Non-Predictive Use of Decision Tree and Friends

How supervised machine learning models can help you beyond the usual task of prediction and classification

Gerhard Svolba, Data Scientist, SAS Austria

Data Scientist @SAS - Medium | LinkedIn | Github | SAS-Books | SAS Articles
Youtube: DataPreparation4DataScience | Data Science Use Cases

Content

- Supervised machine learning: Methods and typical applications
- Beyond prediction and classification:

  1. Imputing missing values | 2. Determining the nature of missing values |
  3. Explaining complex ML models | 4. Profiling the nature of clusters and segments |
  5. Building an expert model | 6. Calculating meaningful reference values |
  7. Validating existing business rules
- Summary and closing

Frequently used methods in supervised machine learning

- Decision Trees
- Logistic Regression
- Linear Regression
- Poisson Regression
- Gradient Boosting
- Random Forests
- Support Vector Machines
- Neural Networks
- Bayesian Networks

Predicting an interval scaled variable, e.g. Customer Revenue

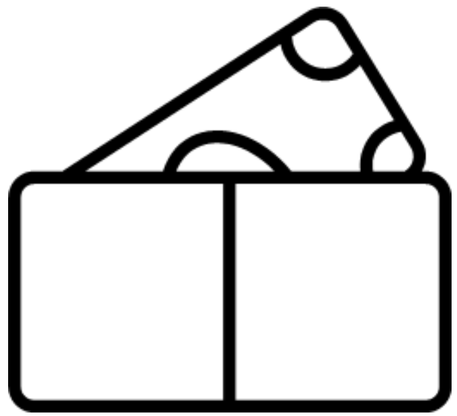
Demographic Information

Product Usage Information

Payment/Delay Information

Products/Bundles/Services Consumed

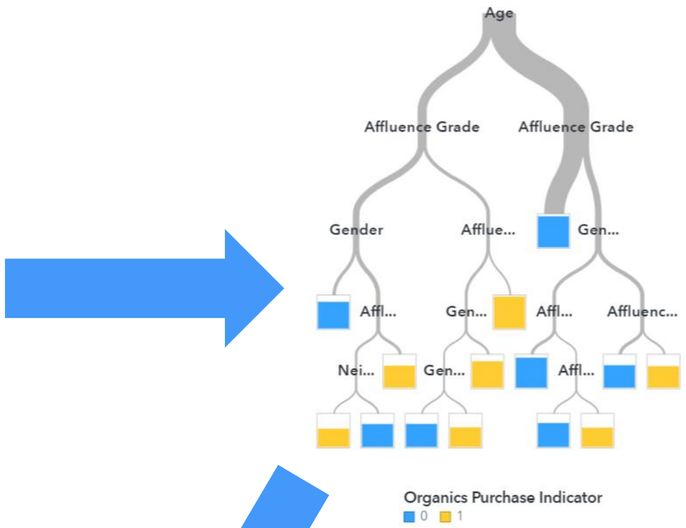
Predicting a categorical variable, e.g. events like Debt Repayment



- Demographic Information
- Credit History
- Past Delinquent Loans
- Debt to Income Ratio

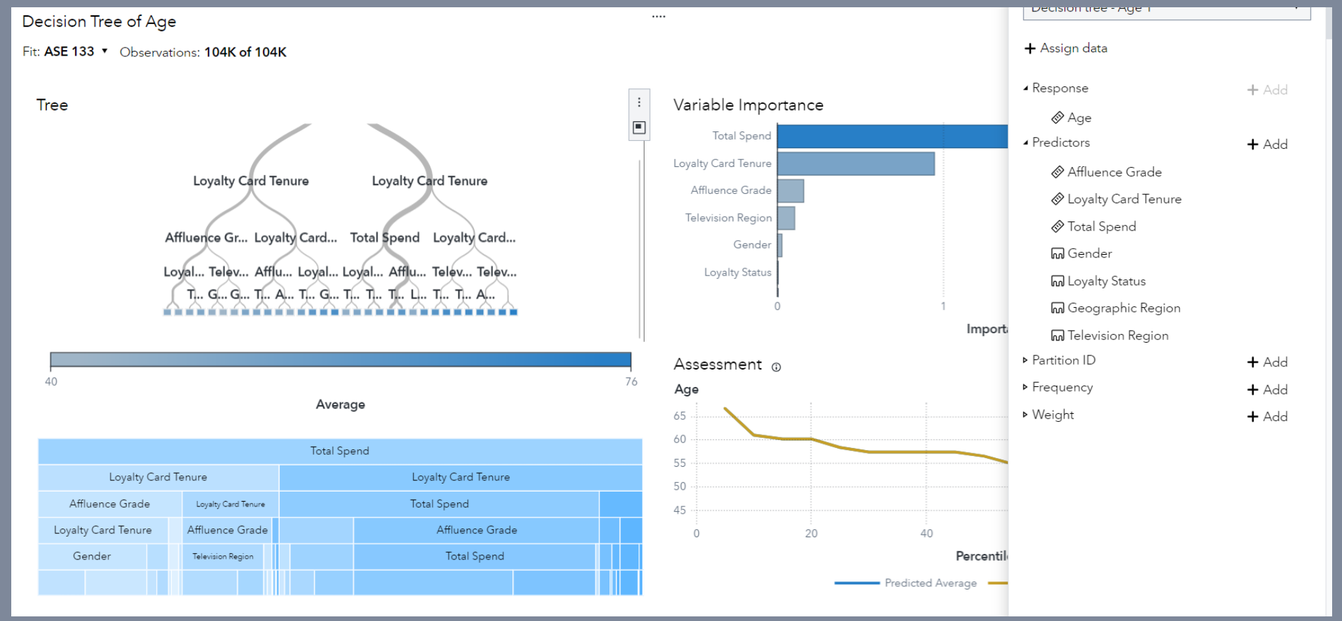
Technically: Add a column with probabilities or expected values to your data tables

Customer Loyalty ID	Gender	Age	Loyalty Status	Geographic Region	Total Spend	Organics Purchase Indicator
A0000054496	F	31.0	Tin	North	0	0
A0000103225	F	18.0	Gold	South East	6,000	0
A0000162485	F	34.0	Silver	Midlands	3,000	1
A0000263419	F	27.0	Tin	South East	0	1
A0000349291	M	26.0	Silver	South East	4,000	0
A0000423211	F	33.0	Tin	North	0	1
A0000500100	F	30.0	Silver	Midlands	100	1
A0000553115	F	29.0	Silver	Midlands	2,000	0
A0000653500	F	32.0	Silver	Midlands	2,500	1
A0000775258	F	31.0	Silver	North	1,100	0
A0000811831		32.0	Tin	North	0	1



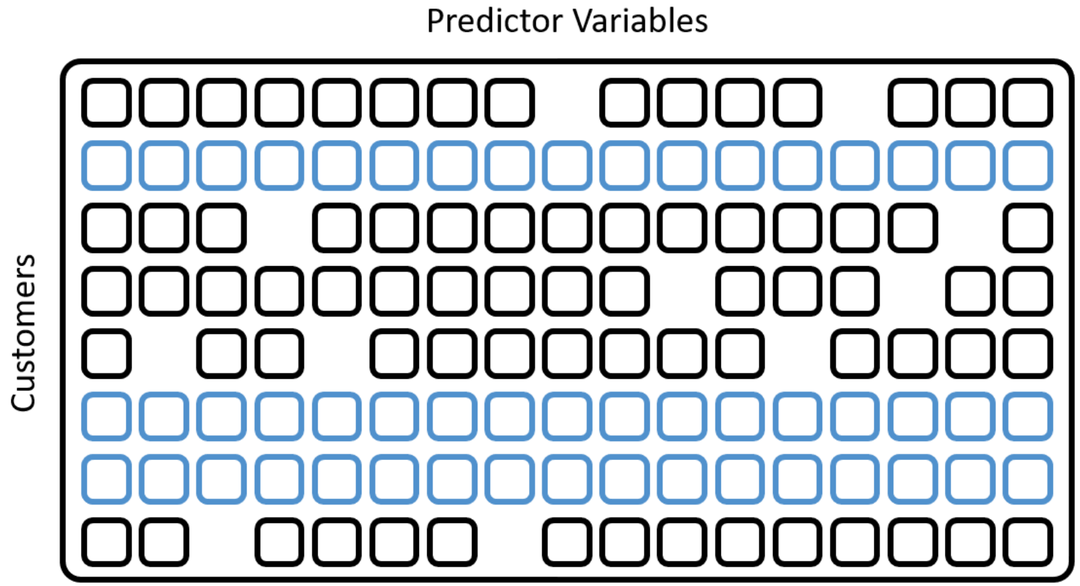
Customer Loyalty ID	Gender	Age	Loyalty Status	Geographic Region	Total Spend	ProbPurchase_DecisionTree
A0000041092	F	50.0	Gold	Midlands	9,800	6.87%
A0000041475	M	66.0	Tin	South East	0	9.81%
A0000042439	F	35.0	Silver	Midlands	1,630	92.42%
A0000045799	F	71.0	Silver	Midlands	1,500	24.26%
A0000046906	F	44.0	Silver	North	1,000	37.68%
A0000050237	M	42.0	Tin	North	0	22.80%
A0000051950	F	68.0	Gold	South West	6,053	22.76%
A0000054496	F	31.0	Tin	North	0	60.91%
A0000056073	F	70.0	Gold	South East	19,054	18.09%
A0000057002	F	72.0	Platinum	North	20,527	13.80%

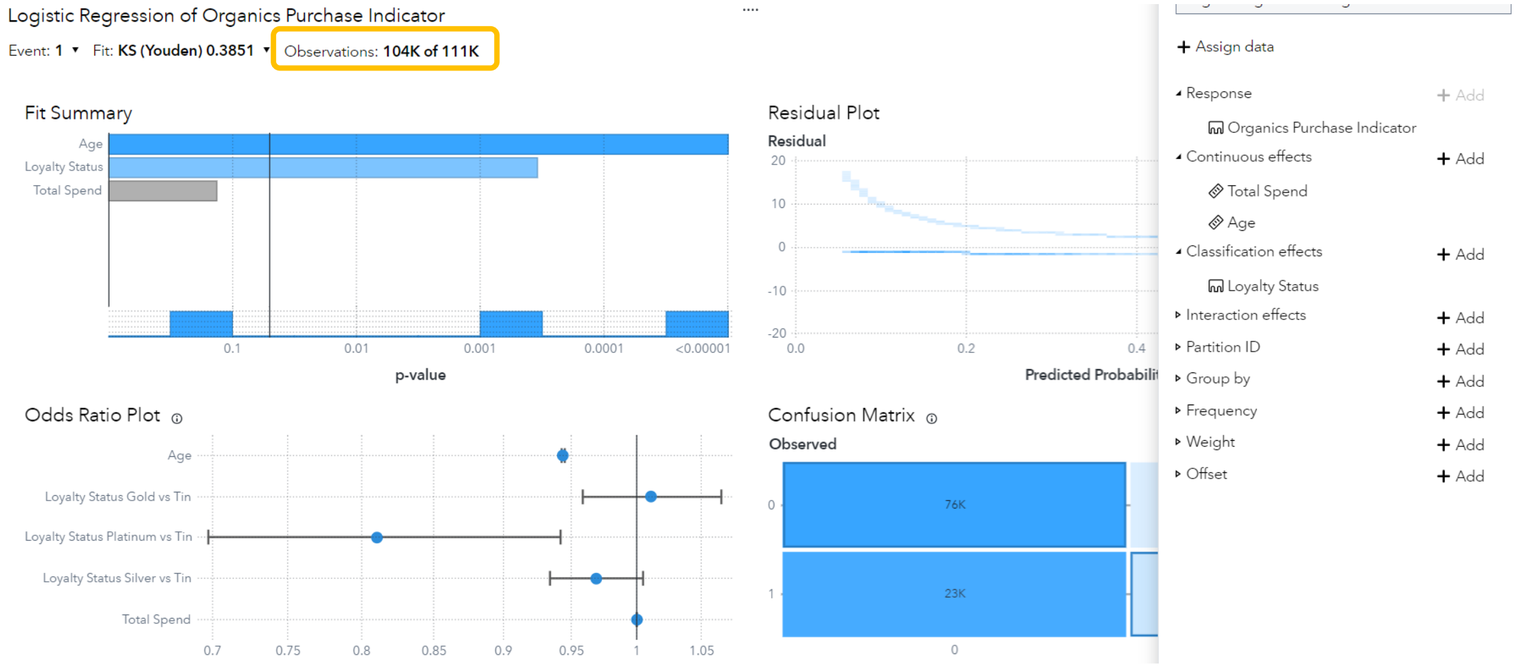
#1
Imputing Missing Values



Analytical requirement: Non-Missing Values

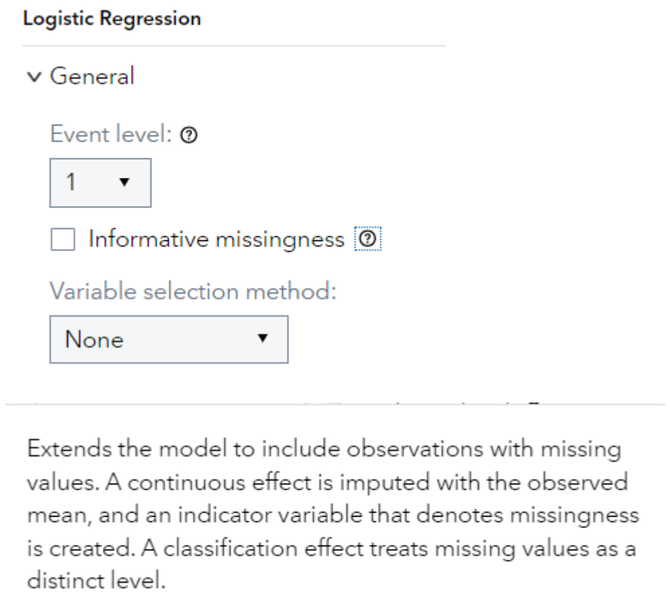
- Methods like Linear and Logistic regression, support vector machines or neural networks require non-missing values for the observations to be used in the analysis.
- Different from decision trees, gradient boosting, or random forests observations with missing values are excluded from the analysis



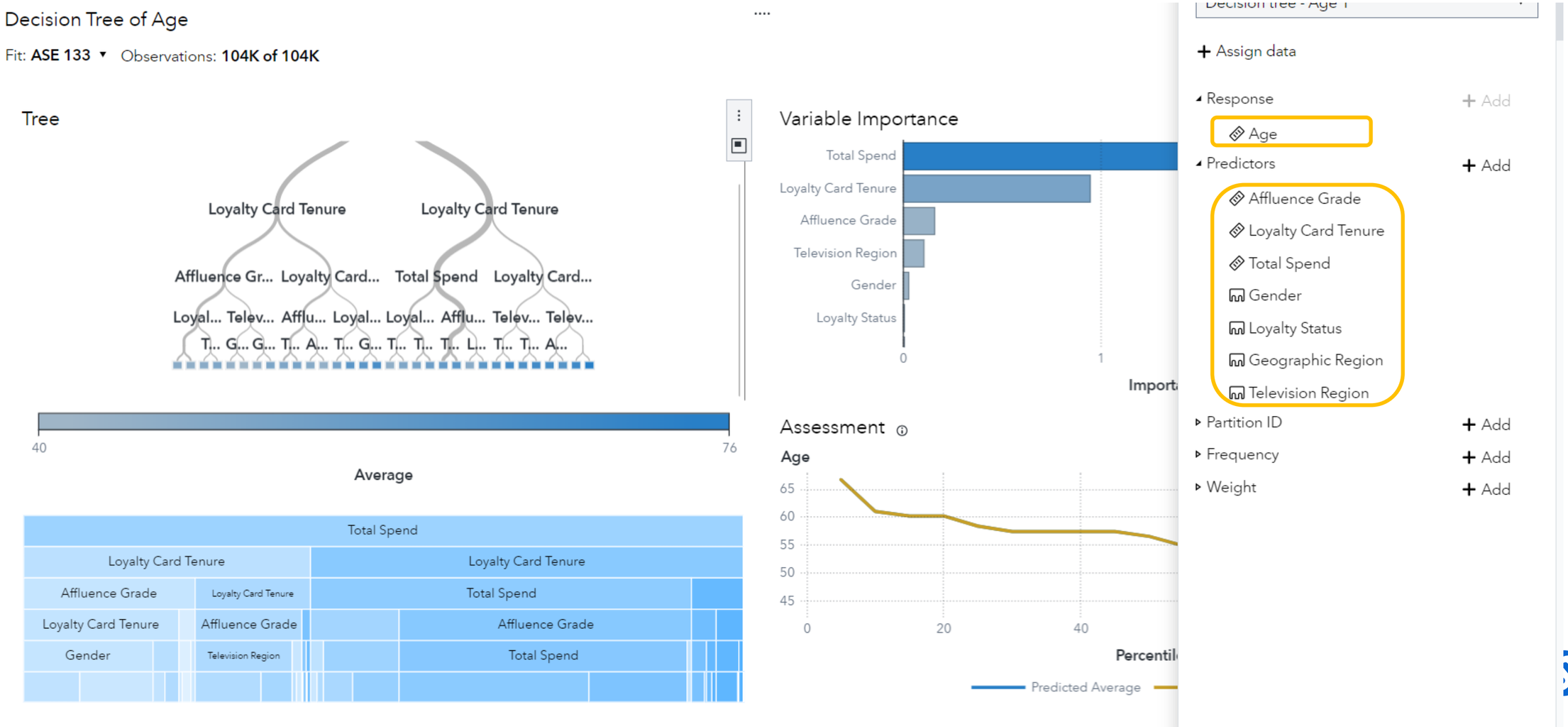


Basic Idea

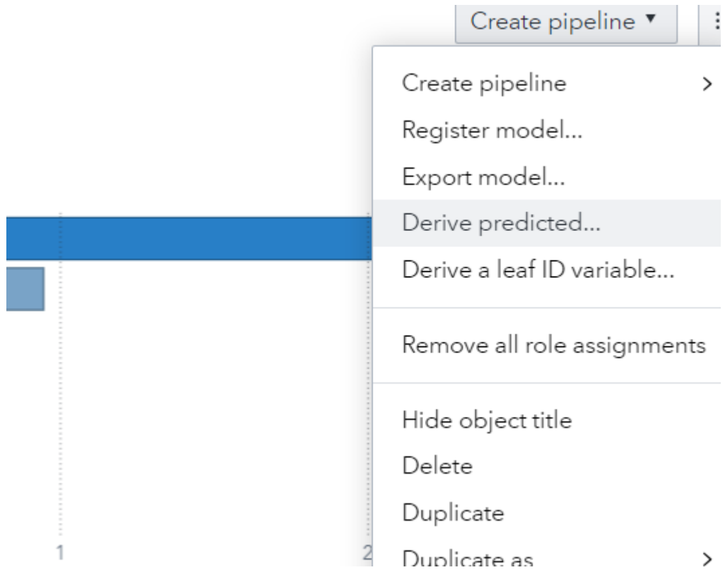
- Automatic options to perform imputation using the average exist.
- However you might want to use better imputation methods that provide a more appropriate imputation value than just the mean.
- 1. Train a predictive model to learn the relationships between a variable (e.g. age) and other predictor variables
- 2. Use this logic to calculate an individual imputation value for each observation based on the value of other variables.

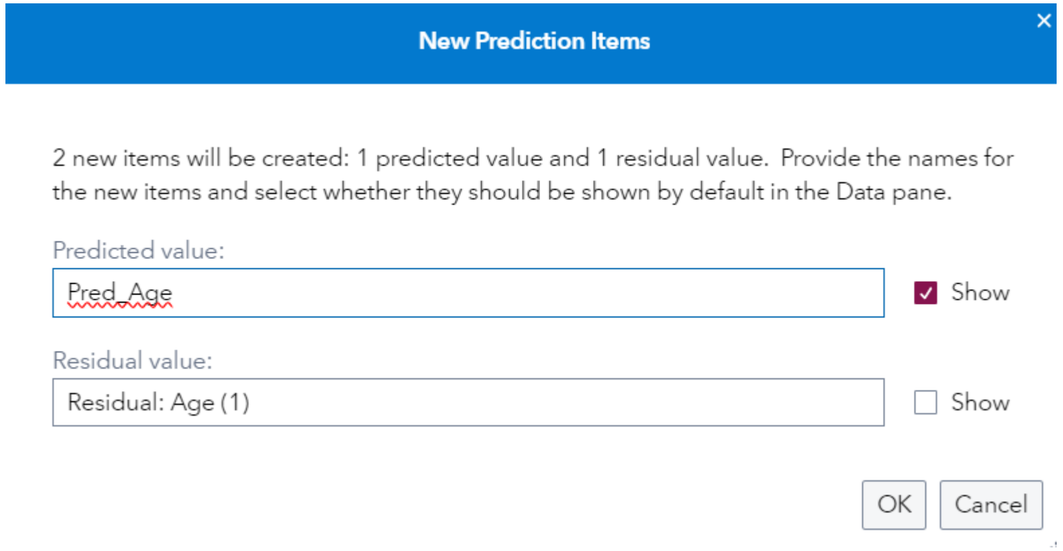


Step 1: Train a predictive model for the variable that shall be imputed (e.g. age) (ideally use a tree-based-method, that can deal with missing values in the other predictor variables)

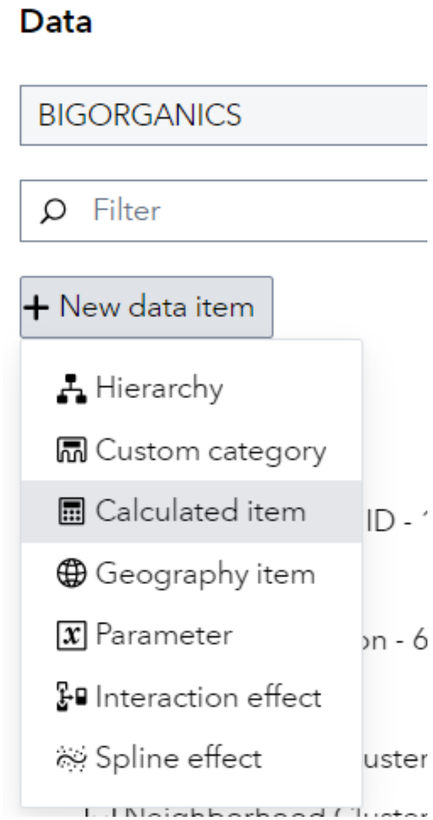


Step 2: Add the predicted values for age to your data
→ Snowman Icon
→ Derived Predicted
→ Name the new variable





Step 3: Create a new calculated item that contains the predicted value, if age is missing otherwise the original age value



Name: *

Age_Imputed

COMMA12.2

Operators | Functions | Data | New parameter

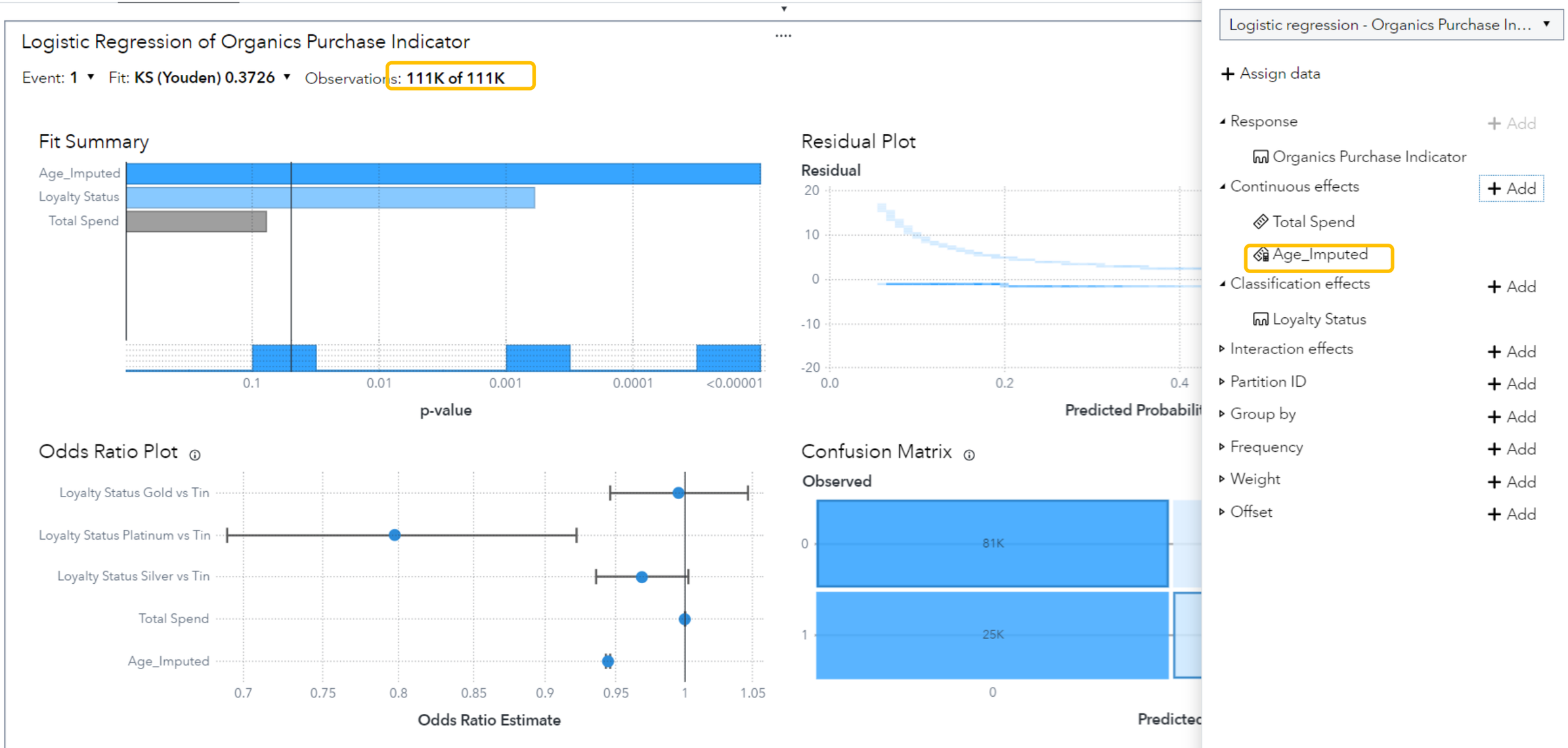
```
1  IF (Missing(Age))
2      RETURN (Pred_Age)
3  ELSE (Age)
```

Results

Preview selection only

Age	Pred_Age	Age_Imputed
.	46.7	46.73
78.0	60.2	78.00

Step 4: Use the newly created variable in your logistic regression.
Now all observations are used for the analysis.



Using SAS Procedures to derive the imputation values

```
proc gradboost data=tundata.bigorganics;

 target DemAge / level = interval;
 input PromSpend DemAffl / level = interval;
 input DemGender PromClass DemTVReg / level=nominal;

  *** Scoring Option 1: Create Prediction for the existing observations in the dat
 output out=casuser.BigOrg_PredAge copyvar=(_all_);

run;

  *** Scoring Option 1:  eate Prediction for the existing observations in the dat

data casuser.BigOrg_PredAge;
 set casuser.BigOrg_PredAge;
 if missing(DemAge) then DemAgeImpute = P_DemAge;
 else DemAgeImpute = DemAge;
run;

proc print data=casuser.BigOrg_PredAge (obs=20);
 var id TargetBuy DemGender DemAge P_DemAge DemAgeImpute;
run;
```

Obs	id	TargetBuy	DemGender	DemAge	P_DemAge	DemAgeImpute
1	A0048728650	1	F	40	48.236314289	40.0000
2	A0048732291	0	M	32	48.851559997	32.0000
3	A0048740120	0	F	76	56.780026186	76.0000
4	A0048742270	0	F	62	58.735273005	62.0000
5	A0048745393	0	F	63	60.832845707	63.0000
6	A0048746411	0	M	46	47.174948547	46.0000
7	A0048746918	0	F	.	53.093781817	53.0938
8	A0048749453	0	F	64	57.85587556	64.0000
9	A0048757073	0	F	67	61.257234116	67.0000
10	A0048760740	0	F	47	62.184909952	47.0000
11	A0048763308	0	M	66	61.296020736	66.0000
12	A0048767389	0	U	63	61.294819194	63.0000
13	A0048767893	0	M	62	51.924448665	62.0000
14	A0048768393	1	F	41	47.358854347	41.0000
15	A0048768890	0	U	46	57.721045161	46.0000
16	A0048769900	0		40	49.609519829	40.0000
17	A0048771416	1	M	42	58.733148439	42.0000
18	A0048779123	1	F	65	52.767847116	65.0000
19	A0048782736	1	M	60	58.977362545	60.0000
20	A0048784259	0	F	31	48.046709331	31.0000

Creating the scores (imputation values) not only for the actual but also for fresh data
→ Create a scoring logic

```
proc gradboost data=tundata.bigorganics;

 target DemAge / level = interval;
 input PromSpend DemAffl / level = interval;
 input DemGender PromClass DemTVReg / level=nominal;

 *** Scoring Option 1: Create Prediction for the existi
 output out=casuser.BigOrg_PredAge copyvar=(_all_);

run;
```

```
proc gradboost data=tundata.bigorganics;

 target DemAge / level = interval;
 input PromSpend DemAffl / level = interval;
 input DemGender PromClass DemTVReg / level=nominal;

 *** Scoring Option 1: Create Prediction for the existing obser
 output out=casuser.BigOrg_PredAge copyvar=(_all_);

 *** Scoring Option 2: Create Predictions for fresh data;
 *** Scoring Option 2a: Using datastep scorecode;
 code file="&path./AgeImpute_Scorecode_2a.sas";
 *** Scoring Option 2b: Using an astore;
 savestate rstore=casuser.BigOrg_AgeImputeScoreLogic_2b;
run;
```

Apply the scoring logic – SAS Score Code and SAS Astores

```
proc gradboost data=tundata.bigorganics;

 target DemAge / level = interval;
 input PromSpend DemAffl / level = interval;
 input DemGender PromClass DemTVReg / level=nominal;

 *** Scoring Option 1: Create Prediction for the existing obser
 output out=casuser.BigOrg_PredAge copyvar=(_all_);

 *** Scoring Option 2: Create Predictions for fresh data;
 *** Scoring Option 2a: Using datastep scorecode;
 code file="&path./AgeImpute_Scorecode_2a.sas";
 *** Scoring Option 2b: Using an astore;
 savestate rstore=casuser.BigOrg_AgeImputeScoreLogic_2b;
run;
```

```
 *** Scoring Option 2: Create Predictions for fresh data;
 *** Scoring Option 2a: Using datastep scorecode;

data casuser.BigOrg_Fresh_PredAgeImpute_2a;
 set tundata.BigOrganics;
 ** Creates variable P_DemAge in the data;
 %include "&path./AgeImpute_Scorecode_2a.sas";
 if missing(DemAge) then DemAgeImpute = P_DemAge;
 else DemAgeImpute = DemAge;
run;
```

```
*** Scoring Option 2: Create Predictions for fresh data;
*** Scoring Option 2b: Using an astore;

proc astore;
 score data=tundata.bigorganics
       out =casuser.BigOrgFresh_PredAgeImpute_Ast
       rstore=casuser.BigOrg_AgeImputeScoreLogic_2b copyvars=(_all_);
run;
```

```
data casuser.BigOrg_AgeImputeScoreLogic_2b;
 set casuser.BigOrg_AgeImputeScoreLogic_2b;
 if missing(DemAge) then DemAgeImpute = P_DemAge;
 else DemAgeImpute = DemAge;
run;
```

All these methods create the predicted + imputed values in the data
Difference: existing (training) data or fresh data

Obs	id	TargetBuy	DemGender	DemAge	P_DemAge	DemAgeImpute
1	A0048728650	1	F	40	48.236314289	40.0000
2	A0048732291	0	M	32	48.851559997	32.0000
3	A0048740120	0	F	76	56.780026186	76.0000
4	A0048742270	0	F	62	58.735273005	62.0000
5	A0048745393	0	F	63	60.832845707	63.0000
6	A0048746411	0	M	46	47.174948547	46.0000
7	A0048746918	0	F	.	53.093781817	53.0938
8	A0048749453	0	F	64	57.85587556	64.0000
9	A0048757073	0	F	67	61.257234116	67.0000
10	A0048760740	0	F	47	62.184909952	47.0000
11	A0048763308	0	M	66	61.296020736	66.0000
12	A0048767389	0	U	63	61.294819194	63.0000
13	A0048767893	0	M	62	51.924448665	62.0000
14	A0048768393	1	F	41	47.358854347	41.0000
15	A0048768890	0	U	46	57.721045161	46.0000
16	A0048769900	0		40	49.609519829	40.0000
17	A0048771416	1	M	42	58.733148439	42.0000
18	A0048779123	1	F	65	52.767847116	65.0000
19	A0048782736	1	M	60	58.977362545	60.0000
20	A0048784259	0	F	31	48.046709331	31.0000

Application Recommendations

- Preferred Method: Tree-based methods (Tree, Gradient Boosting, Forest)
- Recommended SAS Tool: SAS Code and SAS Visual Analytics

- Using SAS Visual Analytics for this task, requires that you create and apply the imputation model for each variable that you would like to impute.
- For a larger number of variables this might be too time consuming.
- Using this approach in SAS Languguage (e.g. using the TREESPLIT procedure and a SAS Datastep) allows you to automate this process.

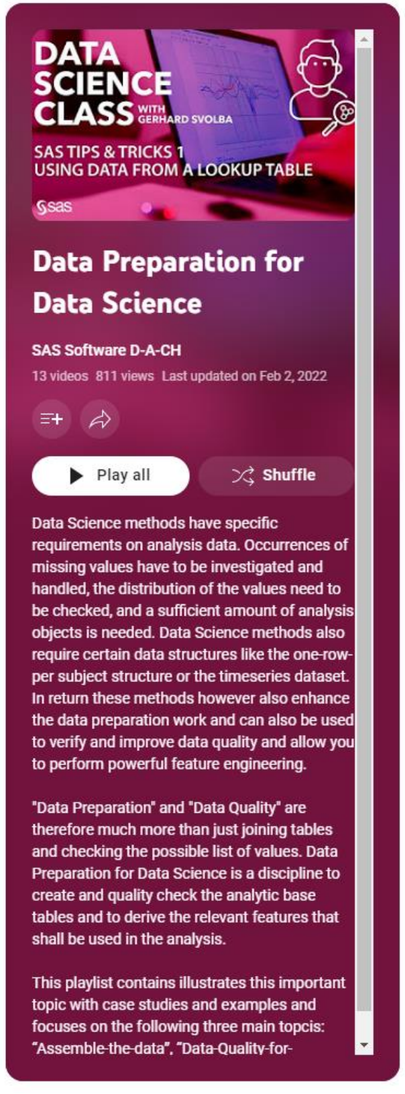
#2
Determining the nature of missing values

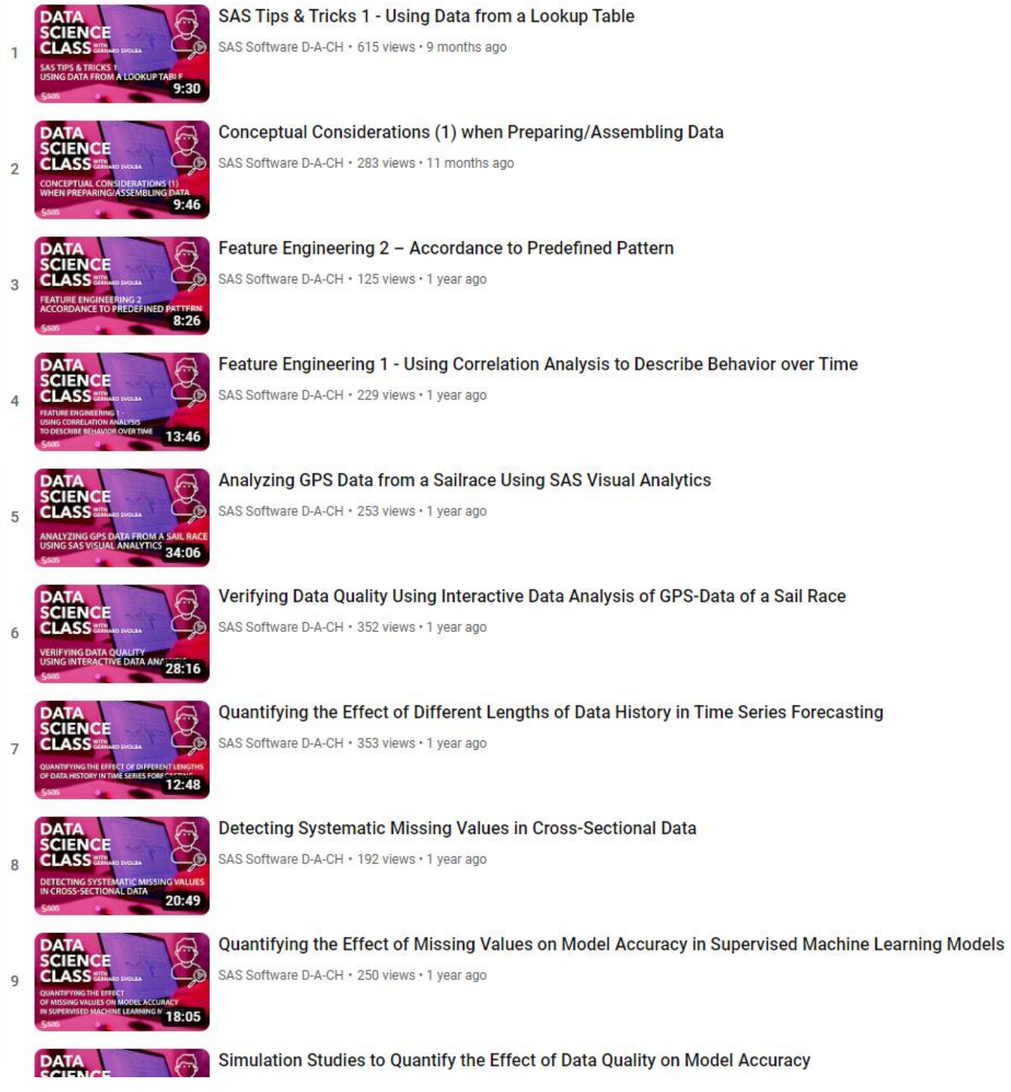
Basic Idea

- It is important to understand the nature of missing values
- Do they occur randomly or systematically?

- This effects
  - The strategy to impute the missing values (average, predictive models like in tip#1, …)
  - The way how the data can be interpreted, and learning from the data that can be made
  - Considerations for the application of the model in production

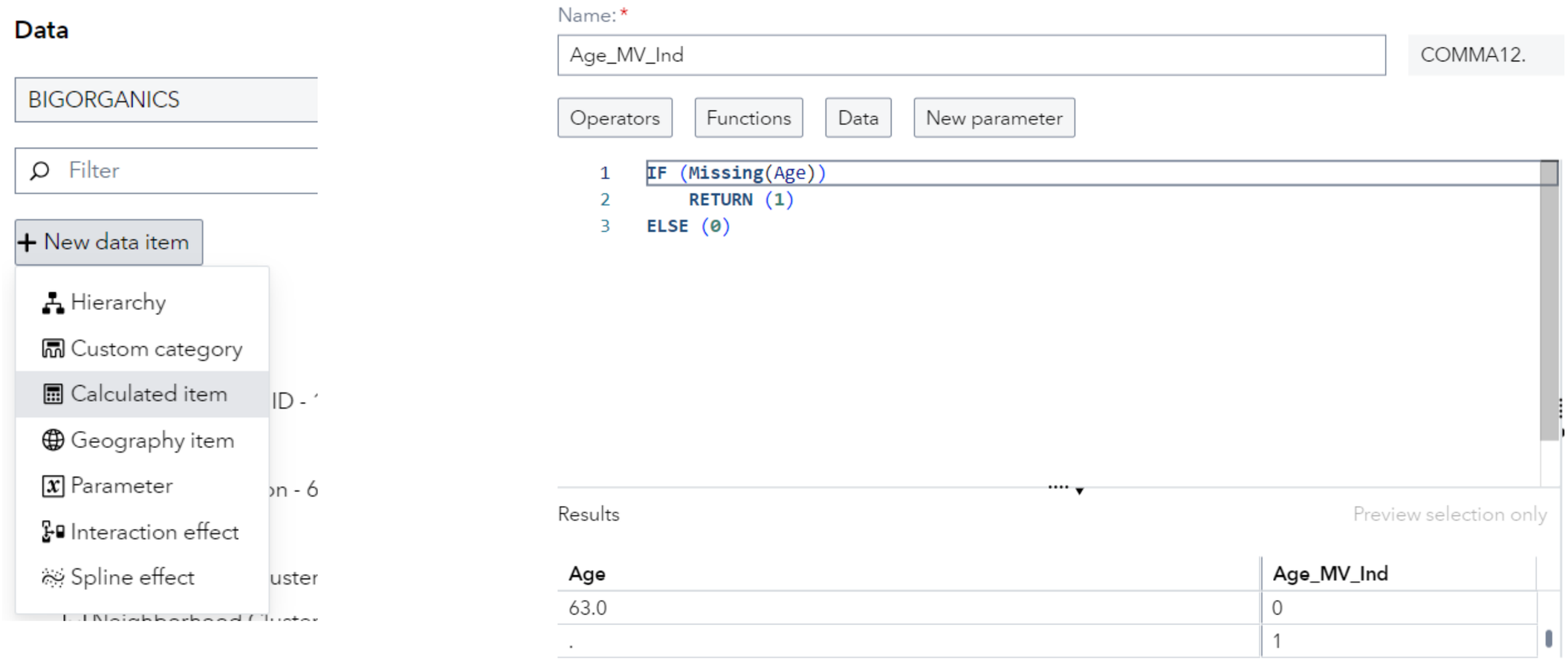
Example case: “Why my old Aunt Susanne gives data scientists a hard time“





For an example case, see the “Data Preparation for Data Science” Playlist
[Data Preparation for Data Science - Playlist]

Step 1: Create a “Calculated Item” in your Data, which returns 1 for a missing value and 0 for a non-missing

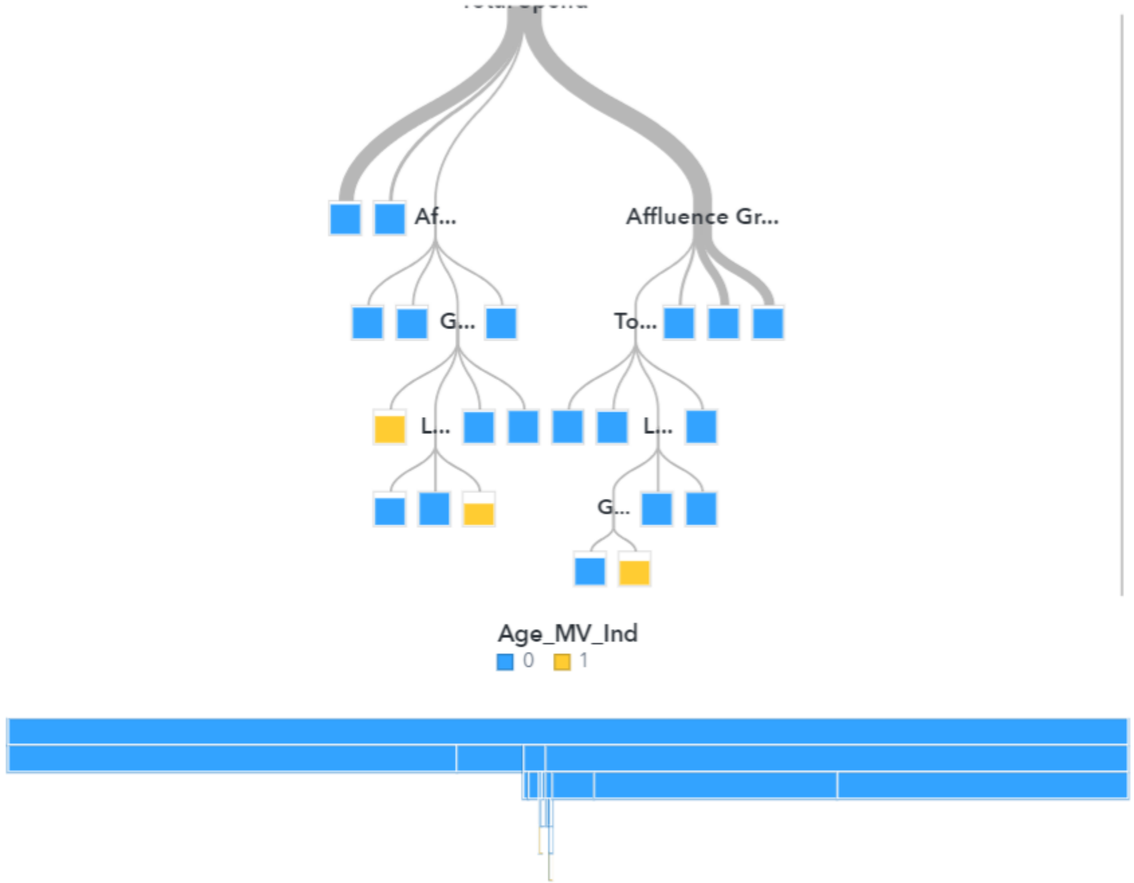


Step 2: Build a predictive model to explain “Missing yes/no”

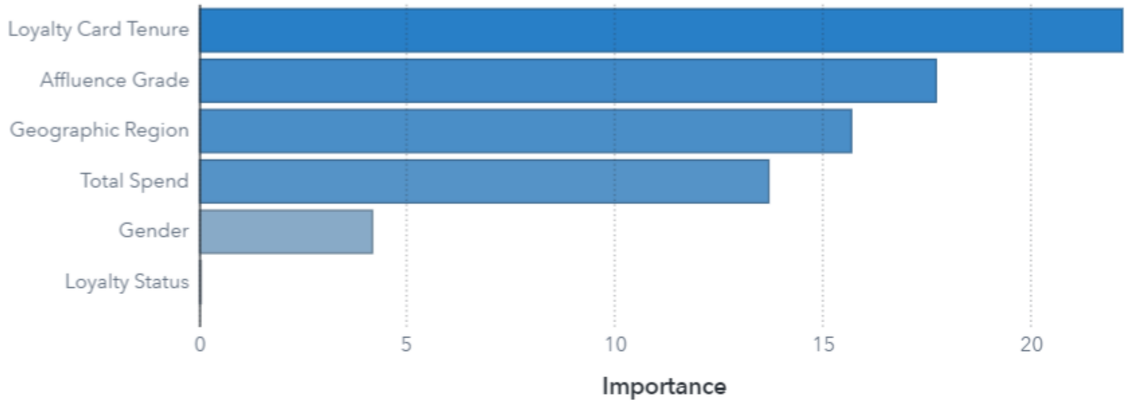
Decision Tree of Age_MV_Ind

Event: **1** ▾ Fit: **KS (Youden) 0.0515** ▾ Observations: **111K of 111K**

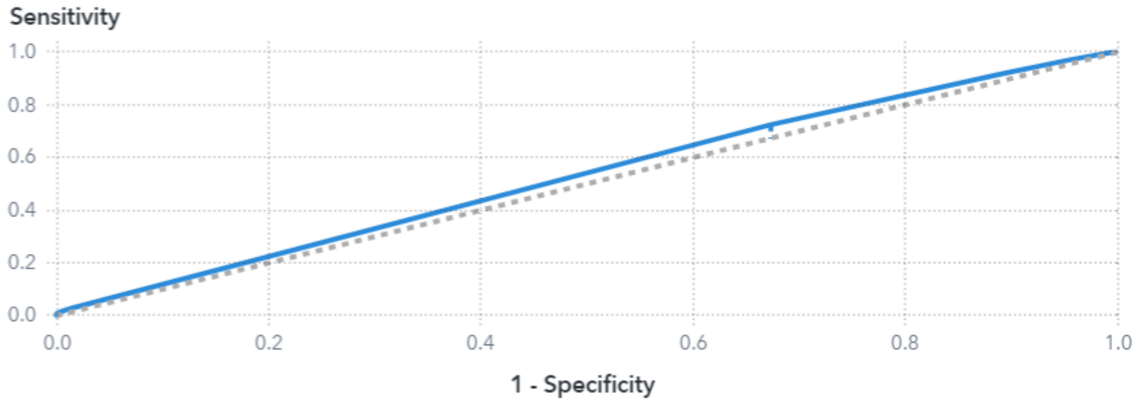
Tree



Variable Importance



ROC



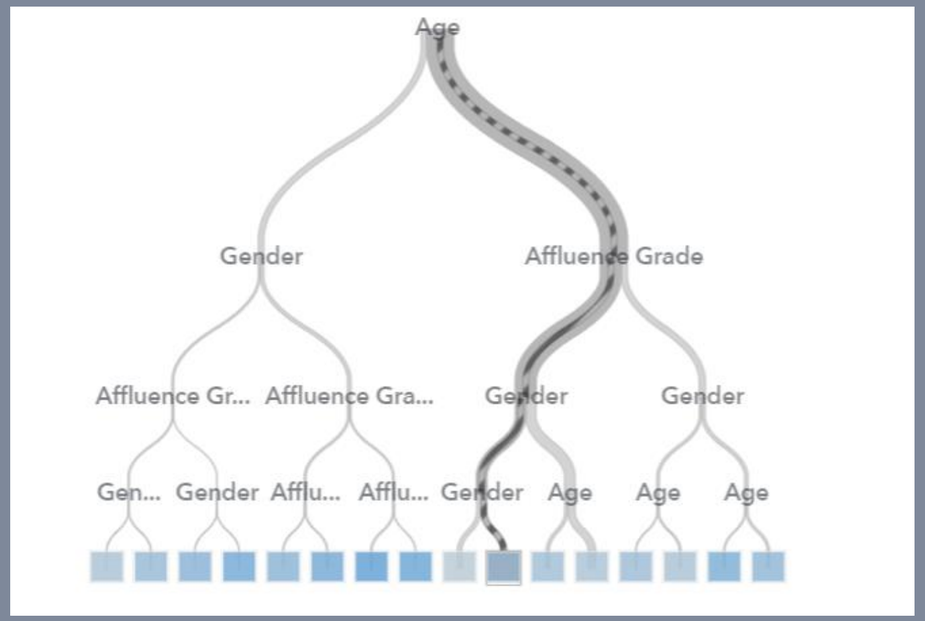
Interpretation

- If no model or a model with a very bad predictive power (e.g. a straight line the ROC-chart) can be built
  - → No relationship between the data and the fact “Age is missing” can be found
  - → Missing values in variable age rather occur randomly
- If a model with strong rules can be found
  - → The missing values of variable age occur rather systematically
  - → Study the rules in the decision tree to see when these missing values usually occur.

Application Recommendations

- Preferred Method: Decision Tree (Interpretation), predictive models in general
- Recommended SAS Tool: SAS Visual Analytics, SAS Procedures

#3
Building a surrogate model (to explain a more sophisticated ML model)

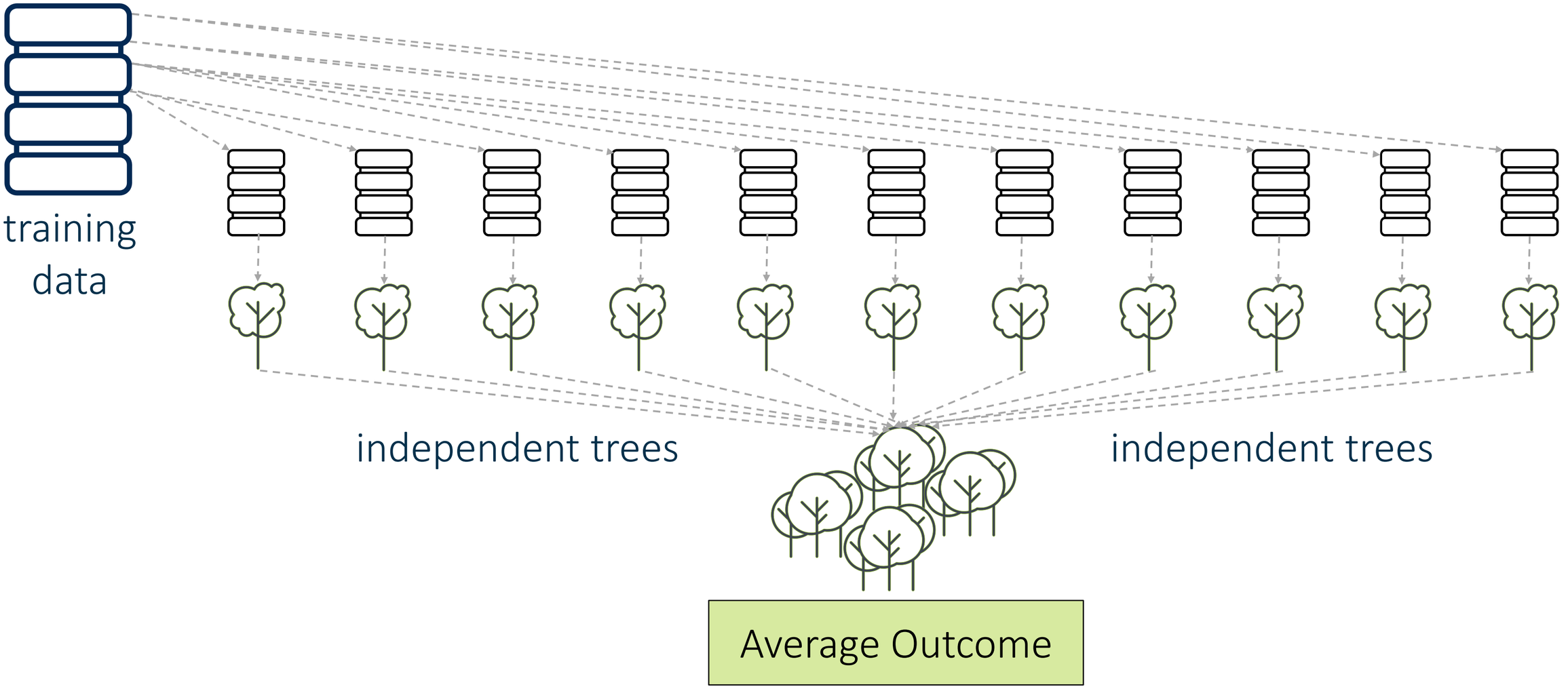


Reviewing the machine learning models by interpretability

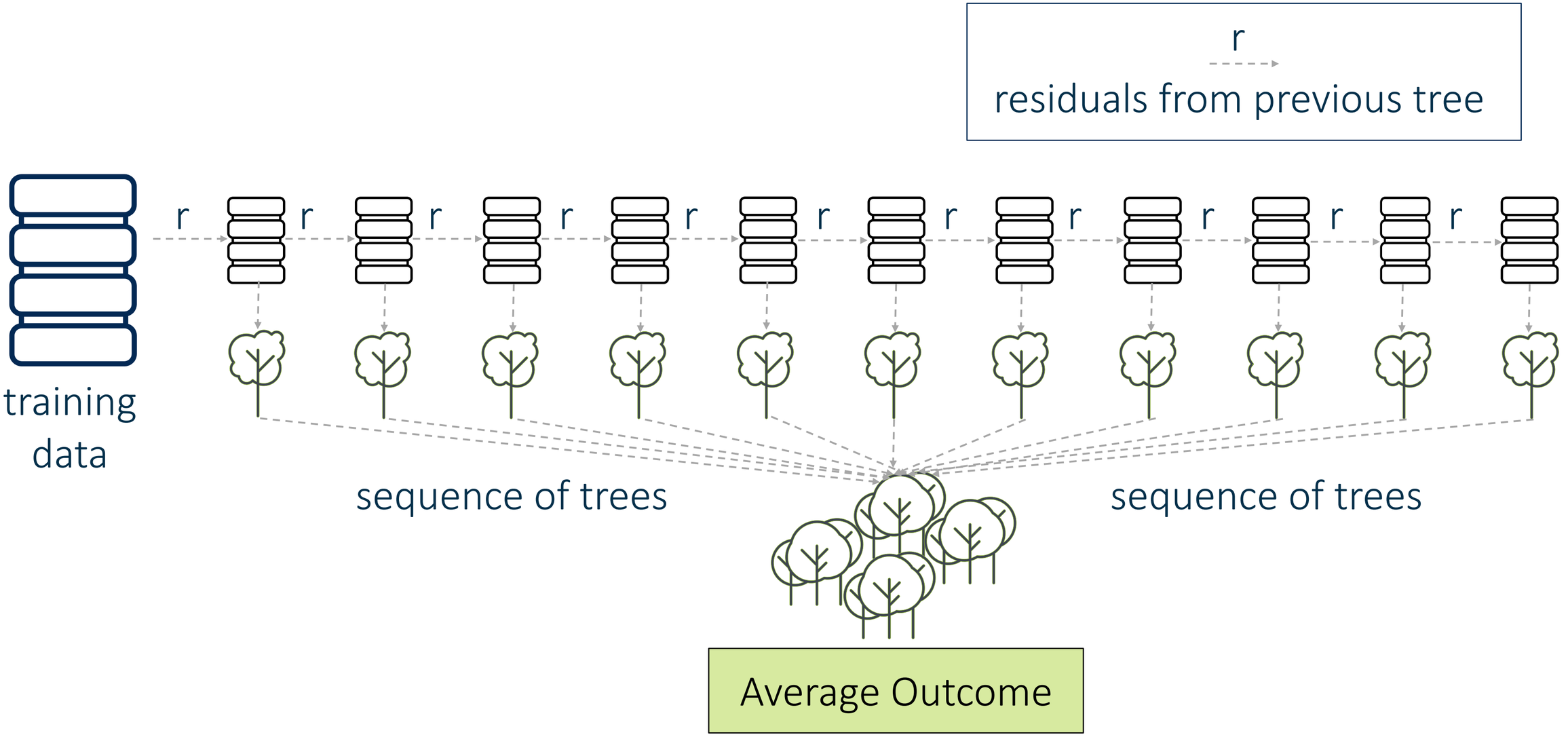
- Decision Trees
- Logistic Regression
- Linear Regression
- Poisson Regression

- Gradient Boosting
- Random Forests
- Support Vector Machines
- Neural Networks
- Bayesian Networks

Ensemble of Trees: Forest



Ensemble of Trees: Gradient Boosting

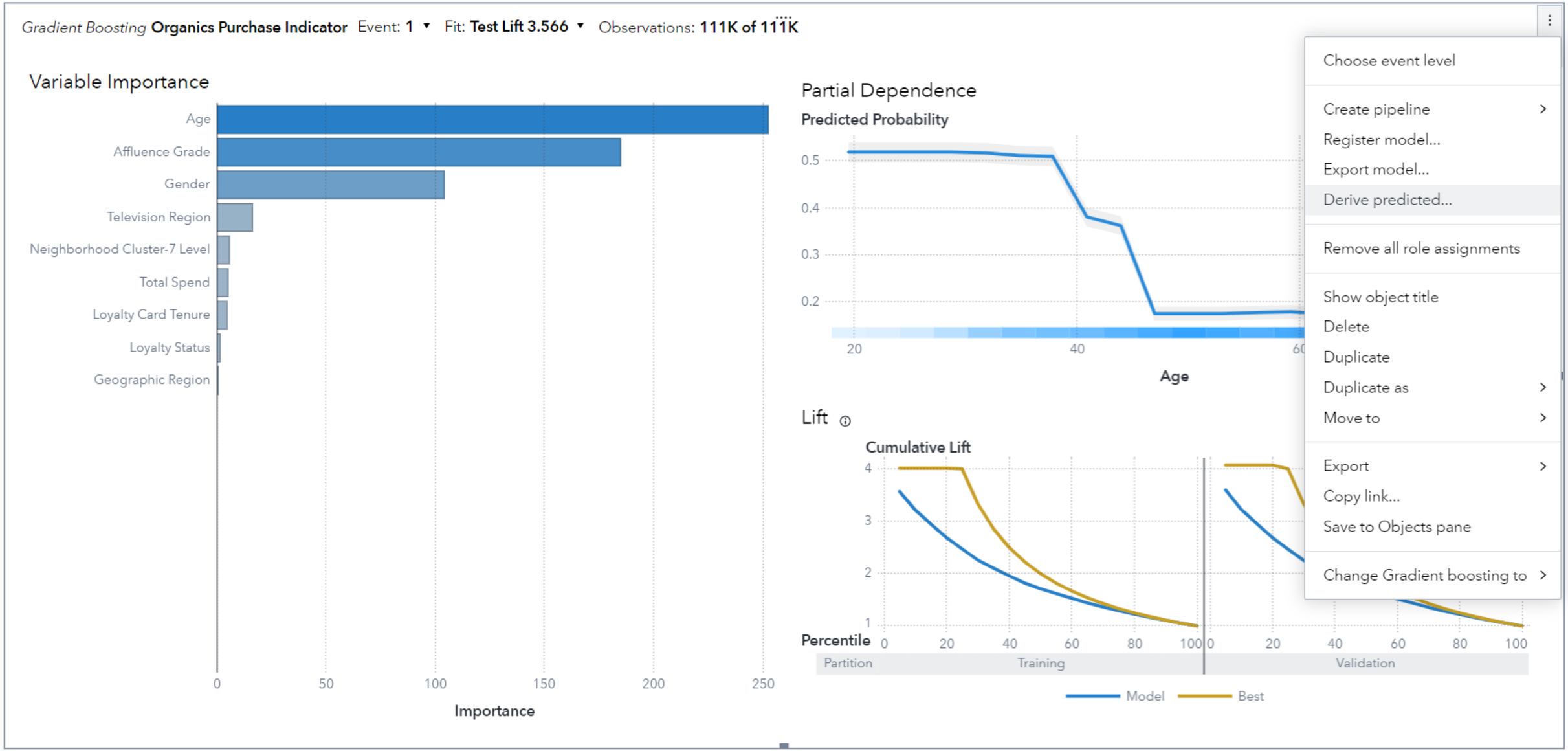


Global and Local Interpretability

- Global Interpretability
  - Variable Importance
  - Partical Dependence (PD)
  - Surrogate Model
- Local Interpretability
  - Independent Local Expectation (ICE)
  - Local interpretable model-agnostic explanations (LIME)
  - Shapley Explainer

Basic Idea:
Use a decision tree to „explain“ why analysis subjects (customers, …) receive a high/low predicted value

Step 1: Derive the predicted values from your champion model

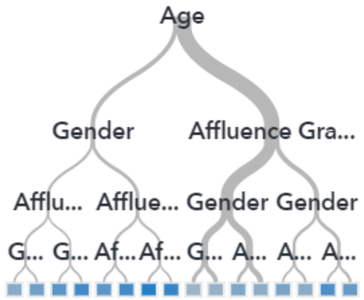


Step 2: Build a decision tree to “explain” why some probabilities are high, others are low

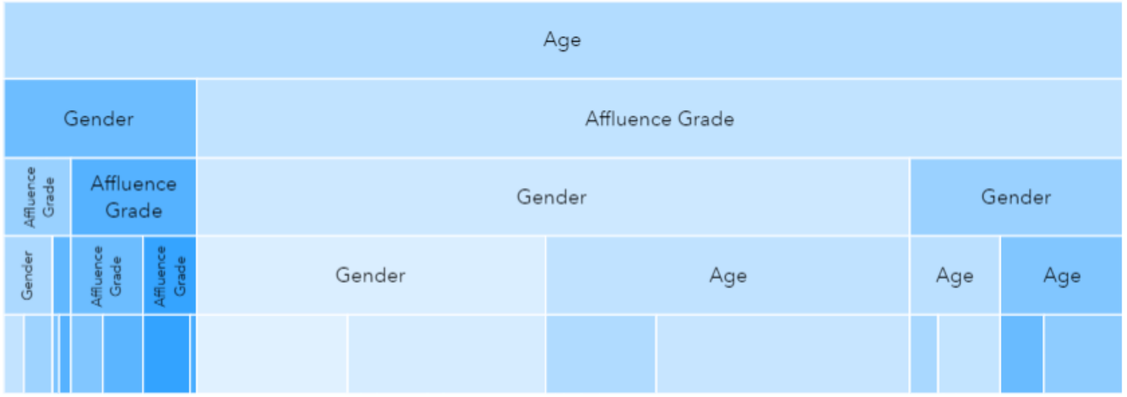
Decision Tree of ProbPurchase_GradBoosting

Fit: **ASE 0.01** ▾ Observations: **111K of 111K**

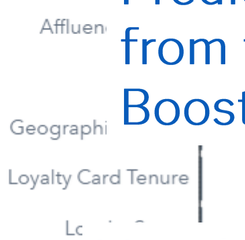
Tree



Average



Variable Importance

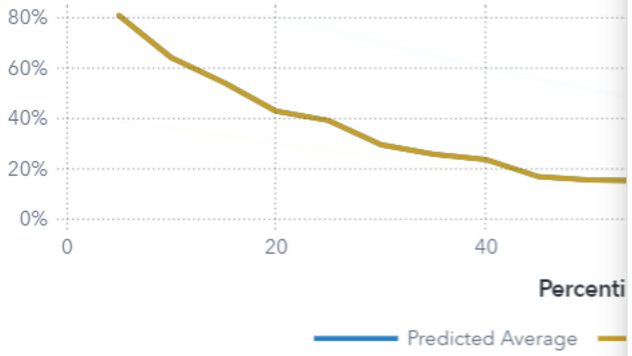


Predicted Probability from the Gradient Boosting Model

Variables you want to use for the explanation

Assessment

ProbPurchase_GradBoosting



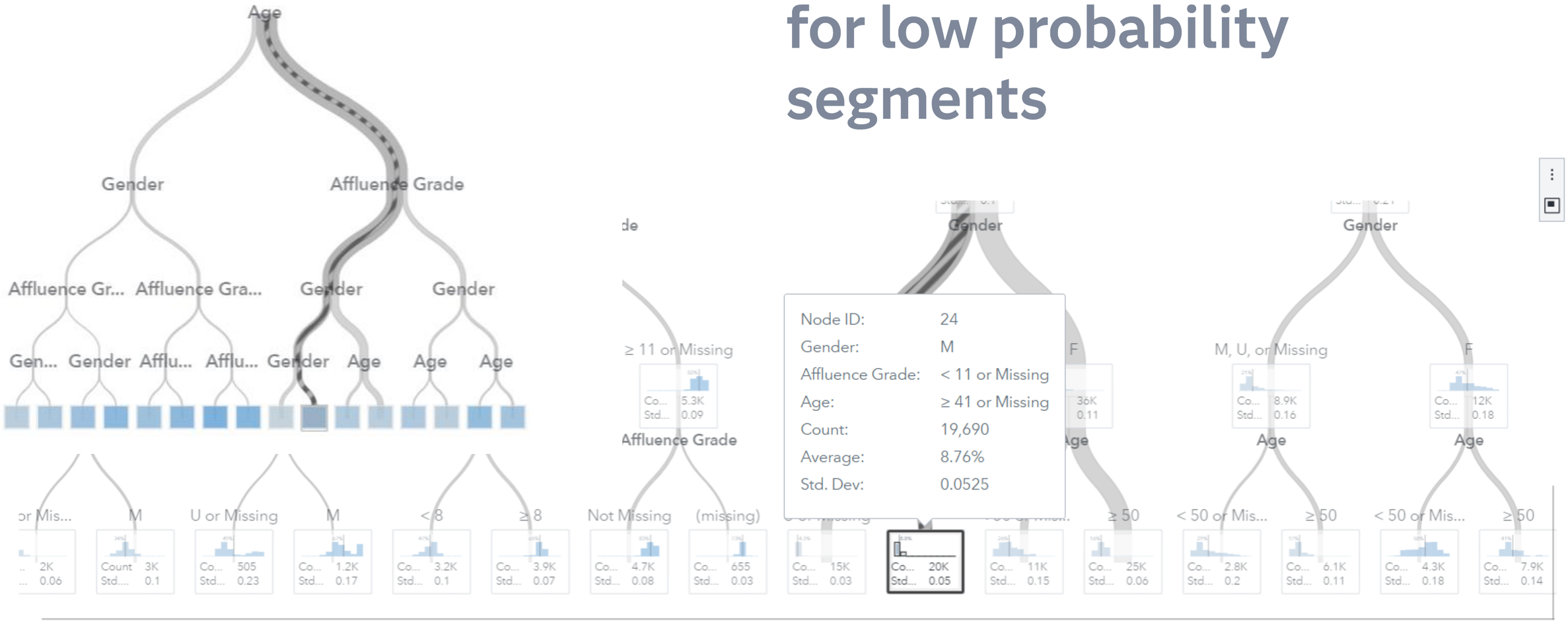
Decision tree - ProbPurchase_GradBoosti…

+ Assign data

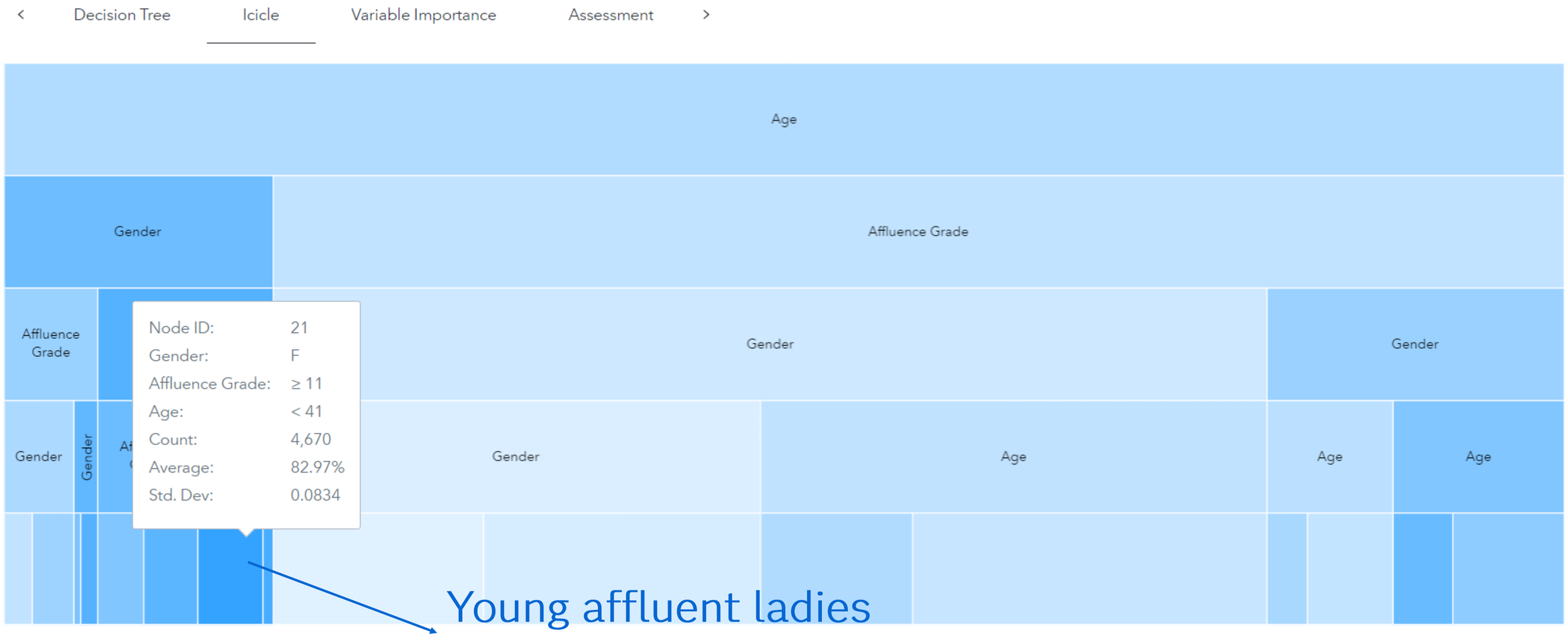
- Response + Add
  - ProbPurchase_GradBoosting
- Predictors + Add
  - Gender
  - Geographic Region
  - Loyalty Status
  - Affluence Grade
  - Age
  - Loyalty Card Tenure
  - Total Spend
- Partition ID + Add
- Frequency + Add
- Weight + Add

Review the decision tree as a whole to understand the over structure

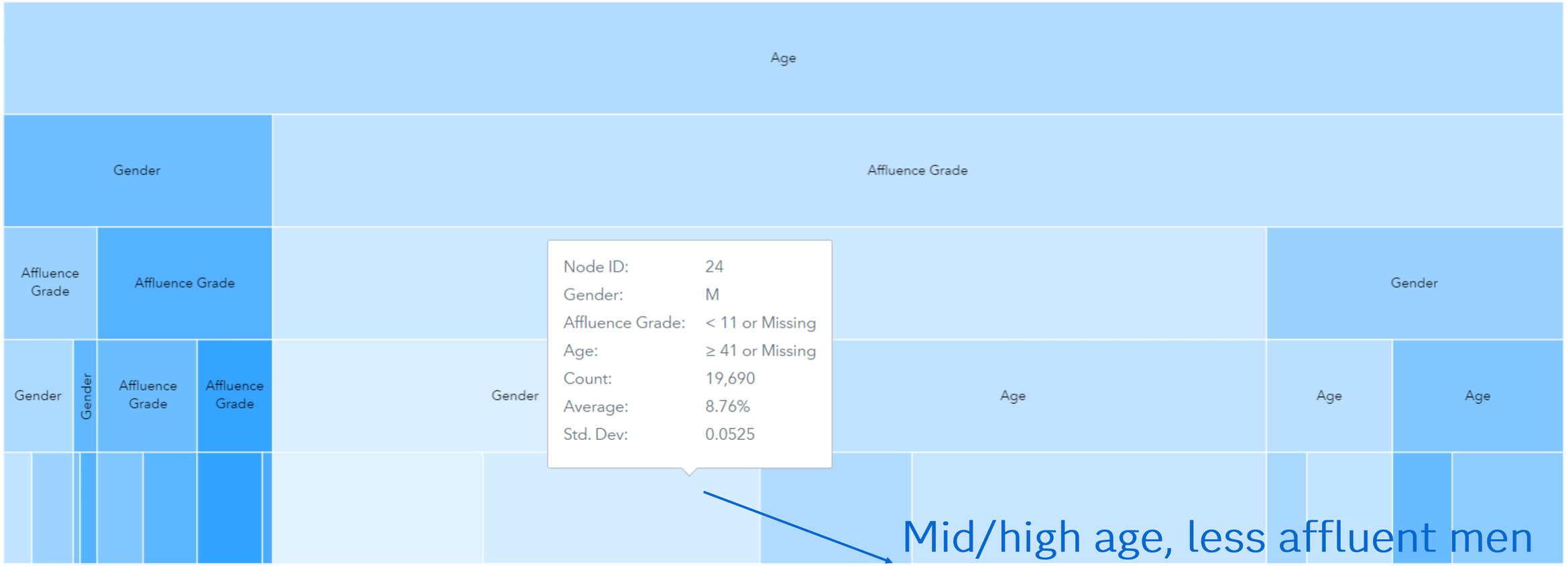
Review individual nodes to see rules for high and for low probability segments



Alternatively, review the Icicle plots (they sometimes provide a better overview)



Young affluent ladies
→ PredictedProb = 83 %



Mid/high age, less affluent men
→ PredictedProb = 83 %

Application Recommendations

- Preferred Method: Decision Tree
- Recommended SAS Tool: SAS Visual Analytics

- This is not limited to models built in SAS Visual Analytics!
- You can use the prediction of models in SAS Model Studio or Models from SAS Procedures to explain them in SAS Visual Analytics.
- You can use the TREESPLIT procedures in SAS to explain your models from PROC GRADBOOST, PROC FOREST, PROC SVMACHINE, …