esProc

Professional high-performance computing engine

Automated modeling and prediction in SPL

Issued by Raqsoft

Introduction

This PPT focuses on three aspects - environment settings for modeling software, model building and data prediction. There are detailed examples for illustrating the modeling process, data requirements, model performance information, model selection, data prediction based on model files or the selected model object, and prediction result output. A good knowledge of these lets you make the best use of the Auto-Modeling tool to perform data mining analysis in esProc.

Content

1 Preface

The flourishing internet economy has changed the business decision-making mode forever. Data, particularly the big data, has become the critical basis of making right decisions. Correct and coherent data flow is the key in making decisions fast and flexibly. In this context, business modeling emerges as AI is becoming a pressing demand.

Adhering to the design concept of "intelligent, efficient and easy to use", YModel creates an innovative process of "data - model -prediction - application". With the support of big data processing techniques and exclusive algorithm engine, it builds an intelligent and easy-to-use AI analysis and application platform to help businesses improve modeling efficiency and reduce modeling cost.

The longest journey begins with the first step. To open up a new world of big data processing, SPL Auto-Modeling is a good start.

2 Environment settings

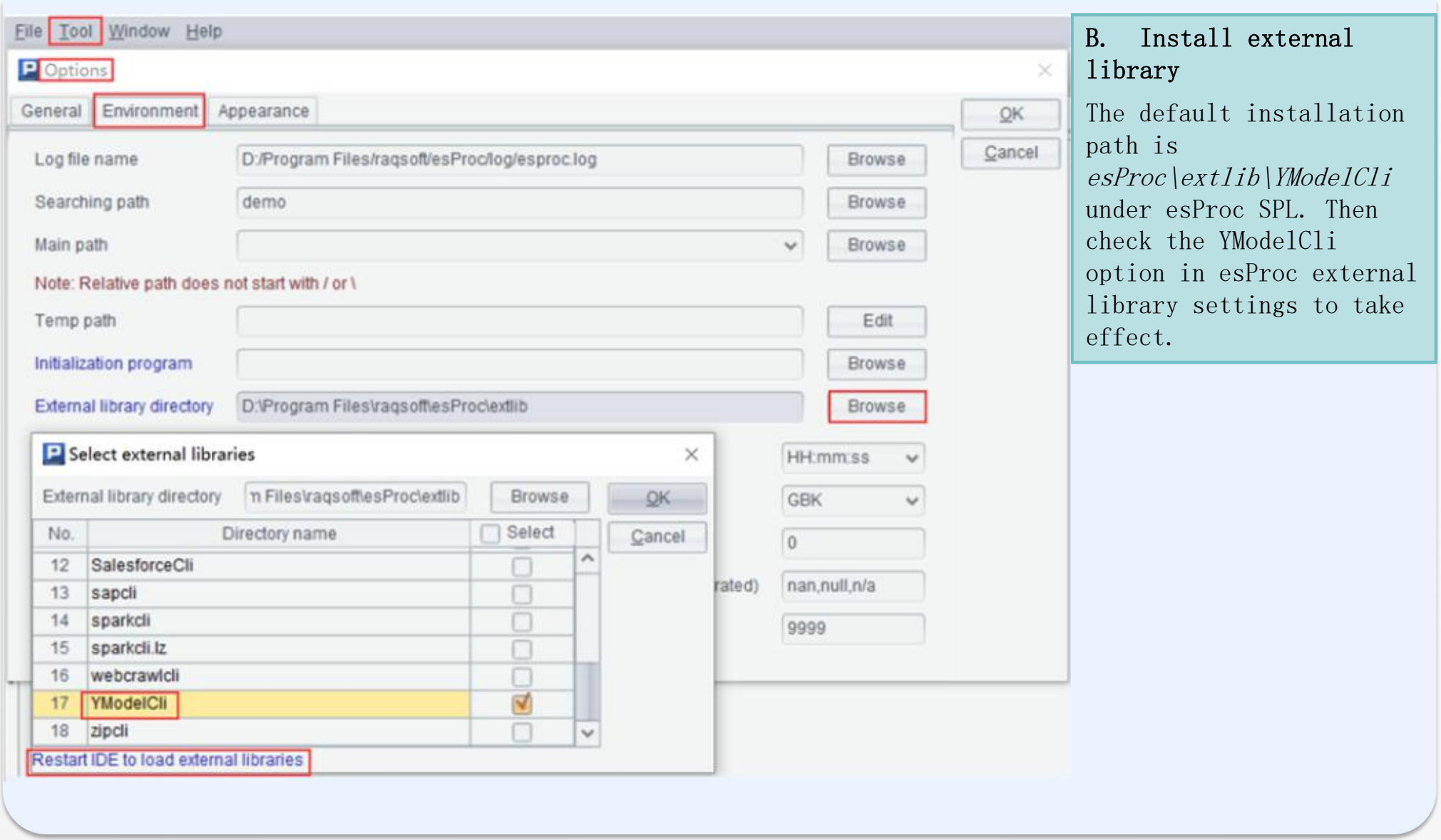
SPL Modeling is composed of YModel Auto-Modeling software and esProc SPL external library YModelCli. The two parts are connected through configuration file *userconfig.xml*.

A. Install YModel

Download YModel installation package HERE.

Install the software and record the installation directory, such as *C:\Program Files\raqsoft\ymodel*.

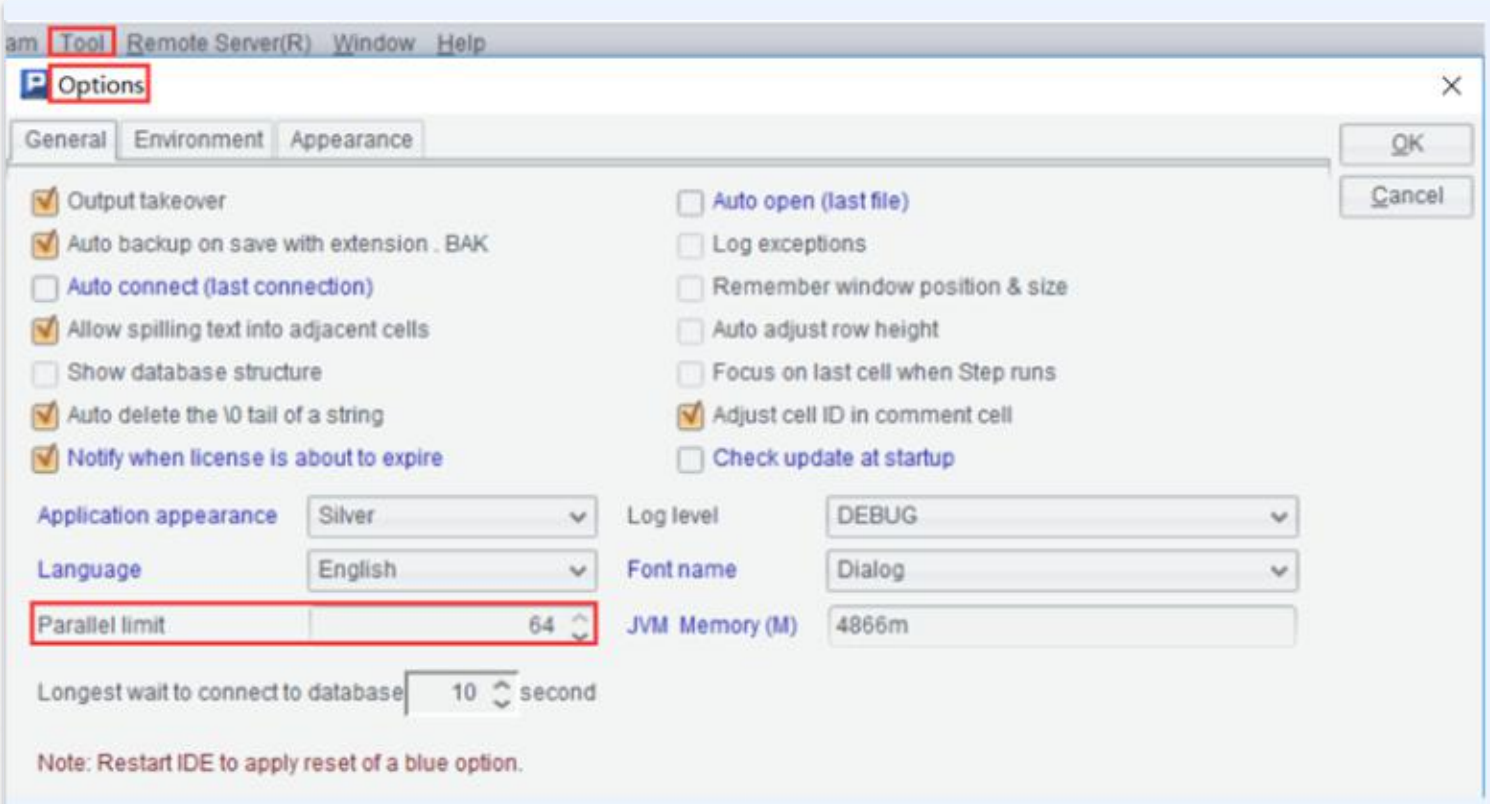
2 Environment settings



B. Install external library

The default installation path is *esProc\extlib\YModelCli* under esProc SPL. Then check the YModelCli option in esProc external library settings to take effect.

2 Environment settings



C. Set number of multithreads

You can set parallel limit as 64 for multithreaded data modeling

2 Environment settings

C. configuration file: The SPL modeling application requires appropriate parameter configurations in *userconfig.xml* file under the external library directory *esProc\extlib\YModelCli*.

Option	Name	Note
sAppHome	C:\Program Files\raqsoft\ymodel	Application directory
sLicenseFile	D:\backup\OneDrive\priv\ymodel_lic.xml	YModel license
sEsprocLicenseFile	D:\backup\OneDrive\priv\esproc_lic.xml	esProc license
sPythonHome	c:\Program Files\raqsoft\ymodel\Python37\python.exe	Python file (for Windows)
	/raqsoft/ymodel/Python37/bin/python3.7	(for Linux)
bAutoDecideImpute	true	Intelligent imputation
iPythonProcessNumber	2	Number of Python processes
iResampleMultiple	150	Resampling frequency

sAppHome is YModel installation directory.

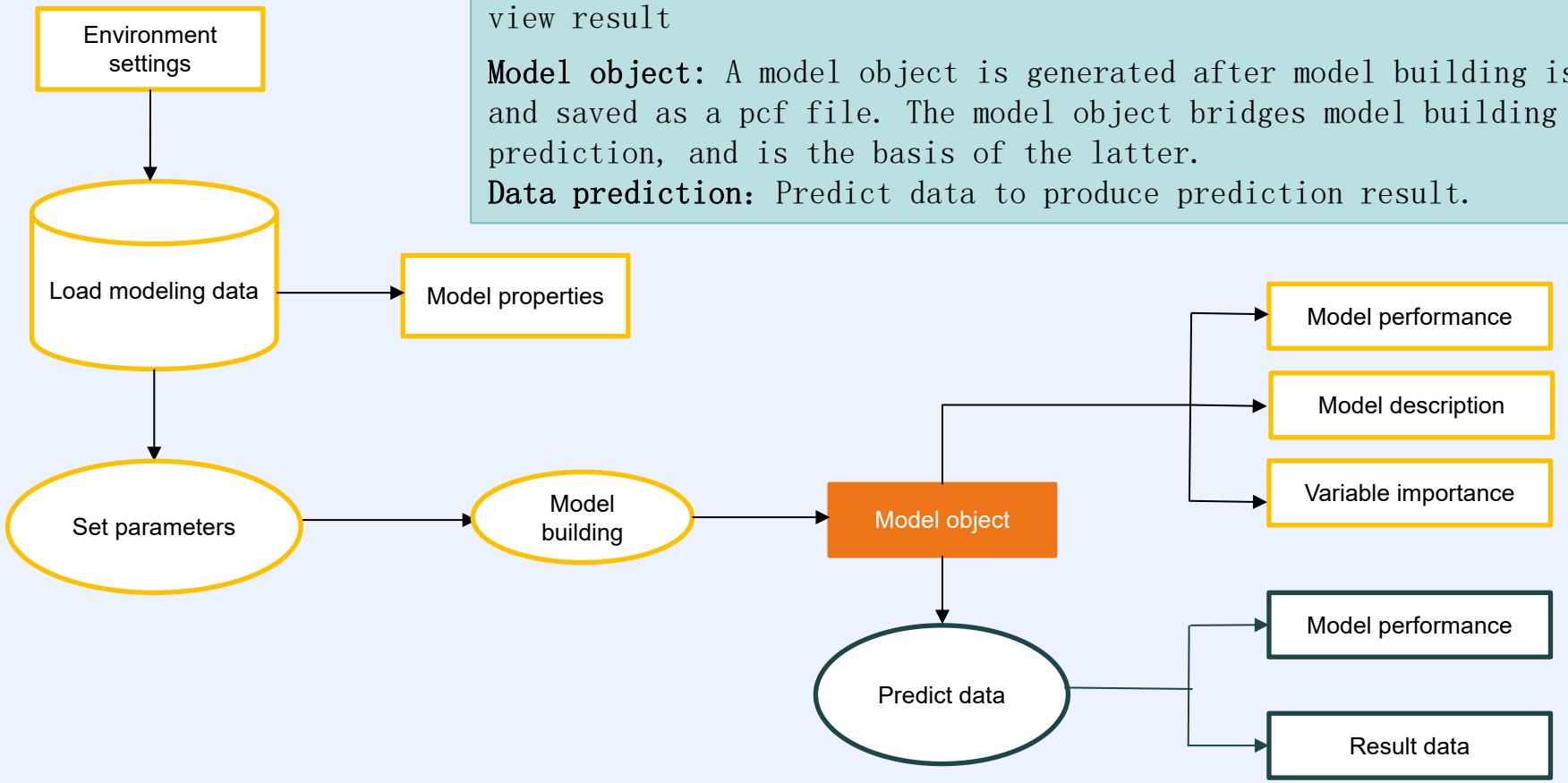
3 Data modeling

3.1 Modeling flowchart

Basic process: load data -> set parameters -> perform model building -> view result

Model object: A model object is generated after model building is finished and saved as a pcf file. The model object bridges model building and data prediction, and is the basis of the latter.

Data prediction: Predict data to produce prediction result.



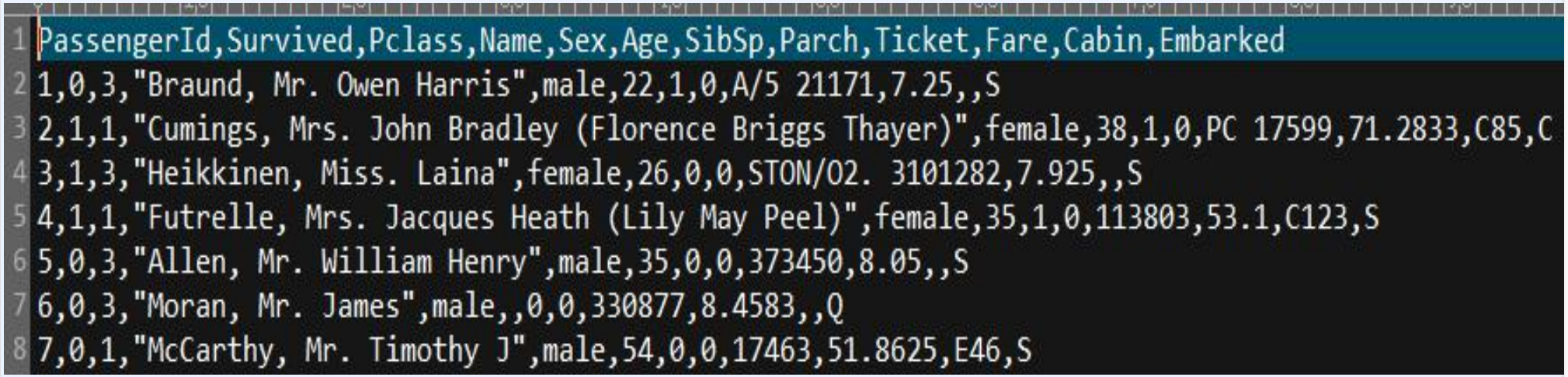
3.2 SPL modeling example

	A	Note
1	=file("train.csv").import@tqc()	Modeling data
2	=ym_env()	Initialize environment
3	=ym_model(A2, A1)	Load data
4	=ym_target(A3, "Survived")	Set target variable
5	=ym_setparam(A3, "intelligence":true, "Balance":2)	Set modeling parameters
6	=ym_statistics(A3,"Age")	Get variable properties
7	=ym_build_model(A3)	Get through the model building process
8	=ym_save_pcf(A7, "demo.pcf")	Save model object as file
9	=ym_json(A7)	Export model information as JSON string
10	=ym_importance(A7)	Get variable importance
11	=ym_present(A7)	Get model description
12	=ym_performance(A7)	Get model performance
13	>ym_close(A2)	Close YModel

The main parts of modeling process are highlighted in yellow. A8 saves the modeling result as a pcf file, which can be used to predict data; A9-A12 outputs model-related information.

3.3 Modeling data

Data used for model building must be structured data. It can come from relational databases, text files, table sequence, CSV files, etc.
Below is a CSV file:



```
1 PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked
2 1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S
3 2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C
4 3,1,3,"Heikkinen, Miss. Laina",female,26,0,0,STON/O2. 3101282,7.925,,S
5 4,1,1,"Futrelle, Mrs. Jacques Heath (Lily May Peel)",female,35,1,0,113803,53.1,C123,S
6 5,0,3,"Allen, Mr. William Henry",male,35,0,0,373450,8.05,,S
7 6,0,3,"Moran, Mr. James",male,,0,0,330877,8.4583,,Q
8 7,0,1,"McCarthy, Mr. Timothy J",male,54,0,0,17463,51.8625,E46,S
```

The first row in the file contains field name information. The other rows are records.
YModel supports access of diverse types of data source and manages all accesses in a uniform way, This ensures a broad data base and coherent data flow.
The data preprocessing covers a series of operations from missing value and high-cardinality variable handling, data smoothing, variable filtering, computed variable derivation to data cleaning such as DOC variable.

3.4 Variable properties

Name	Value
VarName	Age
Miss	0.2102728731942215
Imp	0.0
Card	0
GraphData	
GroupDescStatisticsTable	
GroupFrequencyTable	
Upquar	38.0
Median	28.0
Lwquar	21.0
Sd	14.378831499148678
Max	71.0
Min	0.75
Avg	29.78048780487805
Sk	0.3387264693285246
OuterValues	[64.0,64.0,65.0,65.0,65.0,66.0,70.5,71.0,71.0]
Pearson	NaN
Spearman	NaN
Target0	0
Target1	1
bGraphStatistics	true
bStatistics	true
bTargetStatistics	true

Return the statistical information of a specified variable, including the maximum value, minimum value, importance, missing value rate and skewness, to help perform data exploration & analysis. Take the Age variable as an example, the returned information is shown on the left:

3.5 Modeling parameter settings

Set parameters for the modeling variables. Below are descriptions of relevant parameters: (Refer to YModel JSON-style Parameter Guide for detailed rules.)

Key	Value Type	Description
balance	int	Balance parameters
Target	String	Target variable name
id	String	ID variable name
intelligence	Boolean	use intelligent-imputation or not
misformat	String	Missing value format
optimal	Boolean	Use optimal parameter configuration or not
parallel	int	Number of parallel threads for data preprocessing
resample	Boolean	Resample or not
resamplemul	int	Resampling multiple
resamplenum	int	Resampling frequency
testpercent	int	Test data Percentage(0-99)
vartypes	ArrayList< Byte>	Variable types
ModelFields	ArrayList<String>	Field name order for model building

3.6 Model information

Model information mainly includes model description, model performance and variable importance. They can be exported as JSON strings through the ym_json() interface.

A. Model description

YModel Auto-Modeling encapsulates a variety of algorithms. Algorithms used for building the current model and related model parameters will be returned.

Index	name	value	properties
1	RidgeClassification_1	0.80441281989954 56	[[random_state,0],[alpha,0.5],[m...
2	LogicClassification_1	0.8038148768237263	[[C,1.0],[random_state,0],[verbo...
3	RFClassification_1	0.7885075340827553	[[min_samples_leaf,50],[n_esti...
4	FNNClassification_1	0.7544247787610621	[[warm_start,false],[random_sta...
5	XGBClassification_1	0.8312604640038268	[[max_delta_step,0],[base_scor...
6	GBDTClassification_1	0.8166108586462568	[[min_samples_leaf,50],[learnin...
7	TreeClassification_1	0.79239416407558	[[min_samples_leaf,50],[splitter...

3.6 Model information

B. Model performance

It refers to the performance-related information, such as Gini, AUC, KS index, etc.

Index	Name	Value
1	GINI	0.66276010523798 12
2	AUC	0.8313800526189906
3	KS	0.5908873475245157
4	AccuTable	[[0.05000000074505806,0.4919786096256685,0.4...
5	RocTable	[[0.0,0.013513513513513514],[0.0,0.02702702702...
6	LiftAndRecallTable	[[1,2.5270270270270268,0.1216216216216216 3, ...

3.6 Model information

C. Variable importance

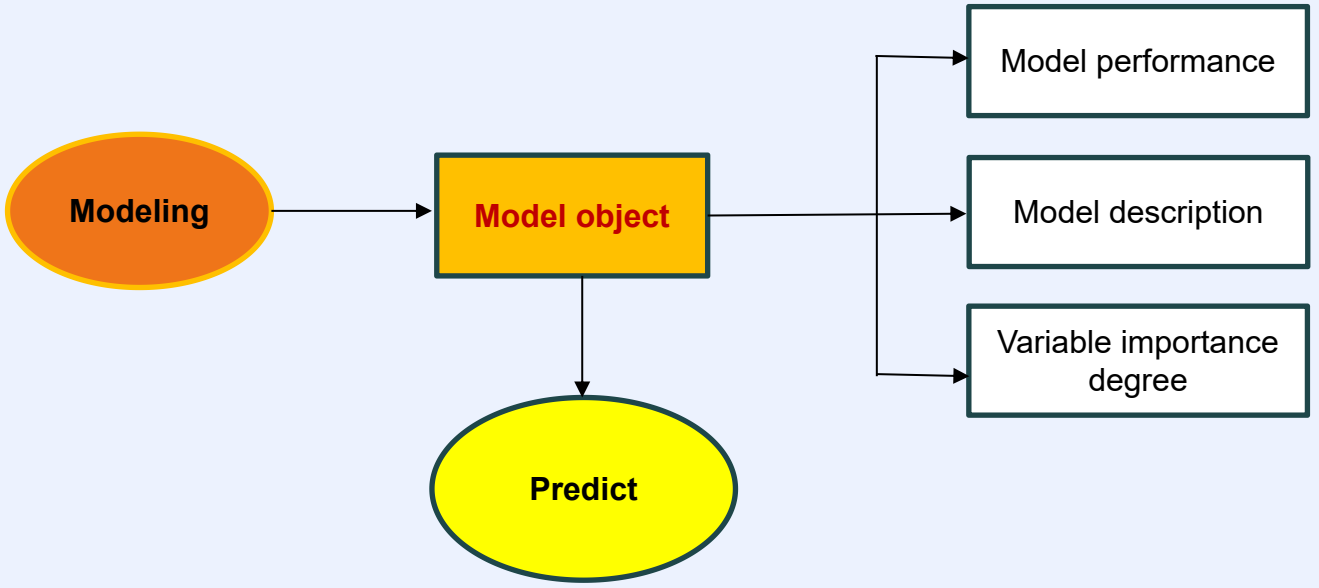
Importance of each variable.

Index	Name	Importance
1	PassengerId	0.0
2	Pclass	0.3348135805855989
3	Sex	1.0
4	Age	0.192042376847223 72
5	SibSp	0.141105179049140 55
6	Parch	0.081413168460130 69
7	Ticket	0.0
8	Fare	0.187676609894185 44
9	Cabin	0.0
10	Embarked	0.080884297469243 28
11	Survived	0.0

3.6 Model information

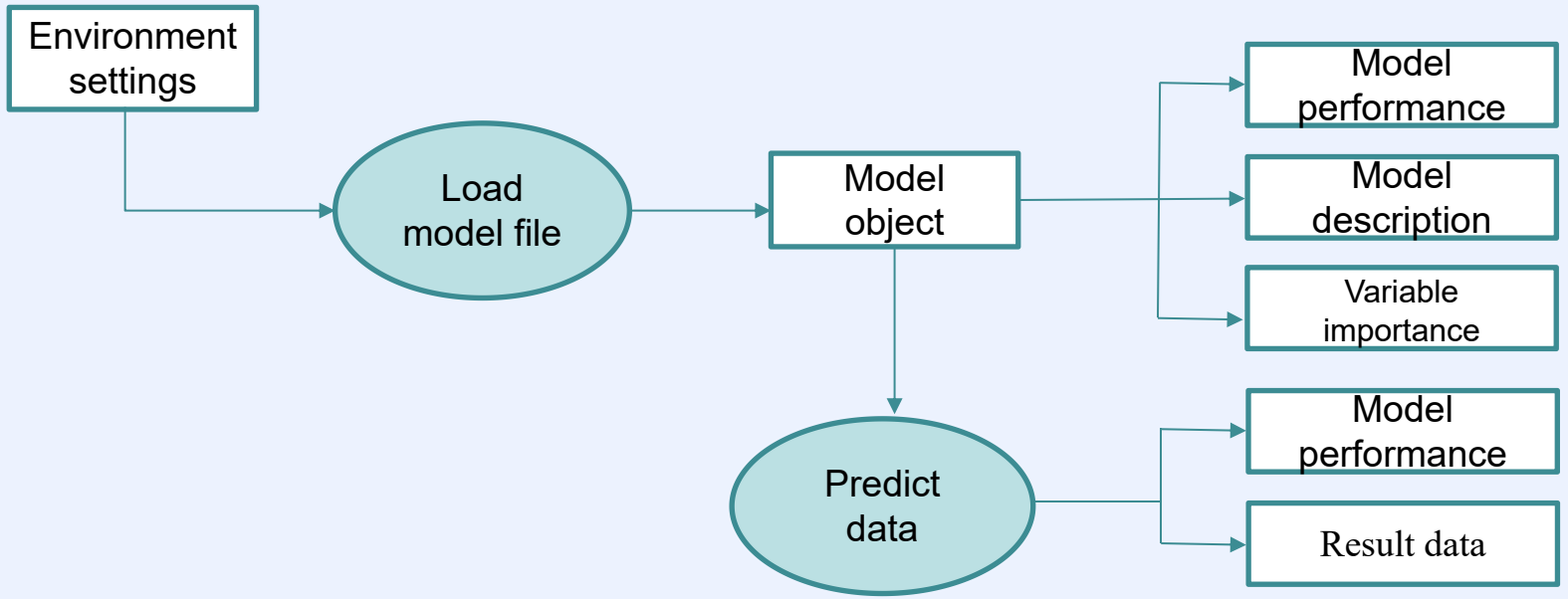
D. Modeling result

One or more model objects are generated after model building is finished and saved as pcf model file(s). The model object links model building and data prediction and provides necessary files for data prediction. The prediction process can start directly from loading a model file.



4 Data prediction

4.1 Prediction flowchart



Basic process: load model file -> predict data -> view result

4.2 SPL prediction example

4.2.1 SPL prediction example: Prediction that returns a table sequence

	A	Note
1	=ym_env()	Initialize environment
2	=ym_load_pcf("demo.pcf")	Load model object from the pcf model file
3	=file("D:/dev/test.csv").import@tqc()	Load the to-be-predicted data from a file and return a table sequence
4	=ym_predict(A2, A3(1))	Perform data prediction and return prediction result
5	>ym_close(A1)	Exit YModel

A4 returns prediction result:

Index	Embarked...	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
1	0.0063696...	591	0	3	Rintamaki, ...	male	35	0	0	STON/O 2. ...

A2 loads the pcf model file resulted from model building. A4 performs data prediction and returns prediction result.

4.2 SPL prediction example

4.2.2 SPL prediction example: Batch prediction that collects model performance at the same time

	A	Note
1	=ym_env()	Initialize environment
2	=ym_load_pcf("demo.pcf")	Load model object from the pcf model file
3	=file("D:/dev/test.csv").import@tqc()	Load to-be-predicted data from a file and return a table sequence
4	=ym_predict(A2, A3)	Perform data prediction and return prediction result
5	=ym_result(A4)	Return prediction result as a table sequence
6	=ym_json(A4)	When the to-be-predicted data is no less than 20 records for a batch prediction, the function will output JSON-format model performance information according to the data evaluation
7	>ym_close(A1)	Exit YModel

A2 loads the pcf model file resulted from model building. A4 performs data prediction and returns prediction result.

4.2 SPL prediction example

4.2.3 SPL prediction example: concurrency-based prediction that returns a table sequence. Use the number of parallel threads configured in "Environment variable" it is not specified.

	A	B	Note
1	=ym_env()	>b=0	Initialize environment
2	=ym_load_pcf("demo.pcf")		Load model object from the pcf model file
3	=file("D:/dev/test.csv").import@tqc()		Load to-be-predicted data from a file and return a table sequence
4	=3.(5.(b=b+1))		Generate a sequence 3 rows, each of which contains 5 numbers
5	=A4.(~.(A3.select(#==A4.~.~)(1)))		Get corresponding records from A3 according to A4's values
6	fork A5	=ym_predict@m(A2, A6)	Use @m to predict data with multithreaded processing and return prediction result as a table sequence
7	>ym_close(A1)		Exit YModel

A4

Index	Member
1	[1,2,3,...]
2	[6,7,8,...]
3	[11,12,13,...]

A5

Index	Member
1	[[1,0,3,...],[2,1,1,...],...]
2	[[6,0,3,...],[7,0,1,...],...]
3	[[11,0,3,...],[12,1,1,...],...]

Click to view details

PassengerId	Survived	Pclass	Name	...
1	0	3	Braund, Mr. ...	...
2	1	1	Cumings, Mrs. Jo...	...
3	1	3	Heikkinen, Miss. ...	...
4	1	1	Futrelle, Mrs...	...
5	0	5	Allen, Mr. Willi...	...

4.2 SPL prediction example

4.2.3 SPL prediction example: concurrency-based prediction that returns a table sequence.

Prediction result:

A6	
Index	Member
1	[[0.0123335141688585 28,1,0,...],...]
2	[[0.0109029887244105 34,6,0,...],...]
3	[[0.0158156286925077 44,11,1,...],...]

Click to view details

Embarked_predictvalue	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	...
0.0123335141688585 28	1	0	3	Braund, Mr. Owen...	male	22	1	0	...
0.7790963053703308	2	1	1	Cumings, Mrs. John...	female	38	1	0	...
0.0102028269320726 4	3	1	3	Heikkinen, Miss. Laina	female	26	0	0	...
0.0197689887136220 93	4	1	1	Futrelle, Mrs. Jacqu...	female	35	1	0	...
0.0029121323022991 42	5	0	3	Allen, Mr. William...	male	35	0	0	...

The regular prediction and concurrency-based prediction won't collect model performance information but directly return prediction result as a table sequence.

4.2 SPL prediction example

4.2.4 SPL prediction example: concurrency-based prediction for which waiting time should be configured and that returns a table sequence.

	A	B	Note
1	=ym_env()		Initialize environment
2	=ym_load_pcf("demo.pcf")		Load model object from the pcf model file
3	=file("D:/dev/test.csv").import@tqc()		Loadto-be-predicted data from a file and return a table sequence
4	=A3.to(100)		Get 100 records
5	fork A4	=ym_predict@m(A2,A5,100)	Use @m to predict data with multithreaded processing and then perform aggregation during the specified 100 milliseconds, and return prediction result as a table sequence
6	=A5.conj()		Concatenate result table sequences returned by threads
7	>ym_close(A1)		Exit YModel

A6	Index	PassengerId	Survived	Pclass	Name	...
	1	1	0	3	Braund, Mr. ...	...
	2	2	1	1	Cumings, Mrs. Jo...	...
	3	3	1	3	Heikkinen, Miss. ...	...
	4	4	1	1	Futrelle, Mrs...	...
	5	5	0	5	Allen, Mr. Willi...	...

4.3 Predictive model object

Load model file

The ym_load_pcf()interface generates a predictive model object according the loaded model file. Based on the same model file, a predictive model object and the model object generated by the model building process have same functionalities. Users can use the predictive model object to predict data, or get relevant model information, such as model description and model performance.

4.4 Prediction result

Predict valid data according to the predictive model and generate prediction result.
The data to be predicted can come from databases, table sequences, CSV files, etc.

A. Prediction result

Return prediction result:

Index	PassengerId	Survived	P...	Name	Sex	Age	...	...	Ticket	Fare	Ca...	Emb...	Survived_1_ratio
1	624	0	3	Hansen, M...	male	21	0	0	350029	7.8542	(null)	S	0.25707278881670026
2	625	0	3	"Bowen, Mr...	male	21	0	0	54636	16.1	(null)	S	0.5680230522833624
3	626	0	1	Sutton, Mr. ...	male	61	0	0	36963	32.3208	D50	S	0.23271421717823756
4	627	0	2	Kirkland, R...	male	57	0	0	219533	12.35	(null)	Q	0.5919246159939965
5	628	1	1	Longley, Mi...	female	21	0	0	13502	77.9583	D9	S	0.4284559885569835
6	629	0	3	Bostandyef...	male	26	0	0	349224	7.8958	(null)	S	0.298675652025568
7	630	0	3	O'Connell, ...	male	(null)	0	0	334912	7.7333	(null)	Q	0.06259809523413039
8	631	1	1	Barkworth, ...	male	80	0	0	27042	30.0	A23	S	0.20693993819873205
9	632	0	3	Lundahl, M...	male	51	0	0	347743	7.0542	(null)	S	0.04720448510306918
10	633	1	1	Stahelin-M...	male	32	0	0	13214	30.5	B50	C	0.5263867128075925
11	634	0	1	Parr, Mr. Wi...	male	(null)	0	0	112052	0.0	(null)	S	0.04965320978124586
12	635	0	3	Skoog, Mis...	female	9	3	2	347088	27.9	(null)	S	0.5855400248475947
13	636	1	2	Davis, Mis...	female	28	0	0	237668	13.0	(null)	S	0.38185936013549643
14	637	0	3	Leinonen, ...	male	32	0	0	STON/O 2. ...	7.925	(null)	S	0.1385752420653835
15	638	0	2	Collyer, Mr....	male	31	1	1	C.A. 31921	26.25	(null)	S	0.7216118101299388
16	639	0	3	Panula, Mr...	female	41	0	5	3101295	39.6875	(null)	S	0.338041641614491
17	640	0	3	Thorneycro...	male	(null)	1	0	376564	16.1	(null)	S	0.4347392114485213
18	641	0	3	Jensen, Mr...	male	20	0	0	350050	7.8542	(null)	S	0.25707278881670026

4.4 Prediction result

B. Model performance

When the to-be-predicted data contains target variables, users can view model performance according to the prediction result. This functionality gets model performance from the prediction result. We can evaluate the model quality, which is described by Gini, AUC, KS or others,by comparing the model performance here with that in the model file. The model performance can be exported to a JSON-style file.

Index	Name	Value
1	AUC	0.830062984496124
2	GINI	0.6601259689922481
3	KS	0.5666182170542636
4	AccuTable	[[0.05000000074505806,0.5074626865671642,0.41818181...
5	RocTable	[[0.0,0.020833333333333332],[0.0,0.041666666666666664]...
6	LiftAndRecallTable	[[1,2.7916666666666667,0.13541666666666666, ...],[2,2.362...

5 Summary

SPL Auto-Modeling Process combines the user's statistical knowledge and algorithm techniques with the business requirements through simple and convenient operation.
Below is the modeling process flowchart: