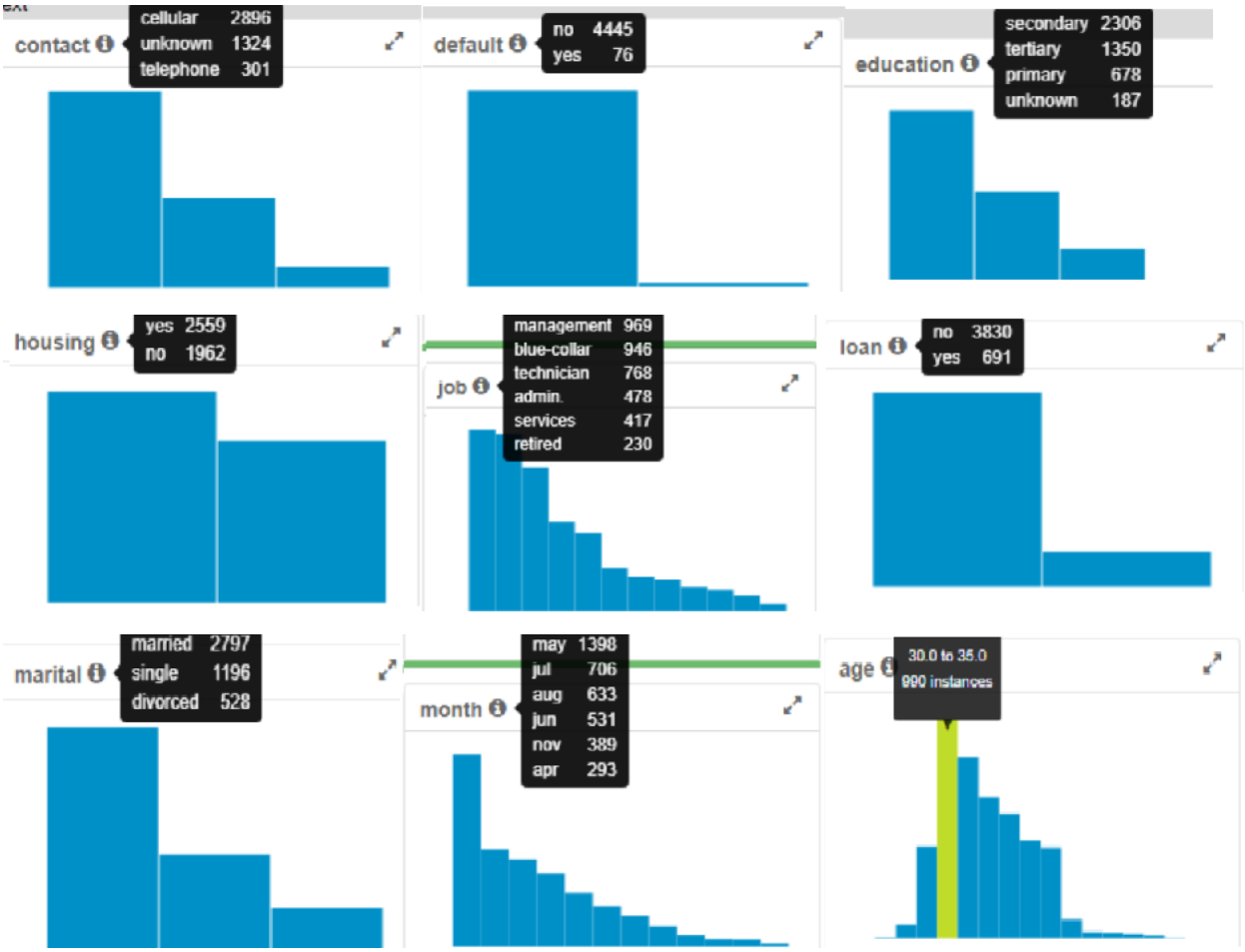
Now, let us see look for some characteristics of our data. We will use the 'Field Summary' tool for this.

Record #	Name	Field Category	Min	Max	Median	Std. Dev.	Percent Missing	Unique Values	Mean
1	age	Numeric	19	87	39	10.576211	0	67	41.170095
2	balance	Numeric	-3313	71188	444	3009.638142	0	2353	1422.657819
3	campaign	Numeric	1	50	2	3.109807	0	32	2.79363
4	day	Numeric	1	31	16	8.247667	0	31	15.915284
5	duration	Numeric	4	3025	185	259.856633	0	875	263.961292
6	pdays	Numeric	-1	871	-1	100.121124	0	292	39.766645
7	previous	Numeric	0	25	0	1.693562	0	24	0.542579
8	contact	String	[Null]	[Null]	[Null]	[Null]	0	3	[Null]
9	default	String	[Null]	[Null]	[Null]	[Null]	0	2	[Null]
10	education	String	[Null]	[Null]	[Null]	[Null]	0	4	[Null]
11	housing	String	[Null]	[Null]	[Null]	[Null]	0	2	[Null]
12	job	String	[Null]	[Null]	[Null]	[Null]	0	12	[Null]
13	loan	String	[Null]	[Null]	[Null]	[Null]	0	2	[Null]
14	marital	String	[Null]	[Null]	[Null]	[Null]	0	3	[Null]
15	month	String	[Null]	[Null]	[Null]	[Null]	0	12	[Null]
16	poutcome	String	[Null]	[Null]	[Null]	[Null]	0	4	[Null]

We observe the summary characteristics of the numerical variables. Its interesting to note that our customers vary from youngest at 19 yrs old to oldest at 87 yrs of age, average age of our customers' being 41.17 yrs. Since non-numeric variables do not have summary characteristics the fields appear as

well. We can also infer that there is no missing data in the dataset by looking at the Percent Missing column.

Next we visualize our data in barplots. Its interesting to see that most of the contacts are cellular, followd by customers who have not mentioned their contact numbers and a very small number of customers have given in their landline numbers. The amount of customers that have defaulted is very low compared to the ones who have not. Majority of the customers have completed their secondary education. 56.6% of the customers have housing. Highest percentage of clients have management jobs while a smile number of clients are retired. 84.72% of the customers do not have a loan currently. 61.87% of our customers are married, 26.45% single and the rest are divorced. Contact tells us when the customer was last contacted, it looks like a lot of customer outreach happens in May. Looking at the ages of our customers, it looks like 66.6% of customers lie between 30-45 years of age.



Next, we look at the correlation between the numeric variables.

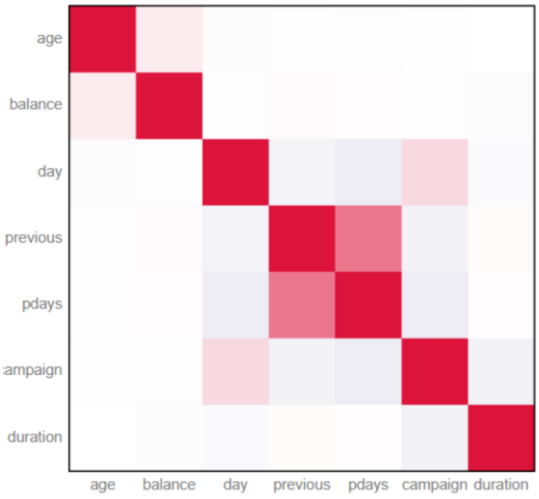
Pearson Correlation Analysis

Full Correlation Matrix

	age	balance	day	duration	campaign	pdays
age	1.0000000	0.0838201	-0.0178526	-0.0023669	-0.0051479	-0.0088935
balance	0.0838201	1.0000000	-0.0086771	-0.0159499	-0.0099762	0.0094367
day	-0.0178526	-0.0086771	1.0000000	-0.0246293	0.1607061	-0.0943515
duration	-0.0023669	-0.0159499	-0.0246293	1.0000000	-0.0683820	0.0103802
campaign	-0.0051479	-0.0099762	0.1607061	-0.0683820	1.0000000	-0.0931368
pdays	-0.0088935	0.0094367	-0.0943515	0.0103802	-0.0931368	1.0000000
previous	-0.0035109	0.0261964	-0.0591144	0.0180803	-0.0678326	0.5775618

	previous
age	-0.0035109
balance	0.0261964
day	-0.0591144
duration	0.0180803
campaign	-0.0678326
pdays	0.5775618
previous	1.0000000

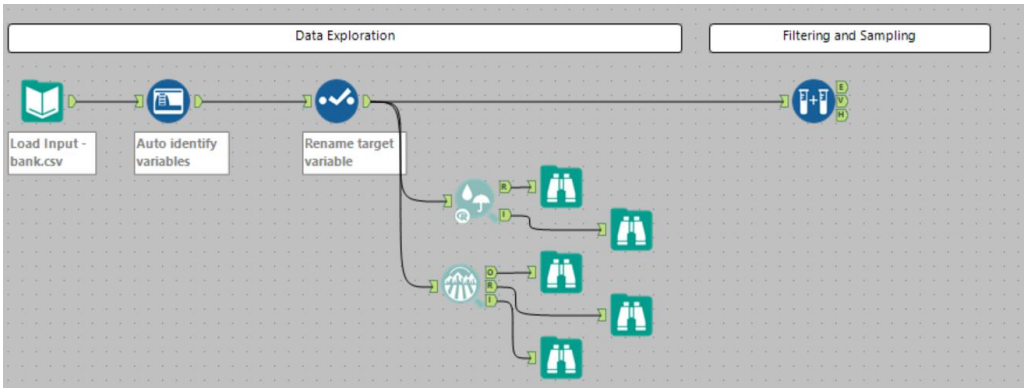
Correlation Matrix with ScatterPlot



No conclusive facts emerge from this analysis, except ‘previous’ and ‘pdays’ which makes sense because ‘pdays’ is the number of days that passed by after the client was last contacted from a previous campaign and ‘previous’ is the number of contacts performed before this campaign and for this client.

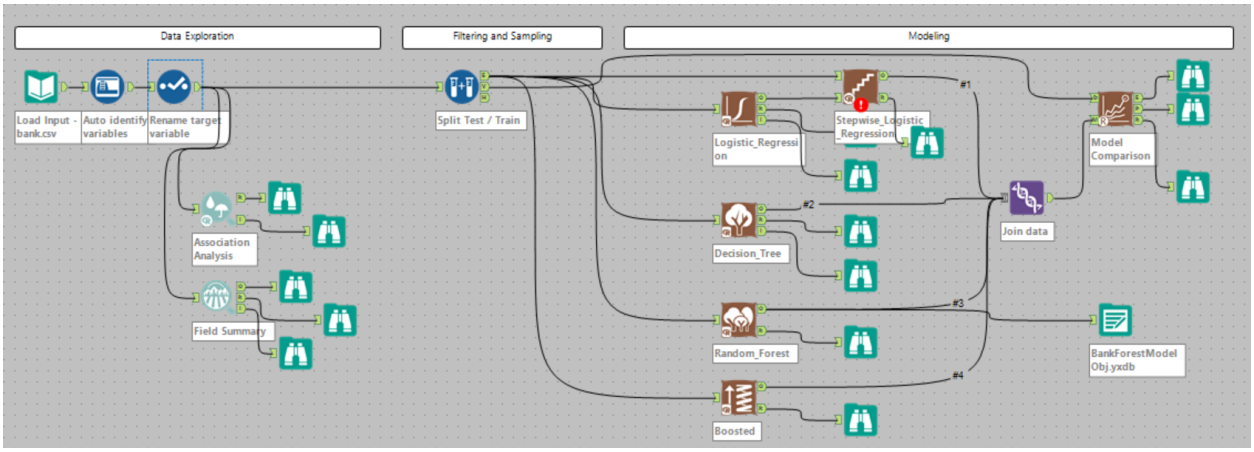
Filtering Data

We use the ‘Create Samples’ tool to split the data into 70% training data and 30% test data.



Building Model

Initially, I tried to use the multiple linear regression model but that clearly a wrong approach as the basic assumption of linear regression models is that the target variable is continuous, however our target variable is binary. Then I decided to go for logistic regression which is often used for classification.



Logistic Regression

Given a set of predictor variables, a logistic regression model allows a user to obtain the estimated probability for each of two possible responses for the target variable. I have added stepwise regression tool to the logistic regression tool. The Stepwise tool determines the best predictor variables to include in a model out of a larger set of potential predictor variables for linear, logistic, and other traditional regression models.

glm(formula = Target ~ job + marital + education + default + loan + contact + month + duration + campaign + poutcome, family = binomial("logit"), data = the.data)

Classification Matrix for the training data:

Actual	Actual Positive	Actual Negative
Predicted Positive	283 (38.1%)	460 (61.9%)
Predicted Negative	66 (2.73%)	2356 (97.3%)

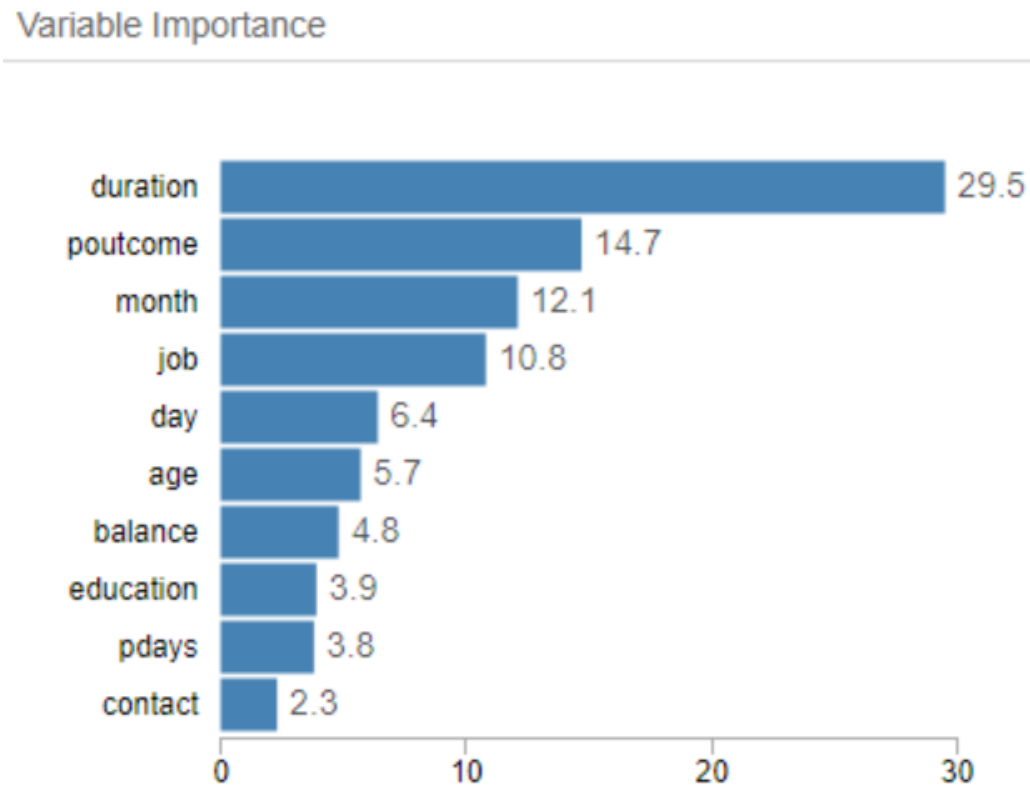
Significance of the variables is as follows:

	Estimate	Std. Error	z value	Pr(>|z|)	
(Intercept)	-2.229492	0.4362789	-5.1102	3.21e-07	***
jobblue-collar	-0.516764	0.2869409	-1.8009	0.07171	.
jobentrepreneur	-0.303119	0.4729571	-0.6409	0.52159	
jobhousemaid	-0.415910	0.4880112	-0.8523	0.39407	
jobmanagement	-0.177623	0.2962238	-0.5996	0.54876	
jobretired	0.494728	0.3295334	1.5013	0.13328	
jobself-employed	-0.518534	0.4794362	-1.0815	0.27945	
jobservices	-0.274387	0.3335358	-0.8227	0.4107	
jobstudent	0.894641	0.4387546	2.0390	0.04145	*
jobtechnician	-0.507289	0.2875293	-1.7643	0.07768	.
jobunemployed	-0.242786	0.4454751	-0.5450	0.58575	
jobunknown	0.666742	0.6645287	1.0033	0.3157	
maritalmarried	-0.427193	0.2046299	-2.0876	0.03683	*
maritalsingle	-0.342394	0.2309369	-1.4826	0.13817	
educationsecondary	0.076854	0.2336380	0.3289	0.7422	
educationtertiary	0.239402	0.2719555	0.8803	0.3787	
educationunknown	-0.796361	0.4548556	-1.7508	0.07998	.
defaultyes	0.841545	0.5078271	1.6571	0.09749	.
loanyes	-0.427626	0.2293119	-1.8648	0.06221	.
contacttelephone	0.082891	0.2652027	0.3126	0.75462	
contactunknown	-1.287968	0.2790060	-4.6163	3.90e-06	***
monthaug	-0.421525	0.2811144	-1.4995	0.13375	
monthdec	0.359510	0.7011224	0.5128	0.60812	
monthfeb	-0.229462	0.3290716	-0.6973	0.48562	
monthjan	-0.773546	0.4220021	-1.8330	0.0668	.
monthjul	-0.911123	0.2961711	-3.0763	0.0021	**
monthjun	0.148146	0.3473908	0.4265	0.66978	
monthmar	1.484358	0.4835199	3.0699	0.00214	**
monthmay	-0.944062	0.2739469	-3.4461	0.00057	***
monthnov	-0.945839	0.3166751	-2.9868	0.00282	**
monthoct	1.297064	0.3901098	3.3249	0.00088	***
monthsep	0.200418	0.4938777	0.4058	0.68489	
duration	0.004215	0.0002463	17.1123	< 2.2e-16	***
campaign	-0.085349	0.0351236	-2.4300	0.0151	*
poutcomeother	0.546475	0.3317312	1.6473	0.09949	.
poutcomesuccess	2.317745	0.3281188	7.0637	1.62e-12	***
poutcomeunknown	-0.099045	0.2213776	-0.4474	0.65459	

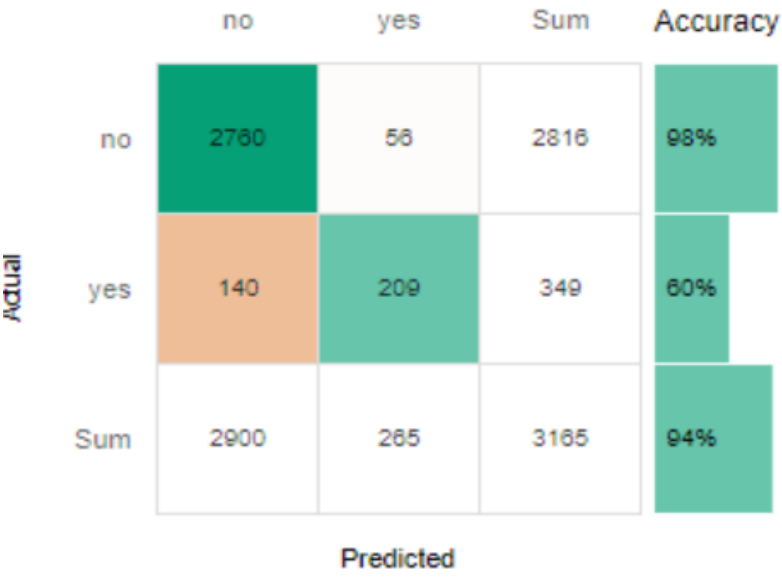
Significance codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Decision Tree model: The Decision Tree tool constructs a set of if-then split rules that optimize a criteria to create a model that predicts a target variable using one or more predictor variables. The criteria used to form these rules depends on the nature of the target variable. If the target variable identifies membership in one of a set of categories, then a classification tree is constructed. I played around the data for a bit and then customized the model to have- minimum 20 records to do a split

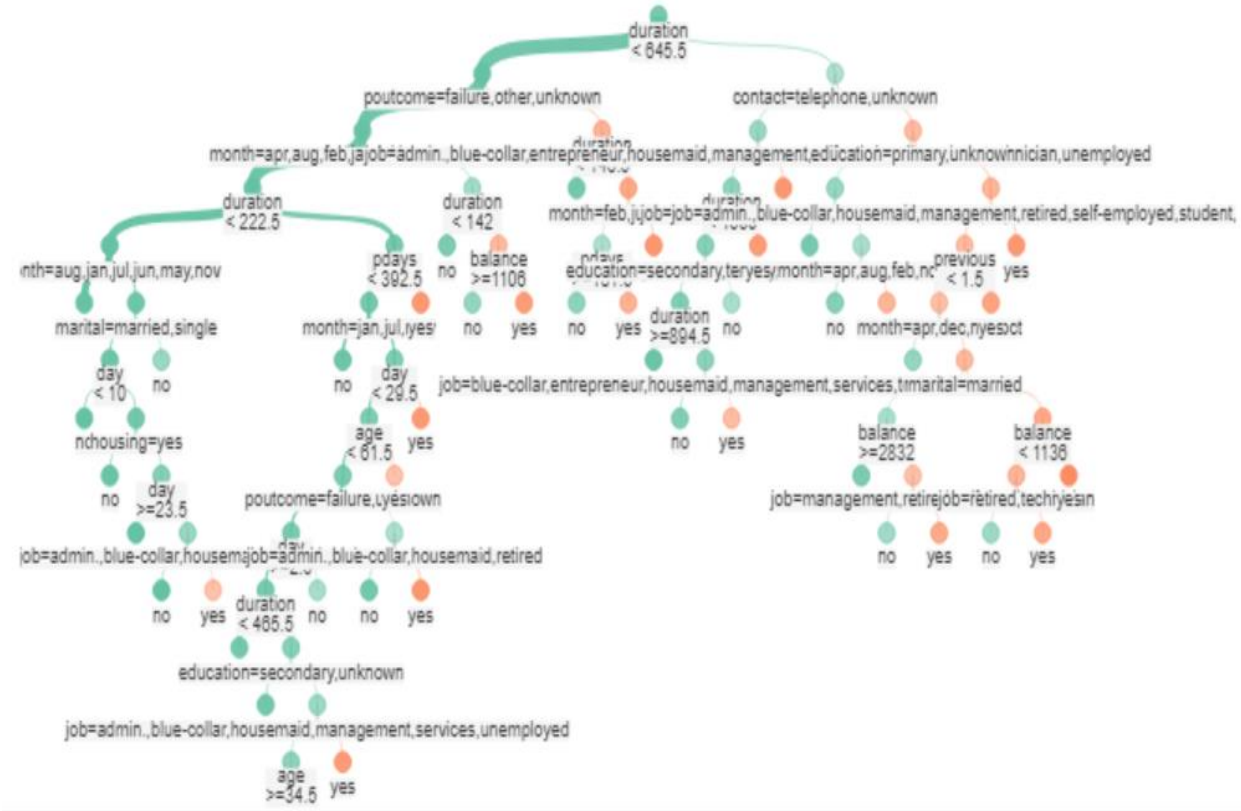
rpart(formula = Target ~ age + job + marital + education + default + balance + housing + loan + contact + day + month + duration + campaign + pdays + previous + poutcome, data = the.data, minsplit = 20)



Classification/ Confusion matrix for training data



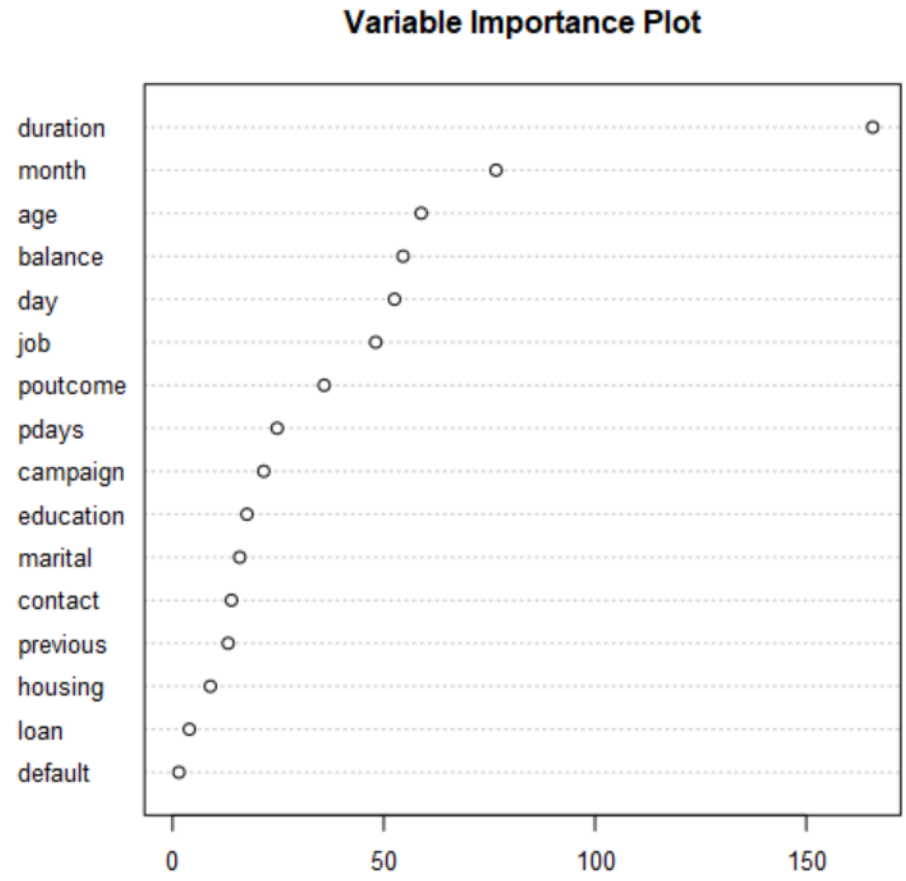
Decision Tree



Random Forest Model

The Forest Model tool creates a model that constructs a set of decision tree models to predict a target variable based on one or more predictor variables. The different models are constructed using random samples of the original data, a procedure known as bootstrapping.

randomForest(formula = Target ~ age + job + marital + education + default + balance + housing + loan + contact + day + month + duration + campaign + pdays + previous + poutcome, data = the.data, ntree = 500)



Confusion Matrix:

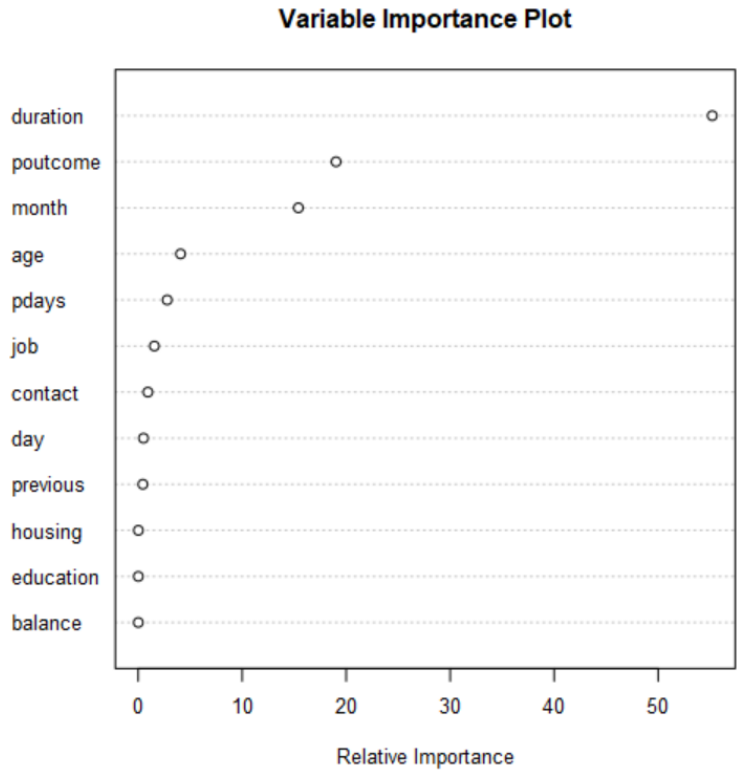
	Predicted		
	No	Yes	Classification Error
no	2725	91	0.032
yes	221	128	0.633

Boosted Model

This tool provides generalized boosted regression models based on the gradient boosting methods of Friedman. This method is a modern statistical learning model, that: (1) self-determines which subset of fields best predict a target field of interest; (2) is able to capture highly nonlinear relationships and interactions between fields; and (3) can automatically address a broad range of regression and classification problems in a way that can be transparent to the user (the user can do as little as specify a target field and a set of predictor fields, but the tool can be extensively fine-tuned by advanced users). The tool is applicable to a wide range of problems (e.g., classification, count data, and continuous target regression problems). It works by serially adding simple decision tree models to a model ensemble so as to minimize an appropriate loss function.

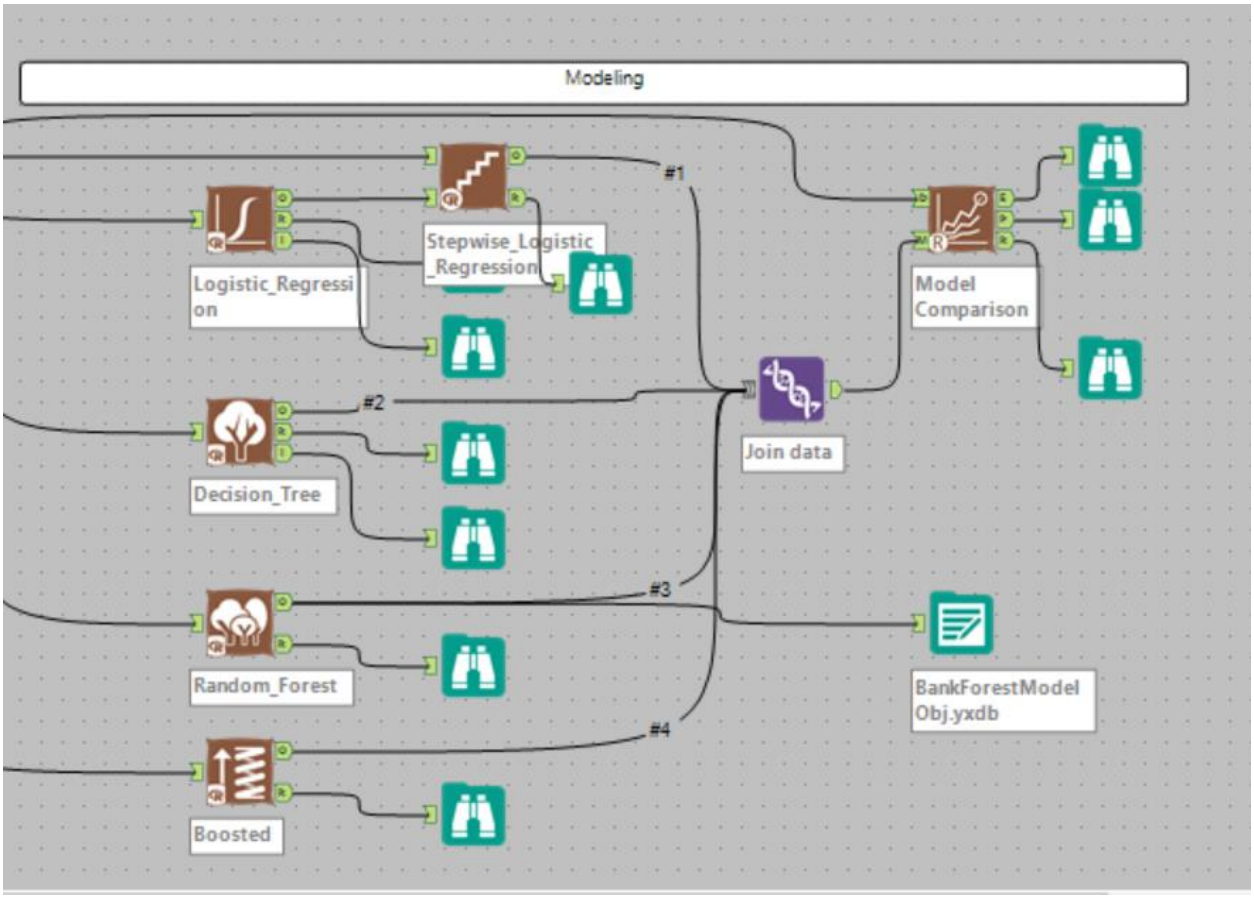
Loss function used : Bernoulli

Total number of trees used = 4000



Model Comparison

Next, we join the outputs from all the models using the join tool and feed it to the 'Model Comparison Tool'. The tool uses test data, the 30% data we split from the original data set to compare characteristics of the models.



Output of the Model Comparison tool

Record #	Model	Accuracy	Accuracy_no	Accuracy_yes	F1	AUC
1	Stepwise_Logistic_Regression	0.893068	0.973818	0.337209	0.94084	0.90047
2	Decision_Tree	0.884218	0.961149	0.354651	0.935471	0.853576
3	Forest	0.890118	0.962838	0.389535	0.938658	0.909054
4	Boosted	0.887906	0.975507	0.284884	0.938262	0.897799

Accuracy: overall accuracy, number of correct predictions of all classes divided by total sample number.

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.

Classification/ Confusion matrix across all models

Confusion matrix of Boosted

	Actual_no	Actual_yes
Predicted_no	1155	123
Predicted_yes	29	49

Confusion matrix of Decision_Tree

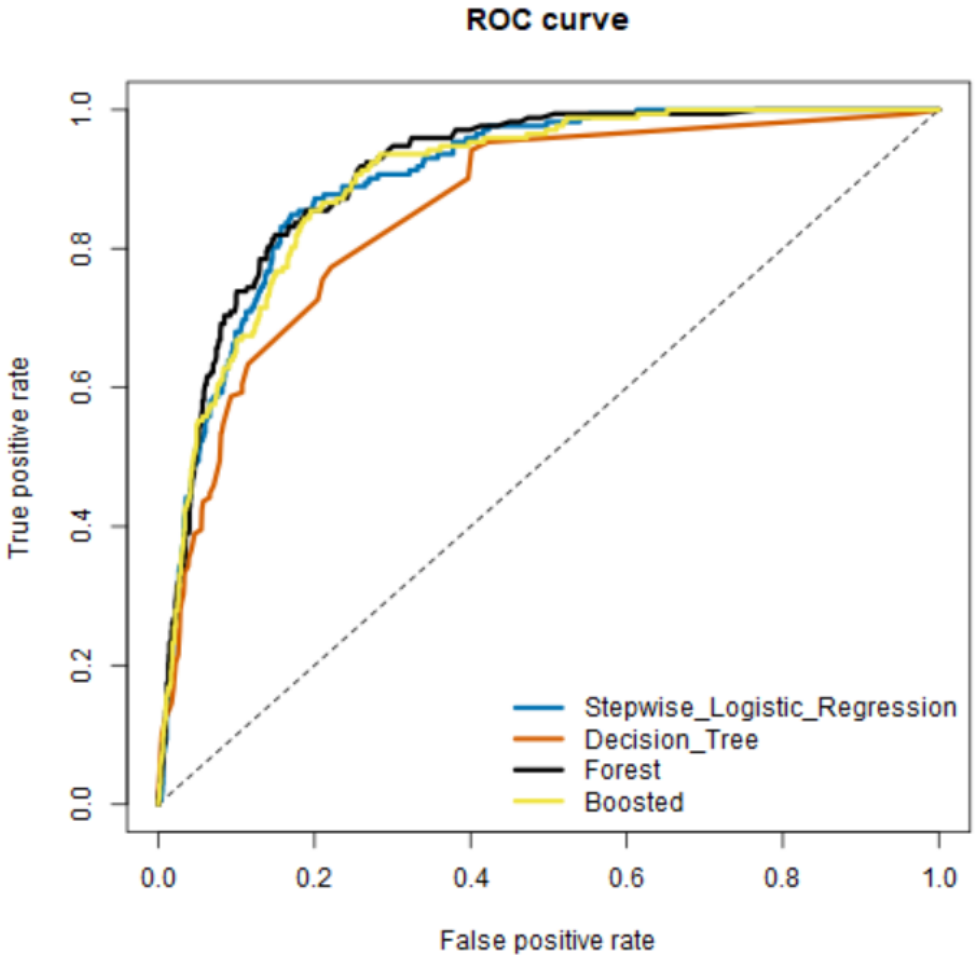
	Actual_no	Actual_yes
Predicted_no	1138	111
Predicted_yes	46	61

Confusion matrix of Forest

	Actual_no	Actual_yes
Predicted_no	1140	105
Predicted_yes	44	67

Confusion matrix of Stepwise_Logistic_Regression

	Actual_no	Actual_yes
Predicted_no	1153	114
Predicted_yes	31	58

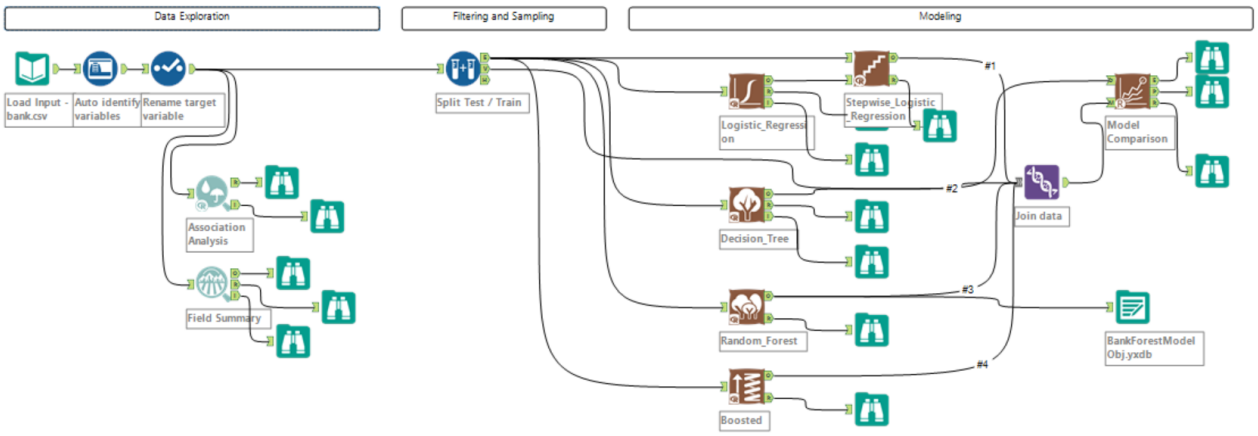


Looking across all the characteristics, it looks like all the models have high accuracy but that can be misleading, it becomes clearer when we look at accuracy of yes / no. While all models perform well when predicting no's, none of them even reach 50% accuracy for predicting yes. However, Random

Forest and Logistic Regression are more accurate than the other two. Random Forest has an accuracy of 89.01% while Logistic Regression is slightly higher at 89.31%. Looking at the ROC curve, it looks like Forest model has an upper hand as it rises faster compared to logistic regression. Now, again talking about accuracy of yes and no, while logistic regression is better at predicting no, forest model is better at predicting yes. Looking at our problem statement, it would make sense to choose a model which is more accurate at predicting yes.

For these reasons, we should go ahead with the Forest Model.

Our final workflow in Alteryx looks like below:



Conclusion

The Random Forest binary classification model with 38.95% yes accuracy and 89.01% accuracy overall will help the bank to optimize targeting efforts for its marketing campaigns. This will help the bank to predict the success of subscribing a long-term deposit even before the telemarketing call is executed.

In future work, I would like to fine tune these models to improve accuracy and possibly use a neural network model in addition to the four models we saw in this project.

References

https://archive.ics.uci.edu/ml/datasets/bank+marketing

http://media.salford-systems.com/video/tutorial/2015/targeted_marketing.pdf

https://community.alteryx.com/t5/Alteryx-Knowledge-Base/Tool-Mastery-Index/ta-p/84593

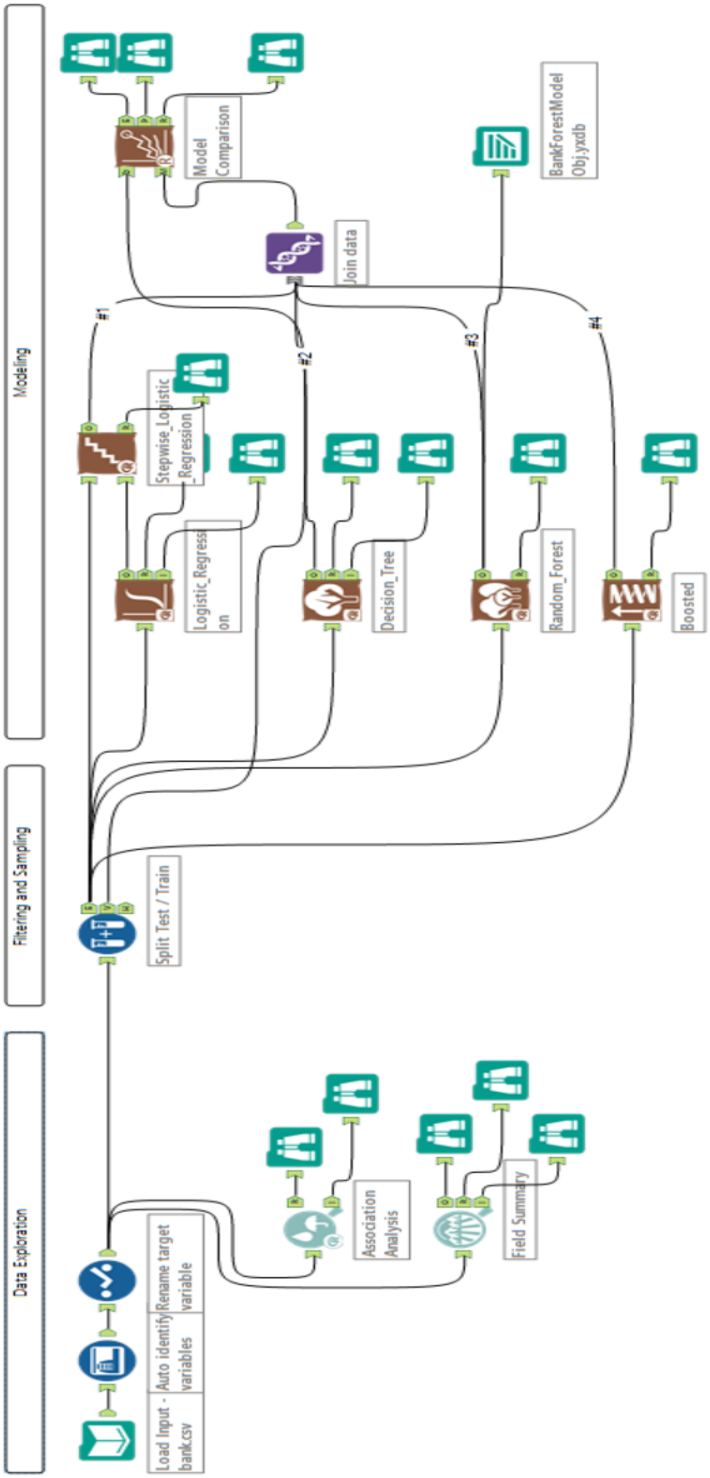
https://help.alteryx.com/9.5/Boosted_Model.htm

https://help.alteryx.com/2018.3/randomForest.htm

https://help.alteryx.com/9.5/logistic.htm

https://help.alteryx.com/9.5/rpart.htm

Workflow


Modeling
Filtering and Sampling
Data Exploration
Model Comparison
BankForestModel Obj.ydb
Join data
#1
#2
#3
#4
Stepwise_Logistic_Regression
Logistic_Regression
Decision_Tree
Random_Forest
Boosted
Split Test / Train
Association Analysis
Field Summary
Load Input - bank.csv
Auto identify variables
Rename target variable