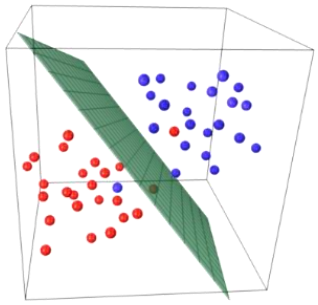
How to Decompose Variable Importance from a Support Vector Machine



By Randy Collica

Sr. Solutions Architect, SAS Institute

11-Dec-15

This note shows how you can use SAS® Enterprise Miner 14.1 to develop a Support Vector Machine (SVM) model and decompose the variables that the SVM used with variable importance! As with Artificial Neural Networks (ANN), the analysis performed is like a black-box: the analyst has a difficult time understanding what the algorithm did and what variables it used to create the output. In this note, I'll show you and point to the references where I obtained this information and you can use this in SAS with a little code writing along with the SAS® Enterprise Miner interface. The data set I'm using in this example is airline performance data set from 2007-2008 time period and the "target" variable is a binary flag that indicates if the aircraft required maintenance or not: 1= yes, 0=no. The idea is to see what set of variables contributed to this target. Now, at the outset, one might say, why not just use a straight Decision Tree and thereby get the variable importance. True, I could have done that, however, the SVM model actually did better than a single tree model and even Gradient Boosting tree model as well.

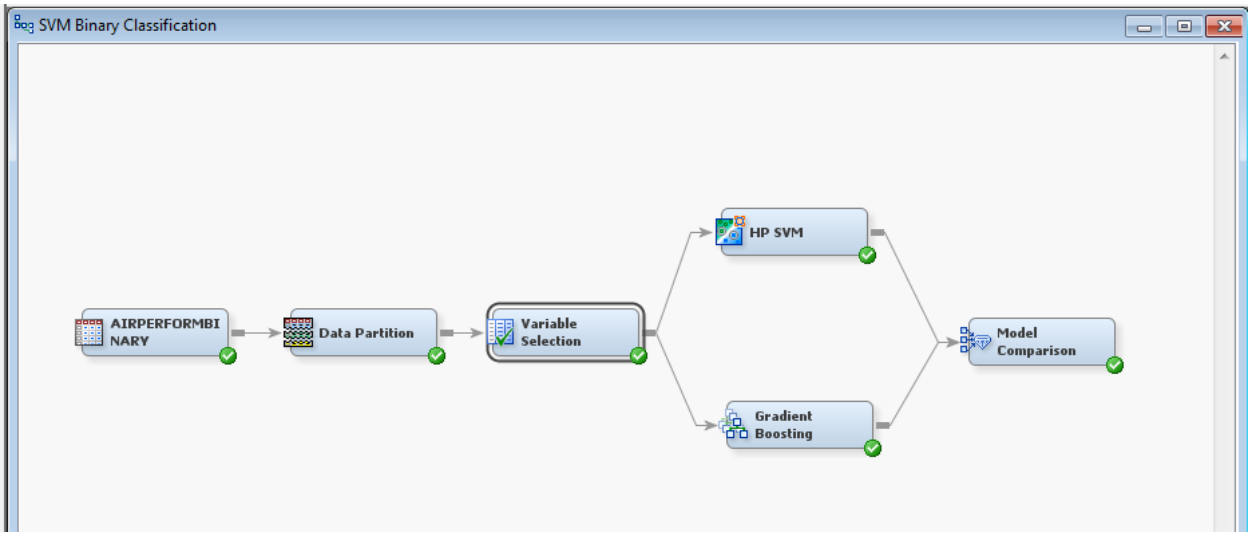
The set of variables listed in this data set are:

Variable Name	Type	Label
Customer	Character	Customer
ESN_install	Character	ESN_install
TemperatureMarginRemaining	Numeric	TemperatureMarginRemaining
TimeSinceRepair	Numeric	TimeSinceRepair
CyclesSinceRepair	Numeric	CyclesSinceRepair
TimeSinceRepair_exponent	Numeric	TimeSinceRepair_exponent
TimeSinceWash	Numeric	TimeSinceWash
CyclesSinceWash	Numeric	CyclesSinceWash
TimeSinceWash_exponent	Numeric	TimeSinceWash_exponent
RelativePowerIndication	Numeric	RelativePowerIndication
TemperatureMsmtAdjusted	Numeric	TemperatureMsmtAdjusted
AircraftAltitude	Numeric	AircraftAltitude
AircraftMach	Numeric	AircraftMach
OutsideTotalAirTemperature	Numeric	OutsideTotalAirTemperature
FlightDurationGroup	Character	FlightDurationGroup

WashNum	Numeric	WashNum
EngineSerialNumber	Numeric	EngineSerialNumber
InstallDate	Date	InstallDate
FlightDurationHours	Numeric	FlightDurationHours
maintdate	Date	Maintenance Date
date	Date	Flight Date
Target	Character	1 = Yes, 0 = No

So, the SAS Enterprise Miner diagram built to predict the target variable ***Target*** is shown in Figure (1) below.

Figure (1) SAS® Enterprise Miner Diagram for SVM Model



The Model comparison used was the ROC statistic on the Test hold out sample which was 20% stratified random sample in the Data Partition node. The Model Comparison's results is shown in Figure (2) below. The SVM did slightly better than a Gradient Boosting model on the same set of variables from the Variable Selection node.

Figure (2) Model Comparison Node Results

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Test: Roc Index
Y	HPSVM	HPSVM	HP SVM	Target		0.971
	Boost	Boost	Gradient Bo...	Target		0.905

The Variable Selection node was used to select only the variable that have the highest association to the target variable. This is shown in Figure (3) below. Only 8 variables were selected from a total of 18. The AOV16 indicates that the numeric variables were binned into 16 bins due to the non-linear nature of these variables have with respect to the target.

Figure (3) Variable Selection Node Results – Eight Variables Selected

Variable Selection

Variable Name	Role ▲	Measurement Level	Type	Label	Reasons for Rejection
AOV16_AircraftAltitude	Input	Ordinal	Numeric		
AOV16_AircraftMach	Input	Ordinal	Numeric		
AOV16_FlightDurationHours	Input	Ordinal	Numeric		
AOV16_OutsideTotalAirTemperature	Input	Ordinal	Numeric		
AOV16_RelativePowerIndication	Input	Ordinal	Numeric		
AOV16_TemperatureMarginRemain...	Input	Ordinal	Numeric		
AOV16_TemperatureMsmtAdjusted	Input	Ordinal	Numeric		
AOV16_TimeSinceRepair_exponent	Input	Ordinal	Numeric		
AircraftAltitude	Rejected	Interval	Numeric	AircraftAltitude	Varsel:Small R-square value, AO...
AircraftMach	Rejected	Interval	Numeric	AircraftMach	Varsel:Small R-square value, AO...
Customer	Rejected	Nominal	Character	Customer	Varsel:Small R-square value
CyclesSinceRepair	Rejected	Interval	Numeric	CyclesSinceR...	Varsel:Small R-square value
CyclesSinceWash	Rejected	Interval	Numeric	CyclesSinceW...	Varsel:Small R-square value
ESN_install	Rejected	Nominal	Character	ESN_install	Varsel:Small R-square value
EngineSerialNumber	Rejected	Interval	Numeric	EngineSerialN...	Varsel:Small R-square value
FlightDurationGroup	Rejected	Nominal	Character	FlightDuration...	Varsel:Small R-square value
FlightDurationHours	Rejected	Interval	Numeric	FlightDuration...	Varsel:Small R-square value, AO...
OutsideTotalAirTemperature	Rejected	Interval	Numeric	OutsideTotalAi...	Varsel:Small R-square value, AO...
RelativePowerIndication	Rejected	Interval	Numeric	RelativePowerI...	Varsel:Small R-square value, AO...
TemperatureMarginRemaining	Rejected	Interval	Numeric	TemperatureM...	Varsel:AOV16 variable preferred
TemperatureMsmtAdjusted	Rejected	Interval	Numeric	TemperatureM...	Varsel:Small R-square value, AO...
TimeSinceRepair	Rejected	Interval	Numeric	TimeSinceRep...	Varsel:Small R-square value
TimeSinceRepair_exponent	Rejected	Interval	Numeric	TimeSinceRep...	Varsel:Small R-square value, AO...
TimeSinceWash	Rejected	Interval	Numeric	TimeSinceWash	Varsel:Small R-square value
TimeSinceWash_exponent	Rejected	Interval	Numeric	TimeSinceWa...	Varsel:Small R-square value
WashNum	Rejected	Interval	Numeric	WashNum	Varsel:Small R-square value

Now the fun part comes in in order to estimate the distances of each support vector to each instance in the training data. You could use a SAS Code node in Enterprise Miner, however, I choose to use Enterprise Guide so I could also generate a nice chart to represent the variable importance that I could customize as well.

Using specific features in the SVM property sheet of Enterprise Miner, I needed to select *Active Set* for the optimization method in the SVM node. This will produce a data set in the diagram workspace folder called HPSVM_SVECTORS.

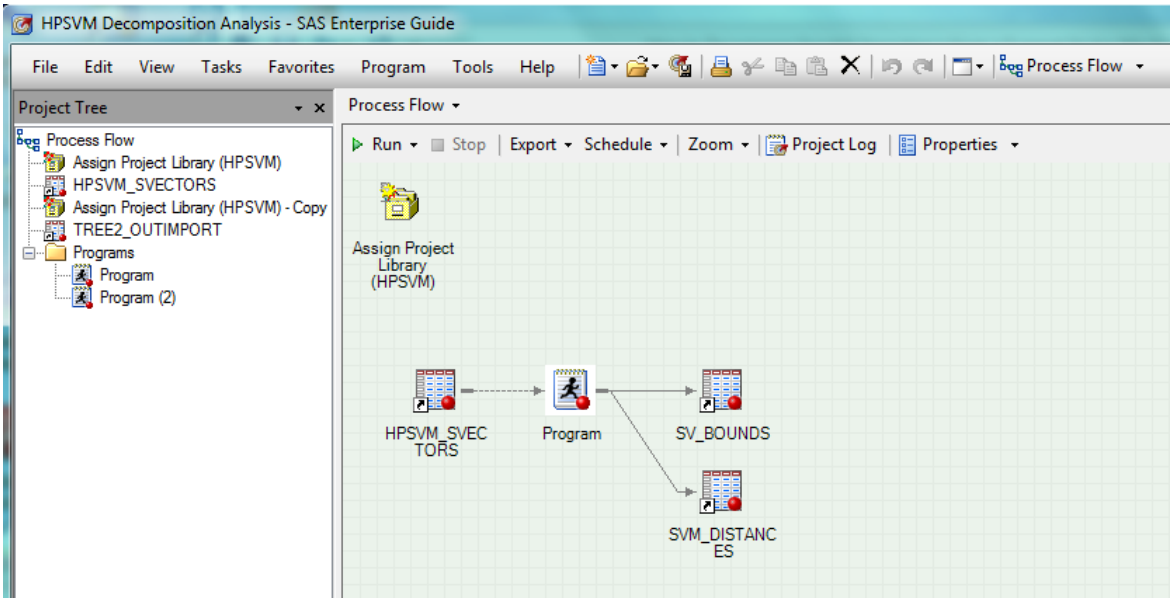
.. Property	Value
General	
Node ID	HPSVM
Imported Data	
Exported Data	
Notes	
Train	
Variables	
Maximum Iterations	25
Use Missing as Level	No
Tolerance	1.0E-6
Penalty	1.0
Optimization Method	
Optimization Method	Active Set
Interior Point Options	
Active Set Options	

Now, the SAS code used in Enterprise Guide to compute the distances from each support vector to each variable's instances. This is accomplished using the decision function equation:

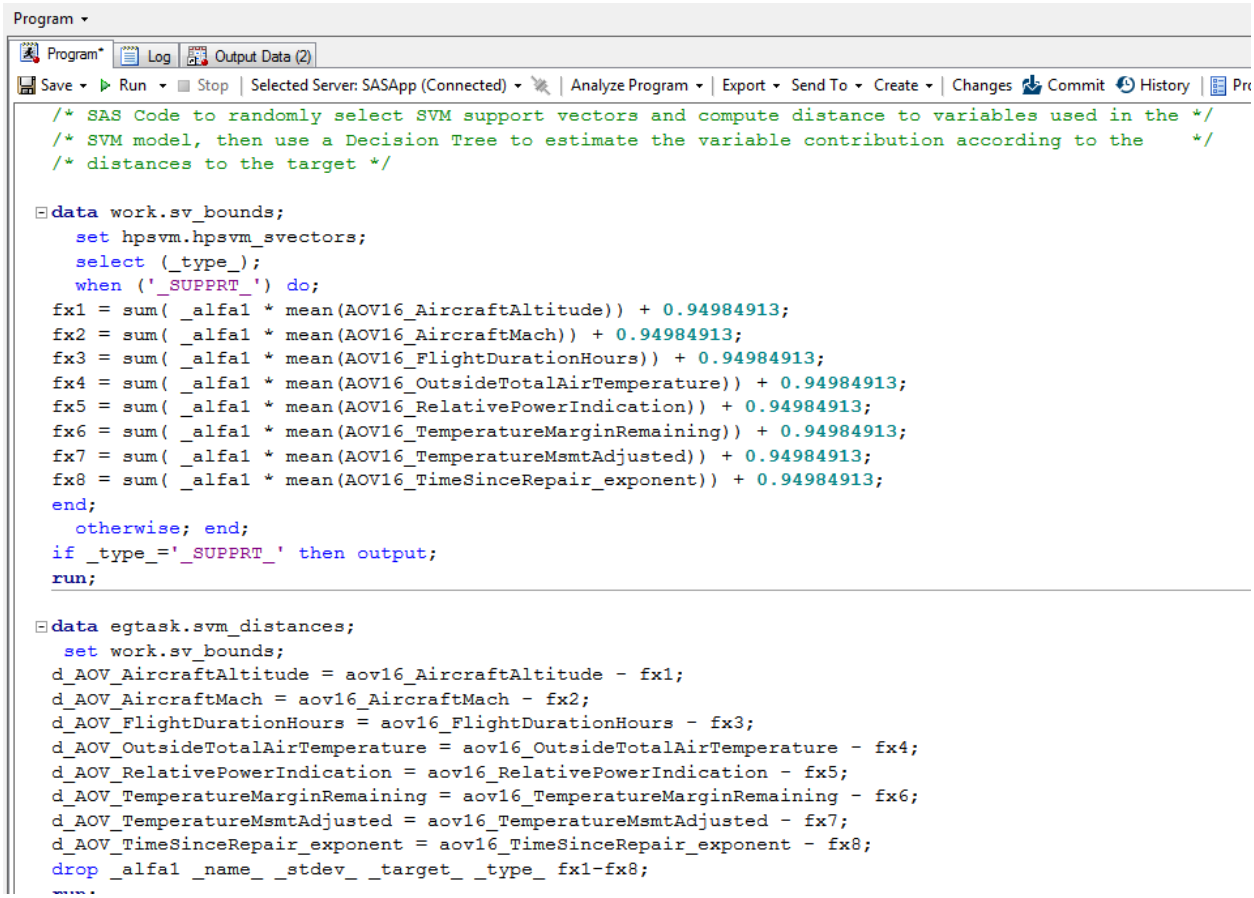
$$f\left(x_j\right) = \sum_{k=1}^{n} \left(K\left(x_j, z_k\right) y_k \right) \cdot \alpha_k + \beta$$

The Enterprise Guide diagram and code is shown in Figure (4) below. The code averages each variable's support instances and then the second data set computes the distance from the decision function equation. A macro could have been developed that does this all in one step of course, but this does show you more on how the computation works.

Figure (4) Enterprise Guide Diagram and Code to Compute Vector Distances

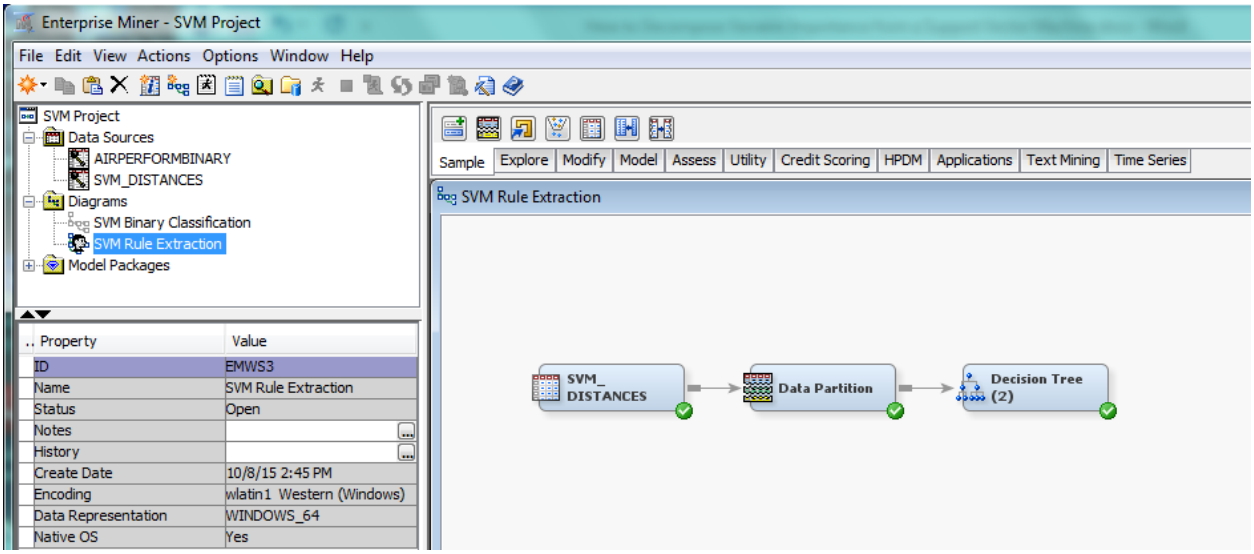


And the Program code is:



Now, we take the distances which I saved in a SAS library called EGTASK on my server and called the data set SVM_DISTANCES. This data set I need to run a Decision Tree model to get the variable importance of these distances to the target variable. The target variable in this case is the variable called _MEAN_ which are 1 and -1 that represent the target levels 0 and 1 respectively. The Enterprise Miner diagram for the Decision Tree model is shown in Figure (5a).

Figure (5a) Enterprise Miner Diagram for Decision Tree Model on Distances Data Set



In the Enterprise Guide I now make a SAS project library to the Decision Tree diagram and create the bar chart of variable importance using the data set from Enterprise Miner called TREE_OUTIMPORT (in this case I had another decision tree so mine is called TREE2 instead of TREE). The SAS code used to generate the bar chart and the bar chart plot is shown in Figures (6) and (7) respectively.

Figure (5b) Enterprise Guide Flow for Variable Importance Chart

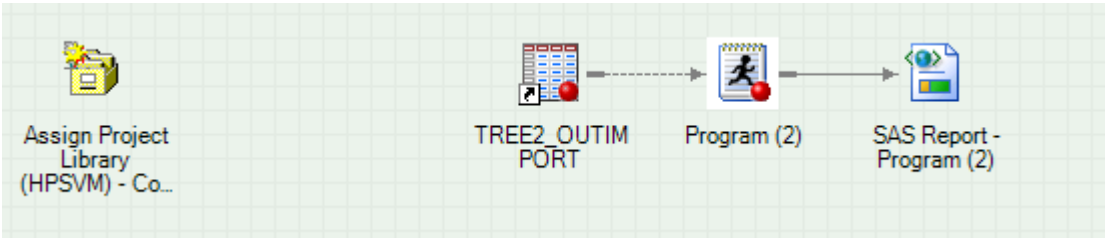
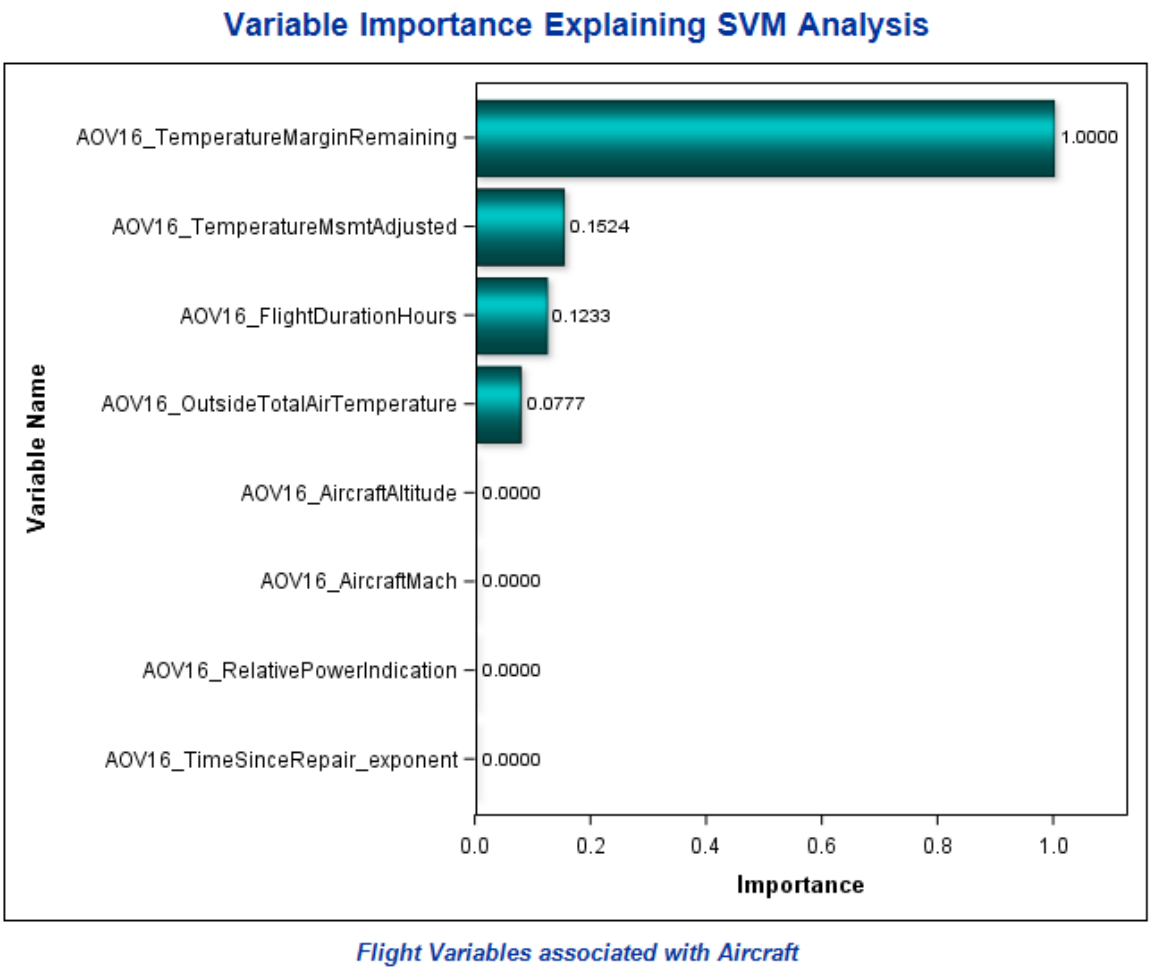


Figure (6) SAS Code to Generate Variable Importance Bar Chart

```
Program (2)
Program | Log | Results
Save | Run | Stop | Selected Server: SASApp (Connected) | Analyze Program | Export | Send To | Create | Changes | Comr
title 'Variable Importance Explaining SVM Analysis';
footnote 'Flight Variables associated with Aircraft';
proc sgplot data=hpsvm_f.tree2_outimport;
  hbar  name /stat=mean datalabel response=importance dataskin=sheen grouporder=data
  fillattrs=(color=darkcyan) categoryorder=respdesc;
run;
title; footnote;
```

Now, the final result is the variable importance chart from the SVM's distance data set.

Figure (7) Variable Importance of SVM Model



References:

[1] B. Baesens, D. Martens, and T. Van Gestel, “Decompositional Rule Extraction from Support Vector Machines by Active Learning,” IEEE Trans. On Knowledge & Data Engineering, Vol. 21, no. 2, Feb. 2009 p. 178-191.

[2] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge Univ. Press, 2000.