SAS Viya

Trustworthy AI & Open-Source

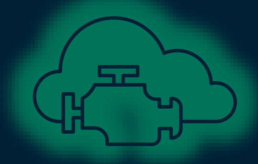
Chris Parrish, Sr. Data Scientist, Financial Services

July 2023

Steps

1. Develop Models in Open-Source IDE with Libraries of Choice
2. Import SAS Viya Libraries SWAT, SASCTL & Connect to SAS Viya
3. Register Open-Source Models to SAS Viya
4. Retrieve & Review/Edit Score Code & Create Table with Score Code
5. Load SAS Viya Trustworthy AI APIs `explainModel` `fairAITools`
6. Use Score Code & Scored Tables to Run Trustworthy AI APIs
7. Save Trustworthy AI Metrics to SAS Viya
8. Create Dashboards/Workflows to Facilitate Decisions with Metrics

Distributed Computing!



Trustworthy AI with SAS Viya & Open-Source

Open-Source Editor

1. Import OS libraries
   Develop OS Models
2. Import SWAT
   Import SASCTL

SAS Viya

2. Connect to SAS Viya
3. Register Open-Source Models
4. Retrieve Score Code
5. Load `explainModel` `fairAITools`
6. Run APIs
7. Save Trustworthy AI Metrics

8. Dashboards

Workflows

ModelOps

RStudio

1 Develop Models in Open-Source IDE with Libraries of Choice

```
######################
### Training Code ###
######################

### estimate & fit model
dm_model <- glm(as.formula(paste(dm_dec_target, " ~ .")), data=train, family=binomial(link=link))

### score full data
full <- subset(dm_inputdf, select=c(dm_dec_target, dm_input))
fullX <- subset(dm_inputdf, select=dm_input)
fully <- subset(dm_inputdf, select=dm_dec_target)
dm_scoreddf_prob_event <- data.frame(predict(dm_model, newdata = full, type = 'response'))
dm_scoreddf_prob_nonevent <- data.frame(1-predict(dm_model, newdata = full, type = 'response'))
dm_scoreddf_class <- data.frame(ifelse(dm_scoreddf_prob_event[[1]] >= avg_prob, 1, 0))
dm_scoreddf <- cbind(dm_scoreddf_prob_nonevent, dm_scoreddf_prob_event, dm_scoreddf_class)
names(dm_scoreddf) <- c(dm_predictionvar[[1]], dm_predictionvar[[2]], dm_classtarget_intovar[[1]])

### create tables with predicted values
trainProba <- data.frame(predict(dm_model, newdata = X_train, type = 'response'))
testProba <- data.frame(predict(dm_model, newdata = X_test, type = 'response'))
validProba <- data.frame(predict(dm_model, newdata = X_valid, type = 'response'))
trainData <- cbind(y_train, dm_classtarget_intovar=trainProba)
testData <- cbind(y_test, dm_classtarget_intovar=testProba)
validData <- cbind(y_valid, dm_classtarget_intovar=validProba)
names(trainData) <- c(dm_dec_target, dm_predictionvar[[2]])
names(testData) <- c(dm_dec_target, dm_predictionvar[[2]])
names(validData) <- c(dm_dec_target, dm_predictionvar[[2]])

### print model & results
summary(dm_model)
```

RStudio

2 Import SAS Viya Libraries SWAT, SASCTL & Connect to SAS Viya

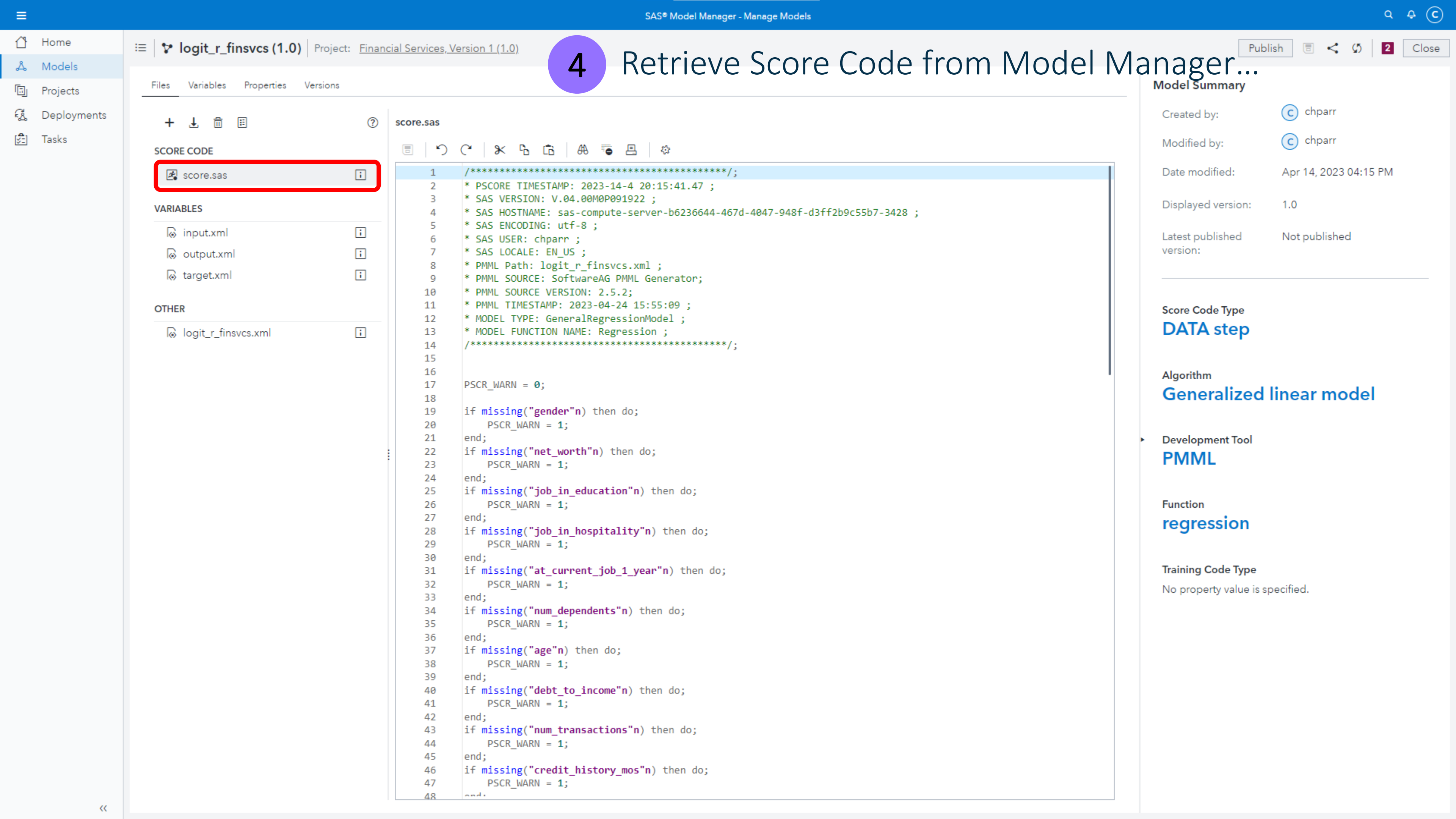
```
###################
### Environment ###
###################

library(swat)

conn <- swat::CAS(hostname=hostname_dev, port=port_dev, username, password, protocol=protocol_dev)
print(cas.builtins.serverStatus(conn))

#############################
### Identify Table in CAS ###
#############################

### caslib and table to use in modeling
caslib <- 'Public'
in_mem_tbl <- 'FINANCIAL_SERVICES_PREP'

### load table in-memory if not already exists in-memory
if (cas.table.tableExists(conn, caslib=caslib, name=in_mem_tbl)<=0) {
  cas.table.loadTable(conn, caslib=caslib, path=paste(in_mem_tbl,('.sashdat'), sep = ""),
                      casout=list(name=in_mem_tbl, caslib=caslib, promote=TRUE))}

### show table to verify
cas.table.tableInfo(conn, caslib=caslib, wildIgnore=FALSE, name=in_mem_tbl)

########################
### Create Dataframe ###
########################

dm_inputdf <- to.casDataFrame(defCasTable(conn, in_mem_tbl, caslib=caslib))
sapply(dm_inputdf, class)
```

RStudio

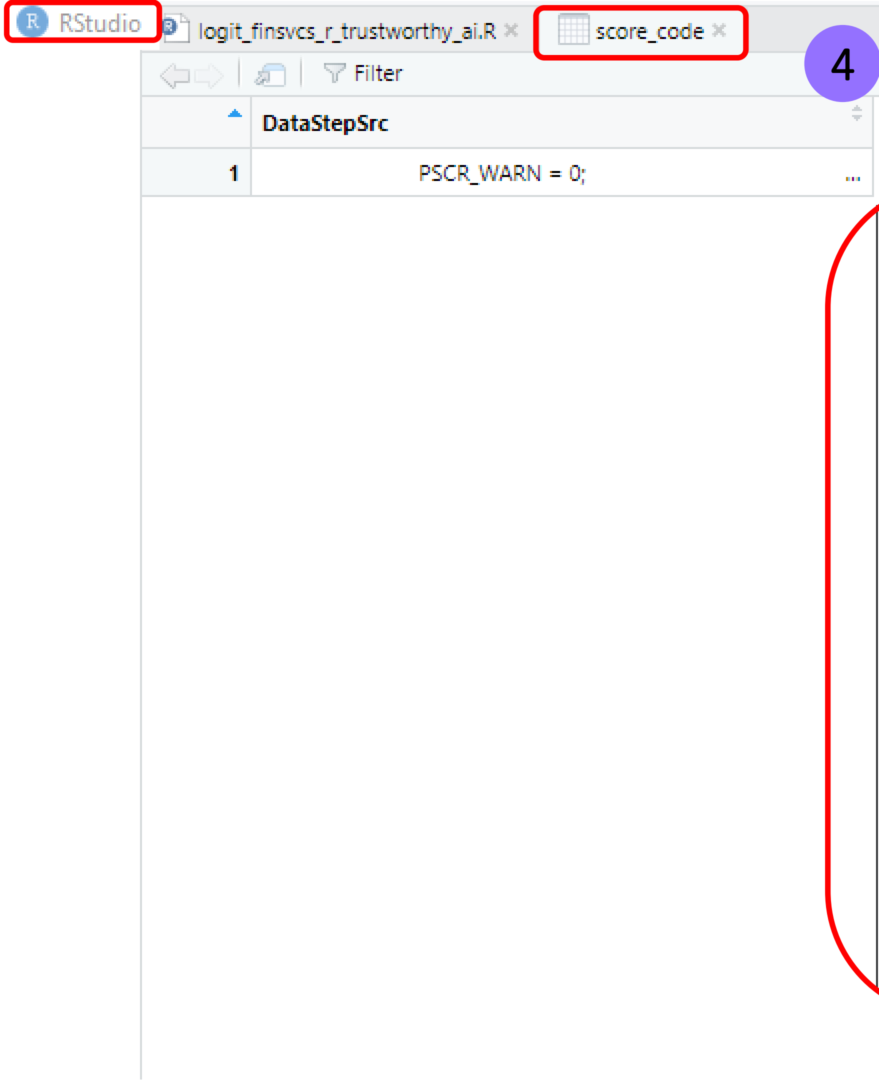
3 Register Open-Source Models to SAS Viya

```r
##################################
### Register to Model Manager ###
##################################

library(jsonlite)
library(sasctl)
library(pmml)
library(XML)
library(zip)

### define macro vars for model manager metadata script
inputData <- dm_inputdf
trainData <- train
testData <- test
targetVar <- dm_dec_target
intervalVars <- dm_input
analysisPrefix <- description
threshPredProb <- avg_prob
typeOfColumn <- as.data.frame(do.call(rbind, lapply(inputData, typeof)))
fitted.prob <- predict(dm_model, newdata = X_train, type = 'response')
trainData[[targetVar]] <- as.factor(trainData[[targetVar]])

### create directories for metadata
output_path <- file.path(output_dir, metadata_output_dir, model_name)
if (file.exists(output_path)) {
  unlink(output_path, recursive=TRUE) }

### create output path
dir.create(output_path)
analysisFolder <- paste(output_path, '/', sep = '')
jsonFolder <- paste(output_path, '/', sep = '')
zip_folder <- paste(output_path, '/', sep = '')

### create pmml (predictive model markdown language)
pmml_file <- saveXML(pmml(dm_model, model.name = model_name, description = model_type),
                     paste0(zip_folder, '/', description, '.pmml'))

### move train code and score code to zip directory
file.copy(file.path(output_dir, train_code_name), file.path(output_path, train_code_name))
file.copy(file.path(output_dir, score_code_name), file.path(output_path, score_code_name))

sess <- session(hostname_model, username=username, password=password)

rm <- register_model(
  session = sess,
  file = paste0(zip_folder, '/', description, '.pmml'),
  name = model_name,
  type = "pmml",
  project = project_name,
  force = FALSE
)
```

4 Retrieve Score Code from Model Manager…



4 …Review/Edit Score Code & Create Table with Score Code




```
PSCR_WARN = 0;

            if missing(gender) then do;
               PSCR_WARN = 1;
            end;

            if missing(net_worth) then do;
               PSCR_WARN = 1;
            end;

            if missing(job_in_education) then do;
               PSCR_WARN = 1;
            end;
            if missing(job_in_hospitality) then do;
               PSCR_WARN = 1;
            end;
            if missing(at_current_job_1_year) then do;
               PSCR_WARN = 1;
            end;
            if missing(num_dependents) then do;
               PSCR_WARN = 1;
            end;
            if missing(age) then do;
               PSCR_WARN = 1;...
```

RStudio

```
#################################################################################
### Trustworthy AI - Partial Dependence, SHAP Values, Assess Bias, Bias Mitigation ###
#################################################################################

loadActionSet(conn, 'explainModel')
loadActionSet(conn, 'fairAITools')
```

5 Load SAS Viya Trustworthy AI APIs

RStudio

6 Use Score Code & Scored Tables to Run Trustworthy AI APIs

Partial Dependency

```
cas.explainModel.partialDependence(conn,
    table=list(caslib=caslib, name=in_mem_tbl),
    seed=12345,
    modelTable=list(name=score_code_tbl),
    modelTableType="DATASTEP",
    predictedTarget=dm_predictionvar[[2]],
    analysisVariable=list(name=pd_var[[i]], nBins=20),
    inputs=dm_input,
    outputTables=list(names=list(PartialDependence=list(name='partialdependence',
                                                         replace=TRUE)))
)
```

SHAPley Values

```
cas.explainModel.shapleyExplainer(conn,
    table=list(caslib=caslib, name=in_mem_tbl),
    query=list(caslib=caslib, name=in_mem_tbl, where=query_part),
    modelTable=list(name=score_code_tbl),
    modelTableType="DATASTEP",
    predictedTarget=dm_predictionvar[[2]],
    inputs=dm_input,
    depth=1,
    outputTables=list(names=list(ShapleyValues=list(name='shapleyvalues',
                                                     caslib=caslib, replace=TRUE)))
)
```

Bias Metrics

```
cas.fairAITools.assessBias(conn,
    table = 'scored_tbl',
    modelTableType = "NONE",
    response = dm_dec_target,
    predictedVariables = list(dm_predictionvar[[1]], dm_predictionvar[[2]]),
    responseLevels = dm_classtarget_level,
    sensitiveVariable = bias_var[[i]]
)
```

RStudio

```
cas.fairAITools.mitigateBias(conn,
                        biasMetric='DEMOGRAPHICPARITY',
                        event='1',
                        learningRate='0.01',
                        maxIters='10',
                        predictedVariables=c('P_event_indicator0', 'P_event_indicator1'),
                        response='event_indicator',
                        responseLevels=c('0', '1'),
                        sensitiveVariable='gender',
                        table='financial_services_prep',
                        tolerance='0.005',
                        tuneBound='True',
                        trainProgram='
                            decisionTree.gbtreeTrain result=train_res /
                               table=table,
                               weight=weight,
                               target="event_indicator",
                               inputs= {
                                  "at_current_job_1_year", "num_dependents",
                                  "age", "amount", "credit_history_mos", "credit_score",
                                  "debt_to_income", "net_worth", "num_transactions"
                               },
                               nominals={"event_indicator"},
                               nBins=50,
                               quantileBin=True,
                               maxLevel=5,
                               maxBranch=2,
                               leafSize=5,
                               missing="USEINSEARCH",
                               minUseInSearch=1,
                               binOrder=True,
                               varImp=True,
                               mergeBin=True,
                               encodeName=True,
                               nTree=15,
                               seed=12345,
                               ridge=1,
                               savestate={
                                  name="finsvcs_gb_astore",
                                  replace=True
                               }
                            ;
                            astore.score result=score_res /
                               table=table,
                               casout=casout,
                               copyVars=copyVars,
                               rstore="finsvcs_gb_astore"
                            ;
                         ')
```

6 Use Score Code & Scored Tables to Run Trustworthy AI APIs

SAS® Data Explorer - Manage Data

Available | Data Sources | Import

Filter

Data Sources > cas-shared-default > CASUSER(chparr)

- LOGIT_R_FINSVCS_BIAS_METRICS
  05/15/23 10:30 AM • chparr
- LOGIT_R_FINSVCS_GROUP_METRICS
  05/15/23 10:30 AM • chparr
- LOGIT_R_FINSVCS_MAX_DIFFERENCES
  05/15/23 10:30 AM • chparr
- LOGIT_R_FINSVCS_PARTIAL_DEPENDENCE
  05/12/23 04:23 PM • chparr
- LOGIT_R_FINSVCS_SHAPLEY_COLS
  05/11/23 04:25 PM • chparr
- LOGIT_R_FINSVCS_SHAPLEY_ROWS
  05/11/23 04:25 PM • chparr

7 Save Trustworthy AI Metrics to SAS Viya

LOGIT_R_FINSVCS_BIAS_METRICS

Actions

Details | Sample Data

Sample rows: 100

BiasMetrics.Metric	BiasMetrics.Metric…	BiasMetrics.Value	BiasMetrics.Base	BiasMetrics.Comp…	BiasMetrics.Note	bias_var
DemographicParity	Demographic Parity (…	0.2700138592	1	0		gender
PredictiveParity	Predictive Parity	0.2608617321	1	0		gender
EqualAccuracy	Equal Accuracy	0.0104854922	1	0		gender
EqualizedOdds	Equalized Odds	0.1143410853	1	0	Max FPR difference is …	gender
EqualOpportunity	Equal Opportunity	0.0221152282	1	0		gender
DemographicParity	Demographic Parity (…	0.5095753114	1	0		at_current_job_1_year
PredictiveParity	Predictive Parity	0.4885545495	1	0		at_current_job_1_year
EqualAccuracy	Equal Accuracy	0.033361479	1	0		at_current_job_1_year
EqualizedOdds	Equalized Odds	0.3402771202	1	0	Max FPR difference is …	at_current_job_1_year
EqualOpportunity	Equal Opportunity	0.0683770024	1	0		at_current_job_1_year
DemographicParity	Demographic Parity (…	0.5532470789	0	1		job_in_education
PredictiveParity	Predictive Parity	0.5405194138	0	1		job_in_education
EqualAccuracy	Equal Accuracy	0.0132791736	1	0		job_in_education
EqualizedOdds	Equalized Odds	0.0921896075	0	1	Max FPR difference is …	job_in_education
EqualOpportunity	Equal Opportunity	0.058453901	0	1		job_in_education
DemographicParity	Demographic Parity (…	0.2836018916	1	0		job_in_hospitality
PredictiveParity	Predictive Parity	0.2775080545	1	0		job_in_hospitality
EqualAccuracy	Equal Accuracy	0.0204527173	1	0		job_in_hospitality
EqualizedOdds	Equalized Odds	0.1980191087	1	0	Max FPR difference is …	job_in_hospitality
EqualOpportunity	Equal Opportunity	0.0255460822	1	0		job_in_hospitality
DemographicParity	Demographic Parity (…	0.6381466046	7	0		num_dependents
PredictiveParity	Predictive Parity	0.6307643342	7	0		num_dependents
EqualAccuracy	Equal Accuracy	0.0719900703	7	2		num_dependents
EqualizedOdds	Equalized Odds	0.25467202	3	0	Max FPR difference is …	num_dependents
EqualOpportunity	Equal Opportunity	0.0985639687	7	0		num_dependents

trustworthy_ai_open_source

partial_dependence_r_logit | shapley_local_r_logit | shapley_dist_r_logit | assess_bias_r_logit

8 Create Dashboards/Workflows to Facilitate Decisions with Metrics

Input Variable

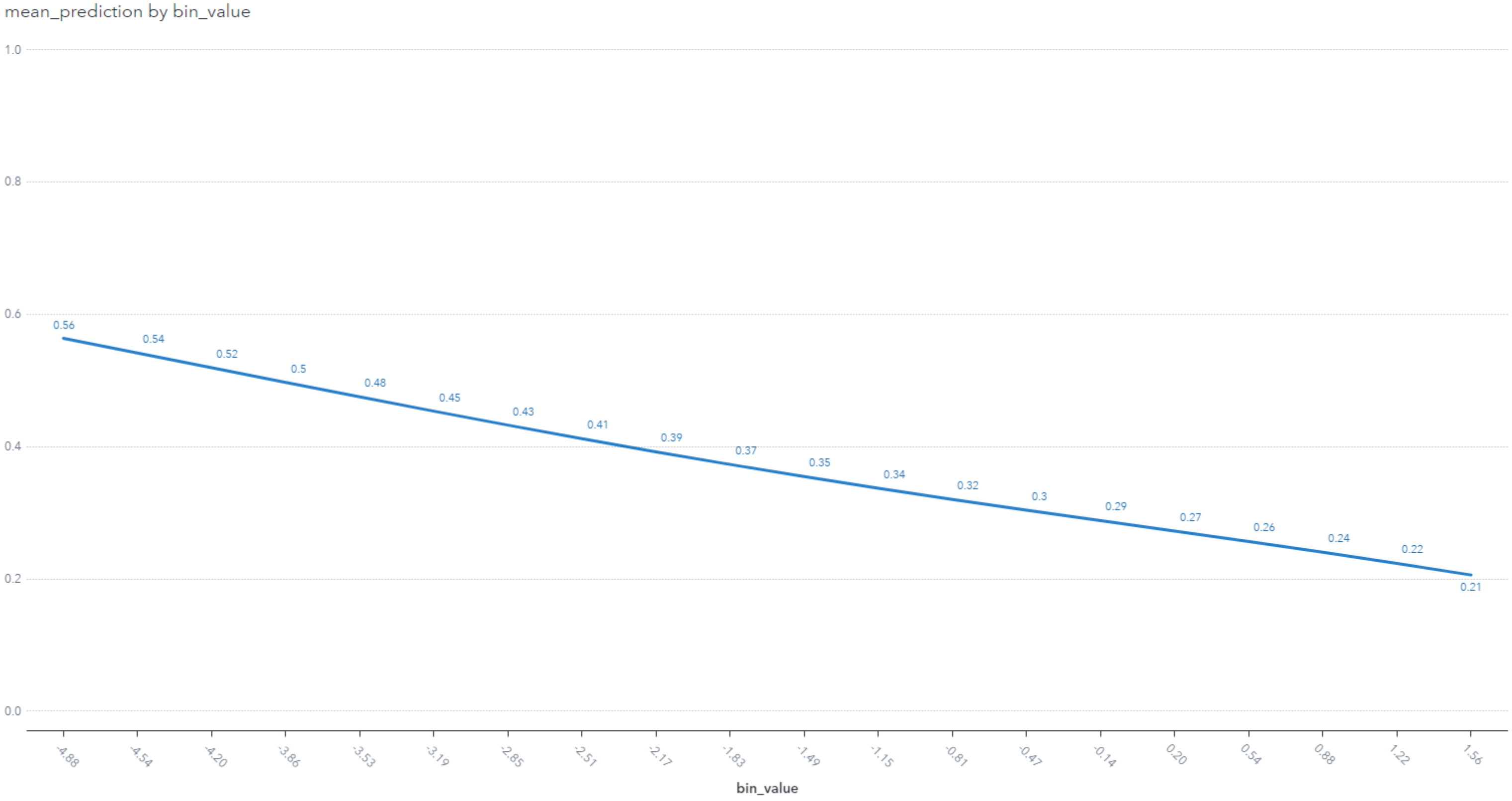
credit_score

The partial dependency plot shows how the change in the average prediction changes as the input variable changes.

Thus, PD plots show how changes to the values of a model input affect the model's predictions.

PD plots can also visualize how a potentially sensitive variable can impact the average target event value.

For example, if the plot includes a protected characteristic (or its proxy), and the average prediction decreases as the input value increases, it could imply that the higher the value of the protected characteristic then the lower the probability of the event. This could be problematic if the event includes a desired outcome for the customer and the input variable measures the increasing extent of the customer's protected characteristic (or its proxy).



SAS® Visual Analytics - Explore and Visualize

trustworthy_ai_open_source

partial_dependence_r_logit | shapley_local_r_logit | shapley_dist_r_logit | assess_bias_r_logit

8 Create Dashboards/Workflows to Facilitate Decisions with Metrics

Account ID

432

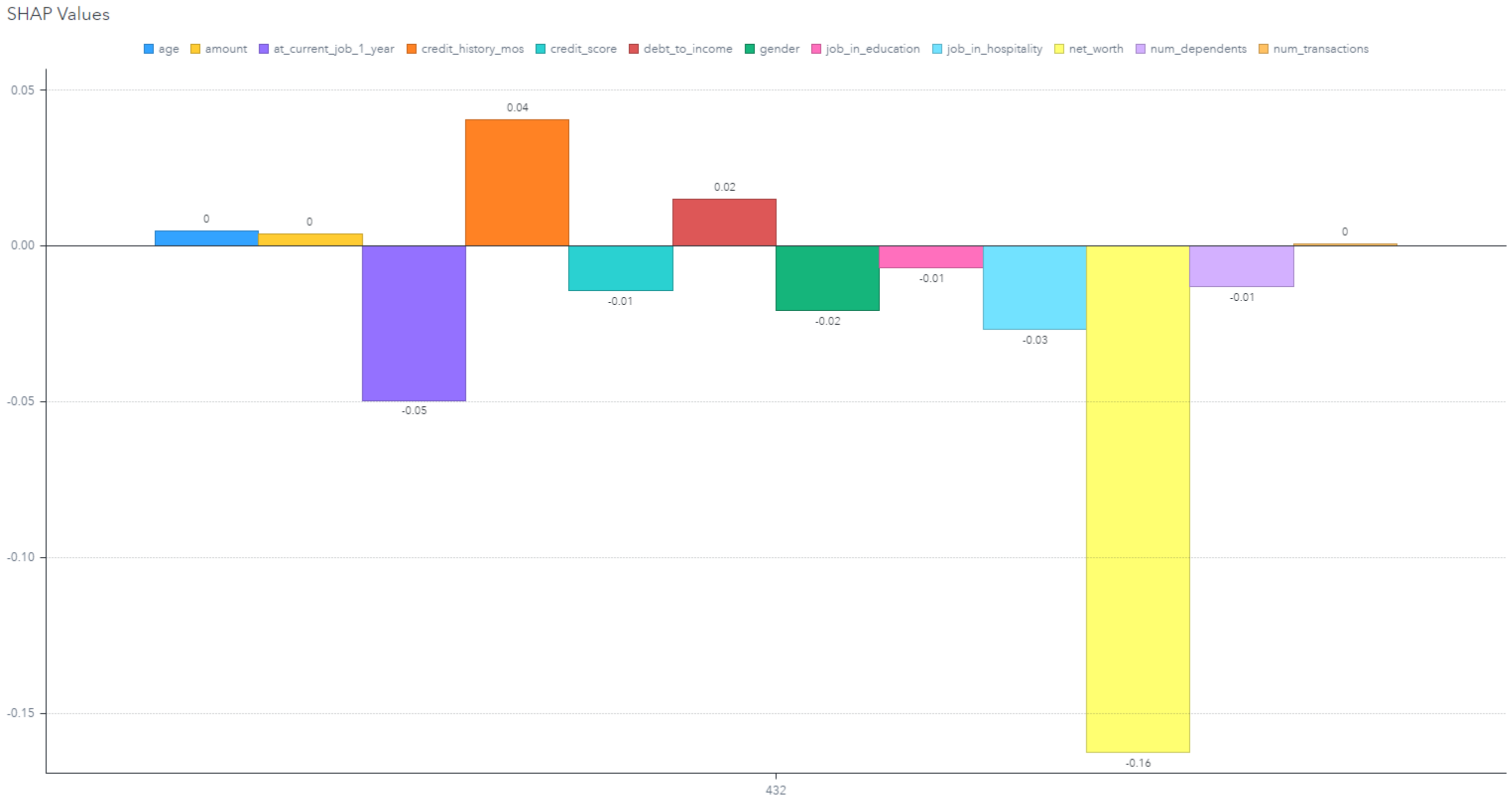
SHAPley values are a measure of variable importance for a single observation.

They are an additive measure of importance, the sum of which equals the predicted output from the model that is being explaining.

Charting HyperSHAP values can help visualize how a potentially sensitive variable can impact the predicted probability of an individual customer.

For example, if the largest (absolute) value in the SHAP Values chart is -0.32, then it decreases the predicted probability of the event by 0.32.

And if this variable was a sensitive variable (e.g., a protected characteristic or its proxy), then it could imply discriminatory behavior if the customer was declined a product based on this sensitive variable.



SAS® Visual Analytics - Explore and Visualize

trustworthy_ai_open_source

partial_dependence_r_logit | shapley_local_r_logit | shapley_dist_r_logit | assess_bias_r_logit

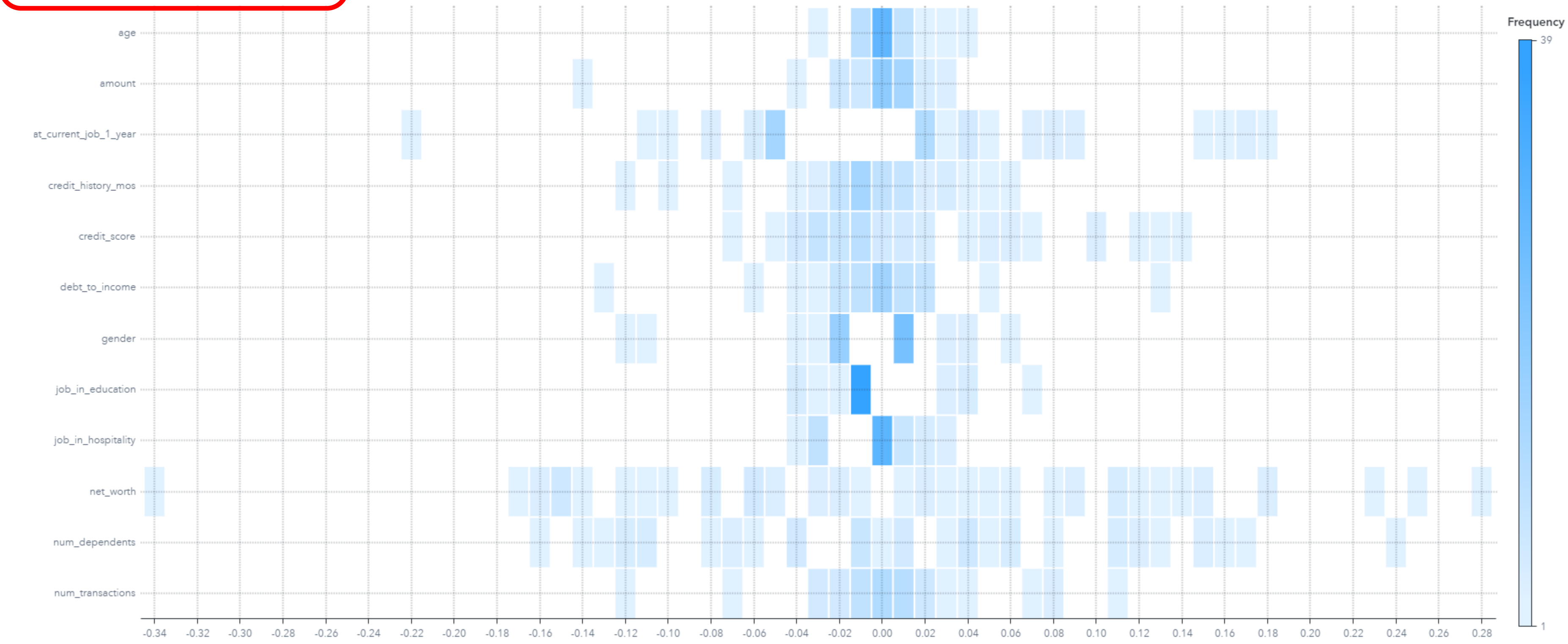
Input Variable

Variable

Distribution of SHAP Value by Selected Input

8

Create Dashboards/Workflows to Facilitate Decisions with Metrics



SAS® Visual Analytics - Explore and Visualize

trustworthy_ai_open_source

partial_dependence_r_logit shapley_local_r_logit shapley_dist_r_logit assess_bias_r_logit

8 Create Dashboards/Workflows to Facilitate Decisions with Metrics

Average Prediction for Event

The maximum differences plot shows the differences in the selected metric in the drop-down box, which may indicate the extent of bias between the base and comparison values of each variable. The base and comparison values can be viewed by hovering over each variable.

Max. Difference in Metric by Variable Assessed for Bias