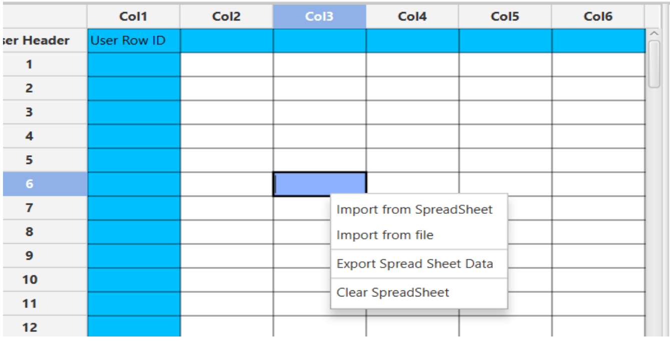
Credit Card (Binary Classification)

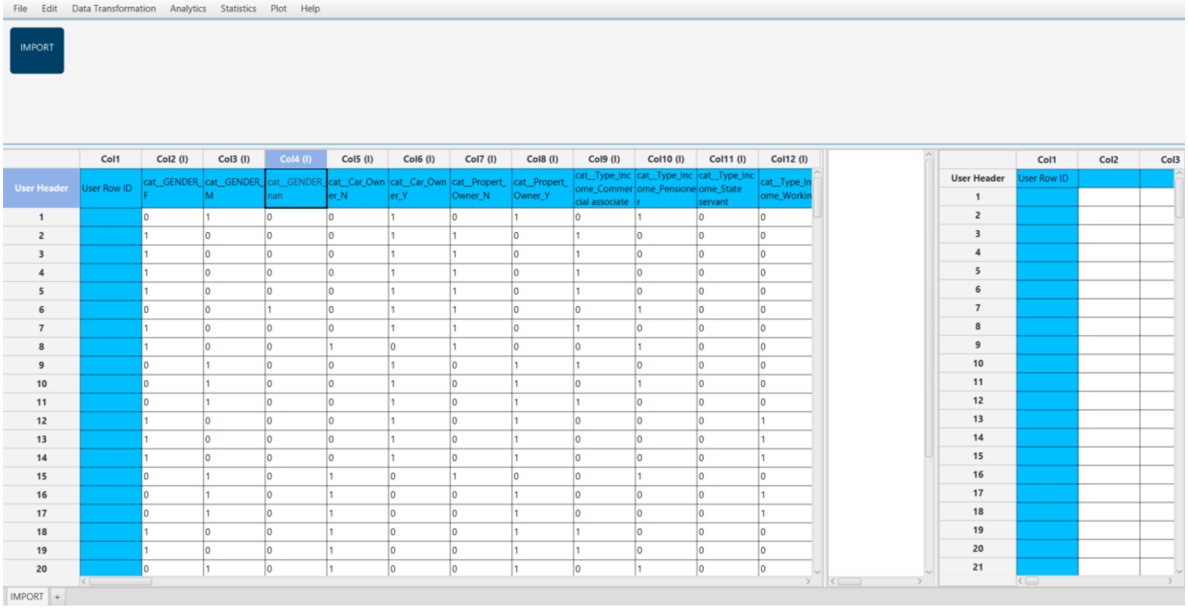
The goal of this study is to train a model in order to predict whether the application is Approved (0) or Rejected (1). The dataset used in this case study is found in https://www.kaggle.com/datasets/rohitudageri/credit-card-details?select=Credit_card_label.csv and has 20 features and 1458 labelled samples.

Step 1: Import Data from the file

Right click on the input spreadsheet and choose the option "Import from file". Then navigate through your files to find the one with the credit card data.



	Col1	Col2	Col3	Col4	Col5	Col6
ser Header	User Row ID					
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						



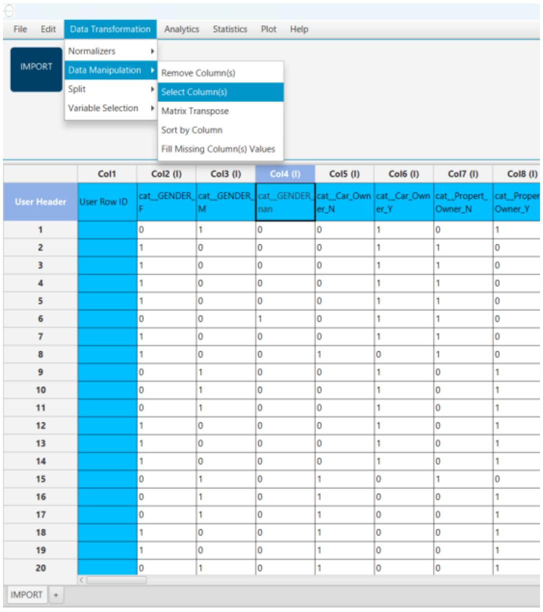
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	Col7 (I)	Col8 (I)	Col9 (I)	Col10 (I)	Col11 (I)	Col12 (I)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat__Type_Income_Commercial associate	cat__Type_Income_Pensioner	cat__Type_Income_State servant	cat__Type_Income_Workin
1		0	1	0	0	1	0	1	0	1	0	0
2		1	0	0	0	1	1	0	1	0	0	0
3		1	0	0	0	1	1	0	1	0	0	0
4		1	0	0	0	1	1	0	1	0	0	0
5		1	0	0	0	1	1	0	1	0	0	0
6		0	0	1	0	1	1	0	0	1	0	0
7		1	0	0	0	1	1	0	1	0	0	0
8		1	0	0	1	0	1	0	0	1	0	0
9		0	1	0	0	1	0	1	1	0	0	0
10		0	1	0	0	1	0	1	0	1	0	0
11		0	1	0	0	1	0	1	1	0	0	0
12		1	0	0	0	1	0	1	0	0	0	1
13		1	0	0	0	1	0	1	0	0	0	1
14		1	0	0	0	1	0	1	0	0	0	1
15		0	1	0	1	0	1	0	0	1	0	0
16		0	1	0	1	0	0	1	0	0	0	1
17		0	1	0	1	0	0	1	0	0	0	1
18		1	0	0	1	0	0	1	1	0	0	0
19		1	0	0	1	0	0	1	1	0	0	0
20		0	1	0	1	0	0	1	0	1	0	0

	Col1	Col2	Col3
User Header	User Row ID		
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			

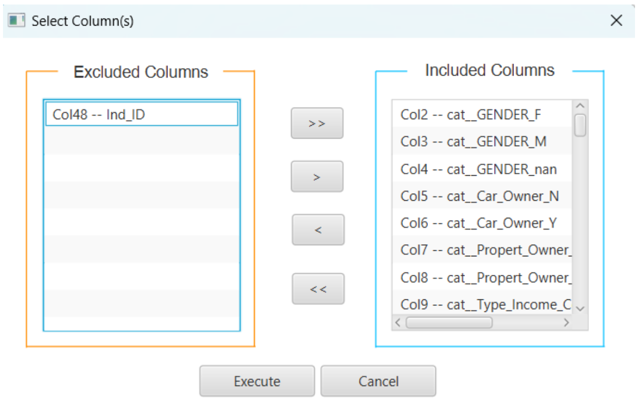
Step 2: Manipulate Data

In order to use the data for training we have to exclude any columns that do not represent factor, like Ind_ID. We follow these steps to execute this:

- Browse: "Data Transformation" → "Data Manipulation" → "Select Column(s)".
- Select all columns except the one that corresponds to the Ind_ID.



	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	Col7 (I)	Col8 (I)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Proper Owner_Y
1		0	1	0	0	1	0	1
2		1	0	0	0	1	1	0
3		1	0	0	0	1	1	0
4		1	0	0	0	1	1	0
5		1	0	0	0	1	1	0
6		0	0	1	0	1	1	0
7		1	0	0	0	1	1	0
8		1	0	0	1	0	1	0
9		0	1	0	0	1	0	1
10		0	1	0	0	1	0	1
11		0	1	0	0	1	0	1
12		1	0	0	0	1	0	1
13		1	0	0	0	1	0	1
14		1	0	0	0	1	0	1
15		0	1	0	1	0	1	0
16		0	1	0	1	0	0	1
17		0	1	0	1	0	0	1
18		1	0	0	1	0	0	1
19		1	0	0	1	0	0	1
20		0	1	0	1	0	0	1



Excluded Columns

Col48 -- Ind_ID

Included Columns

Col2 -- cat__GENDER_F
Col3 -- cat__GENDER_M
Col4 -- cat__GENDER_nan
Col5 -- cat__Car_Owner_N
Col6 -- cat__Car_Owner_Y
Col7 -- cat__Propert_Owner_
Col8 -- cat__Propert_Owner_
Col9 -- cat__Type_Income_C

The data without the Ind_ID column will appear in the output spreadsheet.

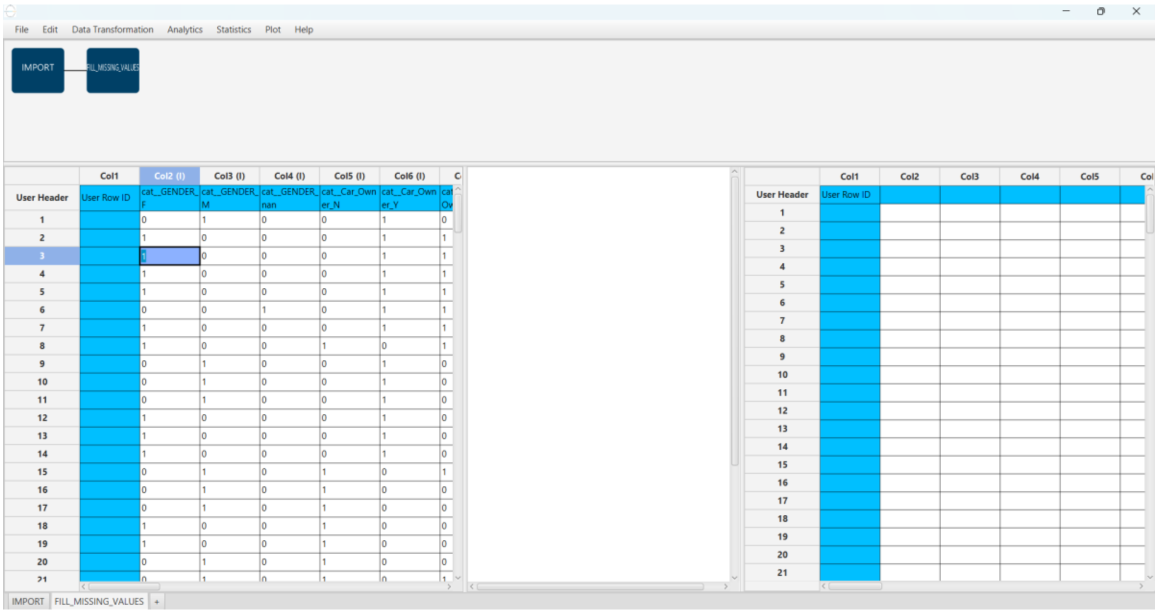
Step 3: Fill missing values

There are empty values in the Dataset. Specifically, we show below how many missing values there are for each feature:

```
Empty data:
Ind_ID              0
GENDER              7
Car_Owner           0
Propert_Owner       0
CHILDREN            0
Annual_income      23
Type_Income         0
EDUCATION           0
Marital_status      0
Housing_type        0
Birthday_count     22
Employed_days       0
Mobile_phone        0
Work_Phone          0
Phone               0
EMAIL_ID            0
Type_Occupation   488
Family_Members      0
dtype: int64
```

Create a new tab by pressing the "+" button on the bottom of the page with the name FILL_MISSING VALUES which will be used to fill the missing values.

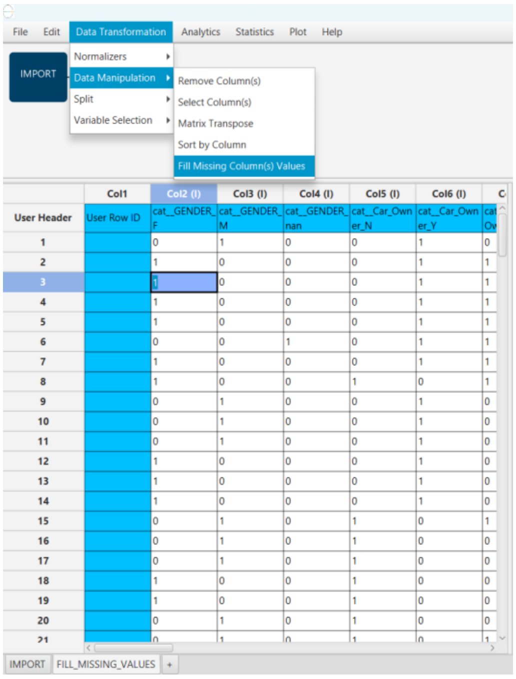
Import Data into the input spreadsheet of the FILL_MISSING_VALUES tab from the output of the IMPORT tab by right-clicking on the input spreadsheet and then choosing Import from SpreadSheet.



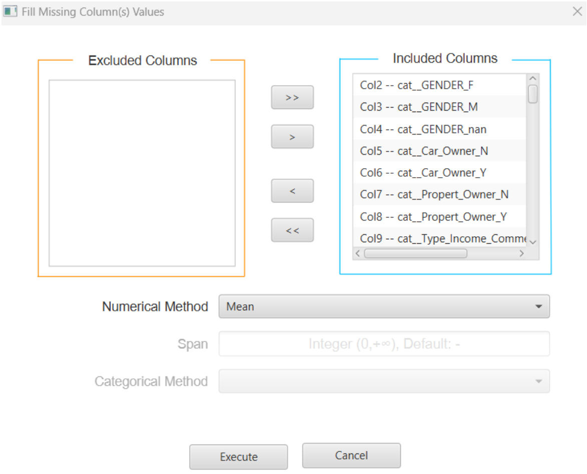
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	0	1	1
3		1	0	0	0	1	1
4		1	0	0	0	1	1
5		1	0	0	0	1	1
6		0	0	1	0	1	1
7		1	0	0	0	1	1
8		1	0	0	1	0	1
9		0	1	0	0	1	0
10		0	1	0	0	1	0
11		0	1	0	0	1	0
12		1	0	0	0	1	0
13		1	0	0	0	1	0
14		1	0	0	0	1	0
15		0	1	0	1	0	1
16		0	1	0	1	0	0
17		0	1	0	1	0	0
18		1	0	0	1	0	0
19		1	0	0	1	0	0
20		0	1	0	1	0	0
21		0	1	0	1	0	1

	Col1	Col2	Col3	Col4	Col5	Col
User Header	User Row ID					
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

Handle missing columns values by browsing: "Data Transformation" → "Data Manipulation" → "Fill missing column(s) Values". Then choose the Mean as the Numerical Method.



	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	0	1	1
3		1	0	0	0	1	1
4		1	0	0	0	1	1
5		1	0	0	0	1	1
6		0	0	1	0	1	1
7		1	0	0	0	1	1
8		1	0	0	1	0	1
9		0	1	0	0	1	0
10		0	1	0	0	1	0
11		0	1	0	0	1	0
12		1	0	0	0	1	0
13		1	0	0	0	1	0
14		1	0	0	0	1	0
15		0	1	0	1	0	1
16		0	1	0	1	0	0
17		0	1	0	1	0	0
18		1	0	0	1	0	0
19		1	0	0	1	0	0
20		0	1	0	1	0	0
21		0	1	0	1	0	1



Excluded Columns

Included Columns

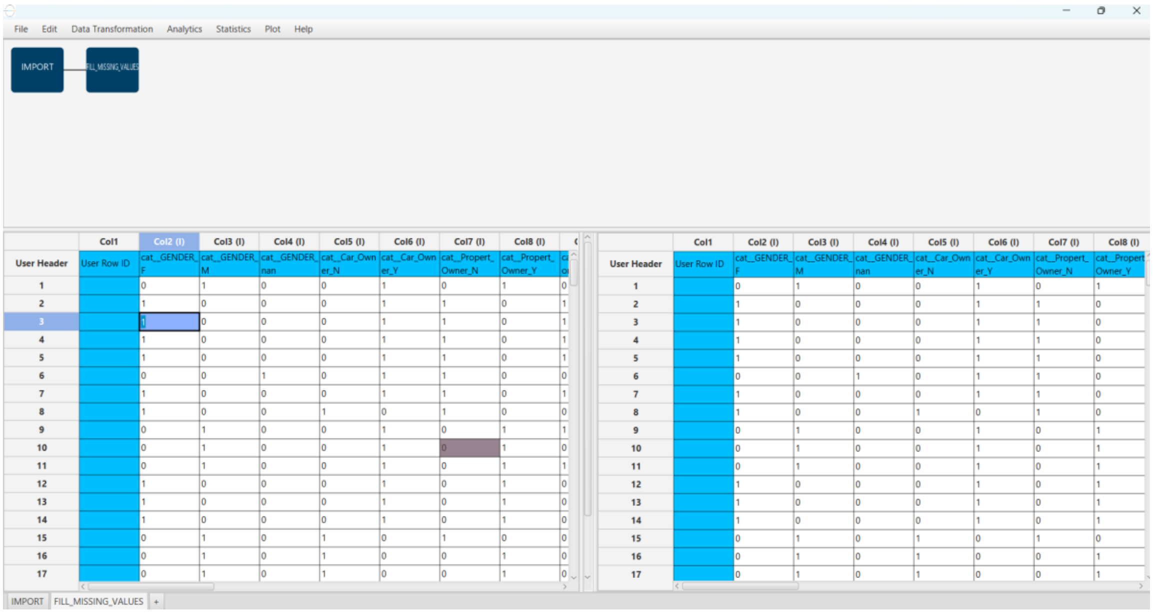
Col2 -- cat__GENDER_F
Col3 -- cat__GENDER_M
Col4 -- cat__GENDER_nan
Col5 -- cat__Car_Owner_N
Col6 -- cat__Car_Owner_Y
Col7 -- cat__Propert_Owner_N
Col8 -- cat__Propert_Owner_Y
Col9 -- cat__Type_Income_Comme

Numerical Method: Mean

Span: Integer (0,+∞), Default: -

Categorical Method:

The results will appear on the output spreadsheet.



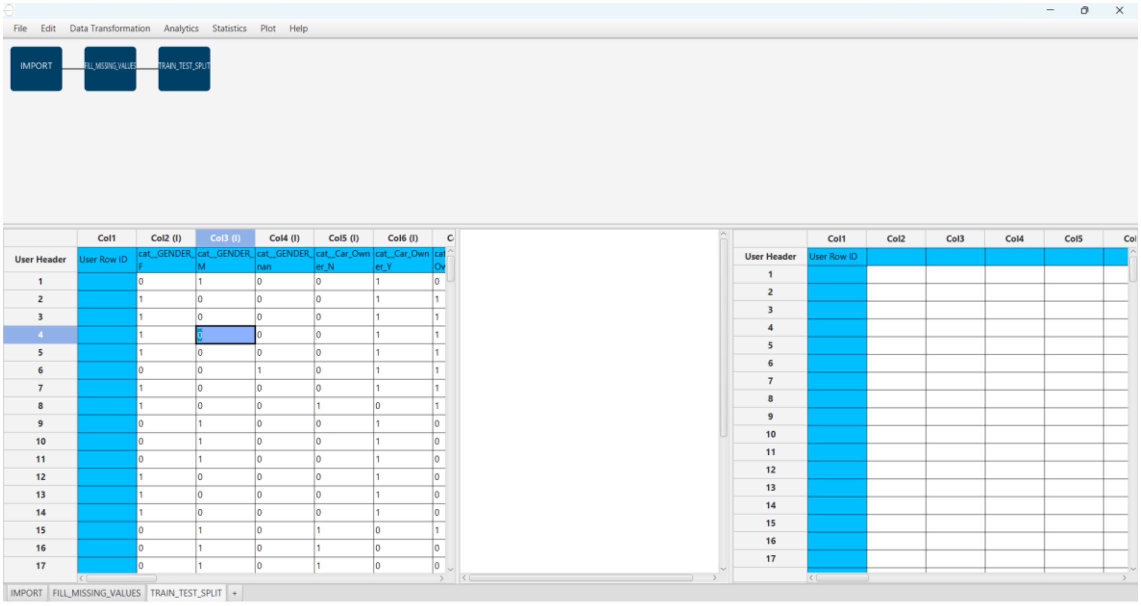
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	Col7 (I)	Col8 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	ca o
1		0	1	0	0	1	0	1	0
2		1	0	0	0	1	1	0	1
3		1	0	0	0	1	1	0	1
4		1	0	0	0	1	1	0	1
5		1	0	0	0	1	1	0	1
6		0	0	1	0	1	1	0	0
7		1	0	0	0	1	1	0	1
8		1	0	0	1	0	1	0	0
9		0	1	0	0	1	0	1	1
10		0	1	0	0	1	0	1	0
11		0	1	0	0	1	0	1	1
12		1	0	0	0	1	0	1	0
13		1	0	0	0	1	0	1	0
14		1	0	0	0	1	0	1	0
15		0	1	0	1	0	1	0	0
16		0	1	0	1	0	0	1	0
17		0	1	0	1	0	0	1	0

	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	Col7 (I)	Col8 (I)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y
1		0	1	0	0	1	0	1
2		1	0	0	0	1	1	0
3		1	0	0	0	1	1	0
4		1	0	0	0	1	1	0
5		1	0	0	0	1	1	0
6		0	0	1	0	1	1	0
7		1	0	0	0	1	1	0
8		1	0	0	1	0	1	0
9		0	1	0	0	1	0	1
10		0	1	0	0	1	0	1
11		0	1	0	0	1	0	1
12		1	0	0	0	1	0	1
13		1	0	0	0	1	0	1
14		1	0	0	0	1	0	1
15		0	1	0	1	0	1	0
16		0	1	0	1	0	0	1
17		0	1	0	1	0	0	1

Step 4: Split Data

Create a new tab by pressing the + button on the bottom of the page with the name TRAIN_TEST_SPLIT which we will use for splitting to create the train and test set.

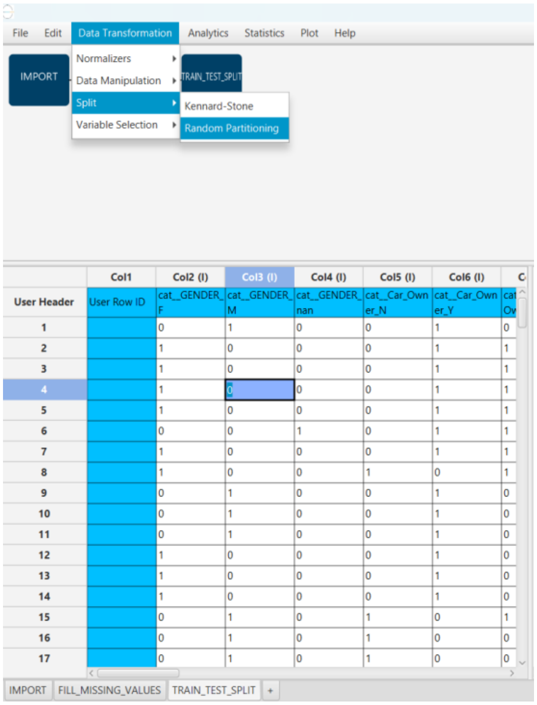
Import Data into the input spreadsheet of the TRAIN_TEST_SPLIT tab from the output of the FILL_MISSING_VALUES tab by right-clicking on the input spreadsheet and then choosing Import from SpreadSheet.



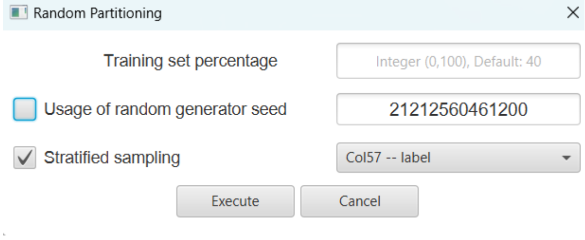
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	0	1	1
3		1	0	0	0	1	1
4		1	0	0	0	1	1
5		1	0	0	0	1	1
6		0	0	1	0	1	1
7		1	0	0	0	1	1
8		1	0	0	1	0	1
9		0	1	0	0	1	0
10		0	1	0	0	1	0
11		0	1	0	0	1	0
12		1	0	0	0	1	0
13		1	0	0	0	1	0
14		1	0	0	0	1	0
15		0	1	0	1	0	1
16		0	1	0	1	0	0
17		0	1	0	1	0	0

	Col1	Col2	Col3	Col4	Col5	Col
User Header	User Row ID					
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

Split the dataset by browsing "Data Transformation" → "Split" → "Random Partitioning". Then choose the training set percentage and the column for the sampling as shown below.



	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	0	1	1
3		1	0	0	0	1	1
4		1	0	0	0	1	1
5		1	0	0	0	1	1
6		0	0	1	0	1	1
7		1	0	0	0	1	1
8		1	0	0	1	0	1
9		0	1	0	0	1	0
10		0	1	0	0	1	0
11		0	1	0	0	1	0
12		1	0	0	0	1	0
13		1	0	0	0	1	0
14		1	0	0	0	1	0
15		0	1	0	1	0	1
16		0	1	0	1	0	0
17		0	1	0	1	0	0



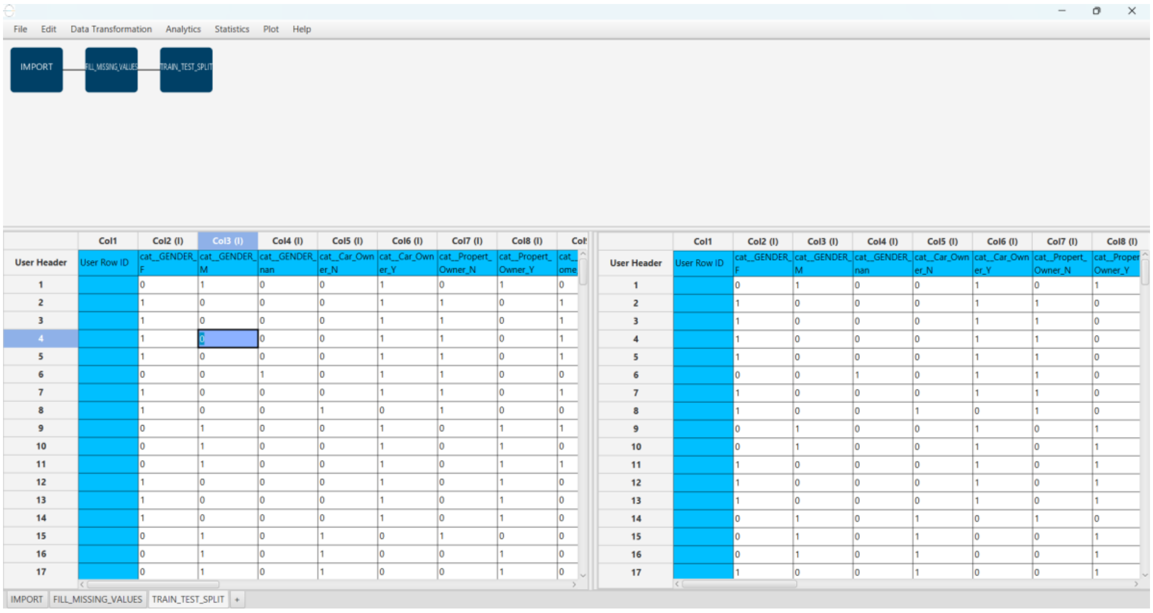
Random Partitioning

Training set percentage: Integer (0,100), Default: 40

☐ Usage of random generator seed: 21212560461200

☑ Stratified sampling: Col57 -- label

The results will appear on the output spreadsheet.



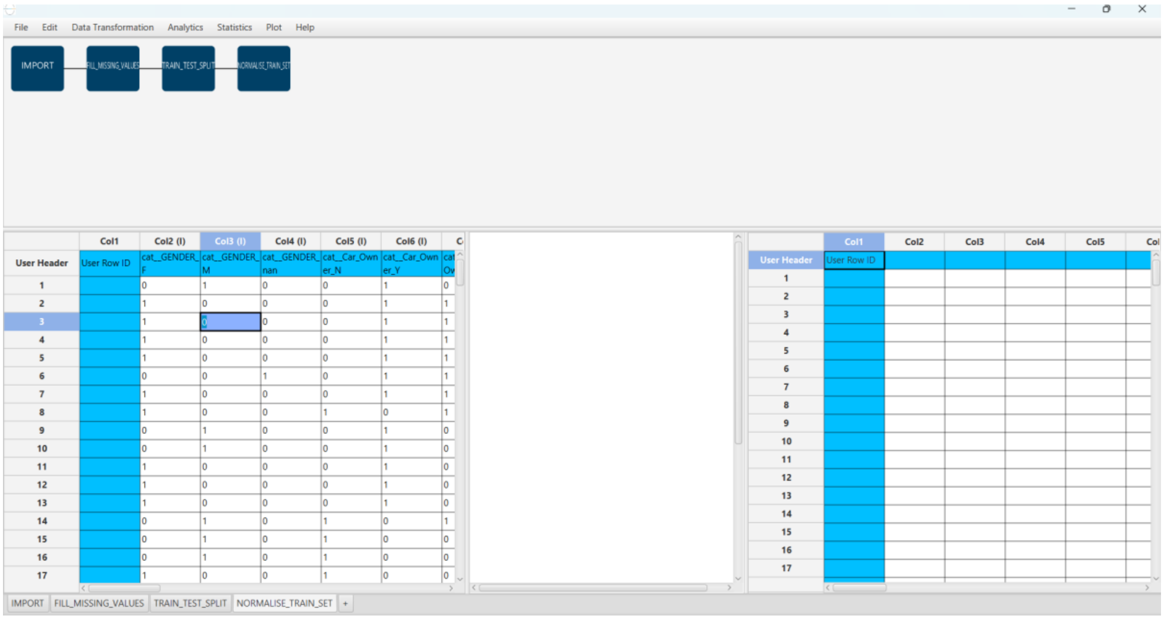
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	Col7 (I)	Col8 (I)	Col
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat__ome
1		0	1	0	0	1	0	1	0
2		1	0	0	0	1	1	0	1
3		1	0	0	0	1	1	0	1
4		1	0	0	0	1	1	0	1
5		1	0	0	0	1	1	0	1
6		0	0	1	0	1	1	0	0
7		1	0	0	0	1	1	0	1
8		1	0	0	1	0	1	0	0
9		0	1	0	0	1	0	1	1
10		0	1	0	0	1	0	1	0
11		0	1	0	0	1	0	1	1
12		1	0	0	0	1	0	1	0
13		1	0	0	0	1	0	1	0
14		1	0	0	0	1	0	1	0
15		0	1	0	1	0	1	0	0
16		0	1	0	1	0	0	1	0
17		0	1	0	1	0	0	1	0

	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	Col7 (I)	Col8 (I)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Proper Owner_Y
1		0	1	0	0	1	0	1
2		1	0	0	0	1	1	0
3		1	0	0	0	1	1	0
4		1	0	0	0	1	1	0
5		1	0	0	0	1	1	0
6		0	0	1	0	1	1	0
7		1	0	0	0	1	1	0
8		1	0	0	1	0	1	0
9		0	1	0	0	1	0	1
10		0	1	0	0	1	0	1
11		1	0	0	0	1	0	1
12		1	0	0	0	1	0	1
13		1	0	0	0	1	0	1
14		0	1	0	1	0	1	0
15		0	1	0	1	0	0	1
16		0	1	0	1	0	0	1
17		1	0	0	1	0	0	1

Step 5: Normalize the Training Set

Create a new tab by pressing the + button on the bottom of the page with the name NORMALISE_TRAIN_SET.

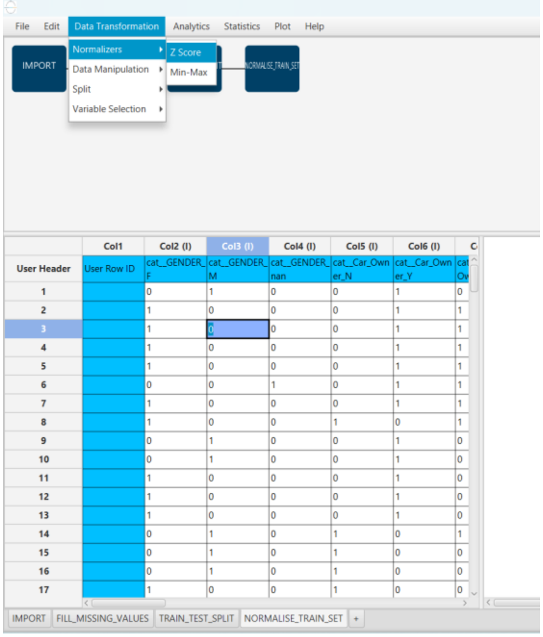
Import Data into the input spreadsheet of the NORMALISE_TRAIN_SET tab the train set from the output of the TRAIN_TEST_SPLIT tab by right-clicking on the input spreadsheet and then choosing Import from SpreadSheet. From the available Select input tab options choose TRAIN_TEST_SPLIT: Training Set



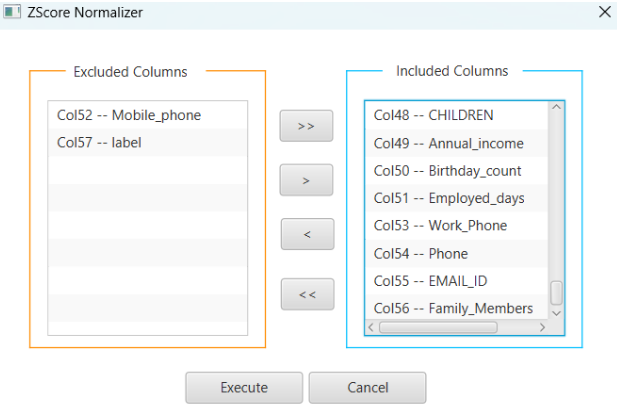
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	0	1	1
3		1	0	0	0	1	1
4		1	0	0	0	1	1
5		1	0	0	0	1	1
6		0	0	1	0	1	1
7		1	0	0	0	1	1
8		1	0	0	1	0	1
9		0	1	0	0	1	0
10		0	1	0	0	1	0
11		1	0	0	0	1	0
12		1	0	0	0	1	0
13		1	0	0	0	1	0
14		0	1	0	1	0	1
15		0	1	0	1	0	0
16		0	1	0	1	0	0
17		1	0	0	1	0	0

	Col1	Col2	Col3	Col4	Col5	Col
User Header	User Row ID					
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

Normalize the Data using Z-score by browsing: "Data Transformation" → "Normalize" → "Z-Score". Then select all columns excluding Mobile_phone and Label and click Execute.



	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	0	1	1
3		1	0	0	0	1	1
4		1	0	0	0	1	1
5		1	0	0	0	1	1
6		0	0	1	0	1	1
7		1	0	0	0	1	1
8		1	0	0	1	0	1
9		0	1	0	0	1	0
10		0	1	0	0	1	0
11		1	0	0	0	1	0
12		1	0	0	0	1	0
13		1	0	0	0	1	0
14		0	1	0	1	0	1
15		0	1	0	1	0	0
16		0	1	0	1	0	0
17		1	0	0	1	0	0



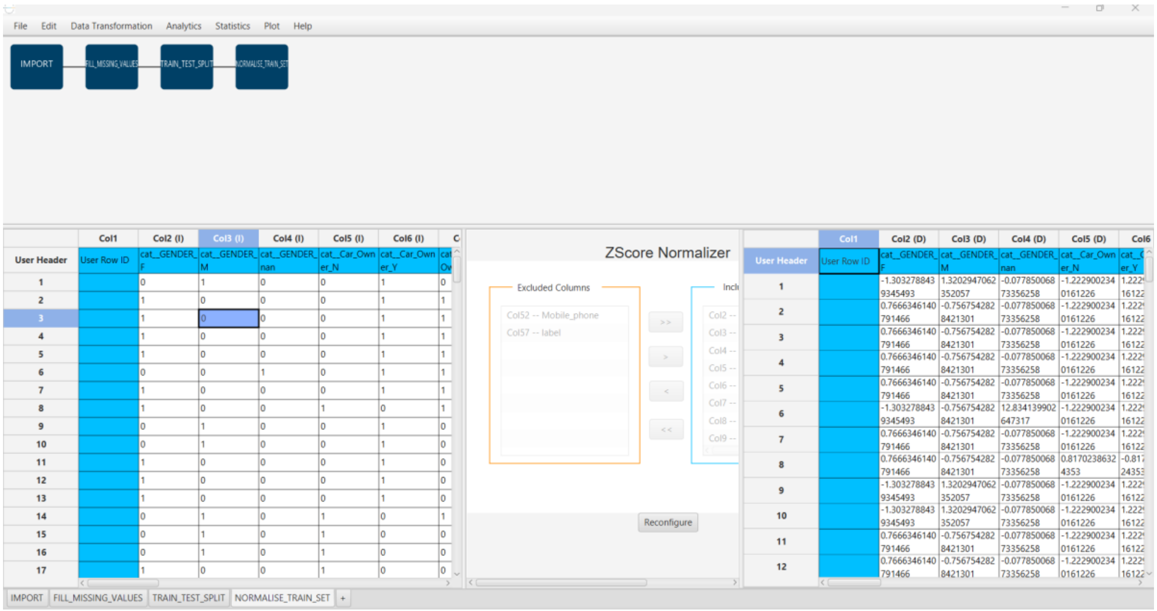
Excluded Columns

- Col52 -- Mobile_phone
- Col57 -- label

Included Columns

- Col48 -- CHILDREN
- Col49 -- Annual_income
- Col50 -- Birthday_count
- Col51 -- Employed_days
- Col53 -- Work_Phone
- Col54 -- Phone
- Col55 -- EMAIL_ID
- Col56 -- Family_Members

The results will appear on the output spreadsheet.



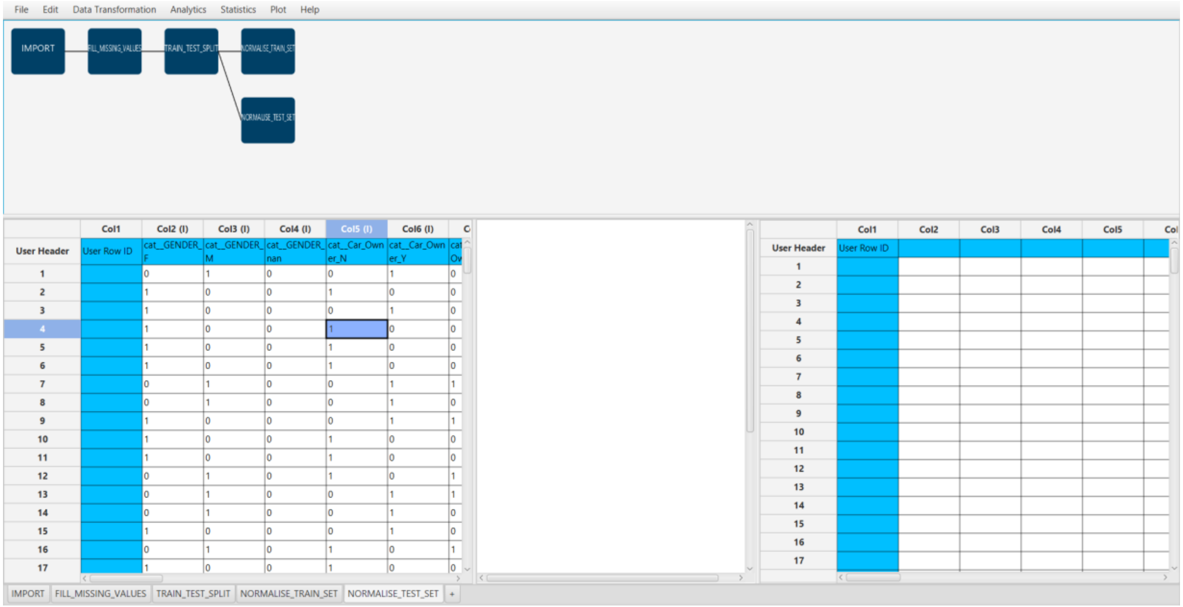
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	0	1	1
3		1	0	0	0	1	1
4		1	0	0	0	1	1
5		1	0	0	0	1	1
6		0	0	1	0	1	1
7		1	0	0	0	1	1
8		1	0	0	1	0	1
9		0	1	0	0	1	0
10		0	1	0	0	1	0
11		1	0	0	0	1	0
12		1	0	0	0	1	0
13		1	0	0	0	1	0
14		0	1	0	1	0	1
15		0	1	0	1	0	0
16		0	1	0	1	0	0
17		1	0	0	1	0	0

	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__C er_Y
1		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.222 16122
2		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.222 16122
3		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.222 16122
4		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.222 16122
5		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.222 16122
6		-1.3032788439345493	-0.7567542828421301	12.834139902647317	-1.2229002340161226	1.222 16122
7		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.222 16122
8		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.817 24353
9		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.222 16122
10		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.222 16122
11		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.222 16122
12		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.222 16122

Step 6: Normalize the Test Set

Create a new tab by pressing the + button on the bottom of the page with the name NORMALISE_TEST_SET.

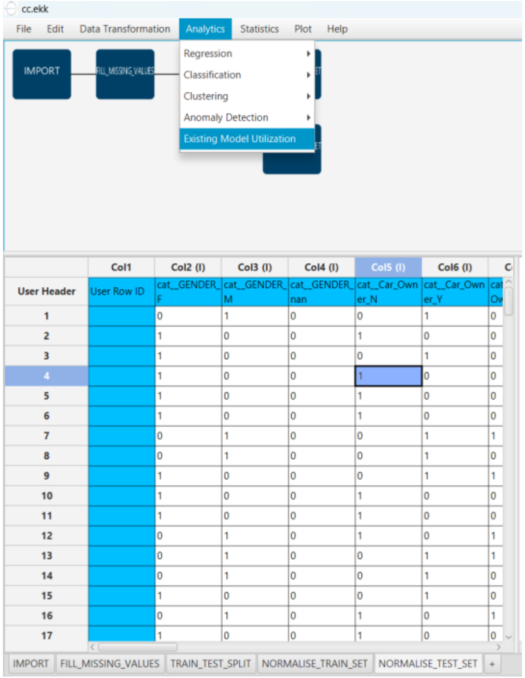
Import Data into the input spreadsheet of the NORMALISE_TEST_SET tab the test set from the output of the TRAIN_TEST_SPLIT tab by right-clicking on the input spreadsheet and then choosing Import from SpreadSheet. From the available Select input tab options choose TRAIN_TEST_SPLIT: Test Set.



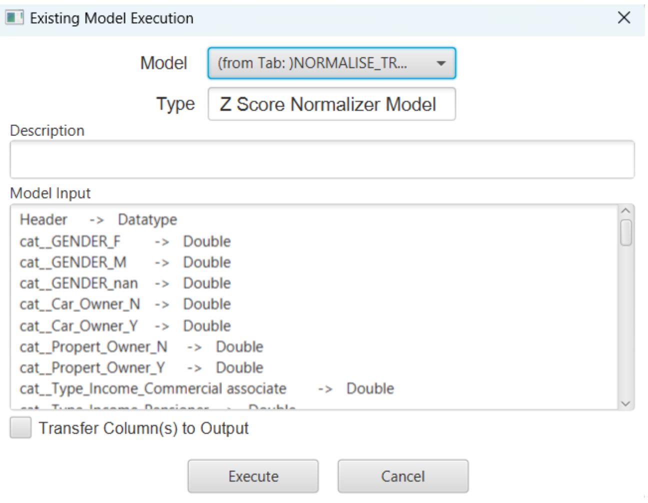
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	1	0	0
3		1	0	0	0	1	0
4		1	0	0	1	0	0
5		1	0	0	1	0	0
6		1	0	0	1	0	0
7		0	1	0	0	1	1
8		0	1	0	0	1	0
9		1	0	0	0	1	1
10		1	0	0	1	0	0
11		1	0	0	1	0	0
12		0	1	0	1	0	1
13		0	1	0	0	1	1
14		0	1	0	0	1	0
15		1	0	0	0	1	0
16		0	1	0	1	0	1
17		1	0	0	1	0	0

	Col1	Col2	Col3	Col4	Col5	Col
User Header	User Row ID					
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						

Normalize the test set using the existing normalizer of the training set by browsing: "Analytics" → "Existing Model Utilization" → "Model: NORMALIZE_TRAIN_SET".



	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	C
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat Ov
1		0	1	0	0	1	0
2		1	0	0	1	0	0
3		1	0	0	0	1	0
4		1	0	0	1	0	0
5		1	0	0	1	0	0
6		1	0	0	1	0	0
7		0	1	0	0	1	1
8		0	1	0	0	1	0
9		1	0	0	0	1	1
10		1	0	0	1	0	0
11		1	0	0	1	0	0
12		0	1	0	1	0	1
13		0	1	0	0	1	1
14		0	1	0	0	1	0
15		1	0	0	0	1	0
16		0	1	0	1	0	1
17		1	0	0	1	0	0


Existing Model Execution

Model (from Tab:)NORMALISE_TR...

Type Z Score Normalizer Model

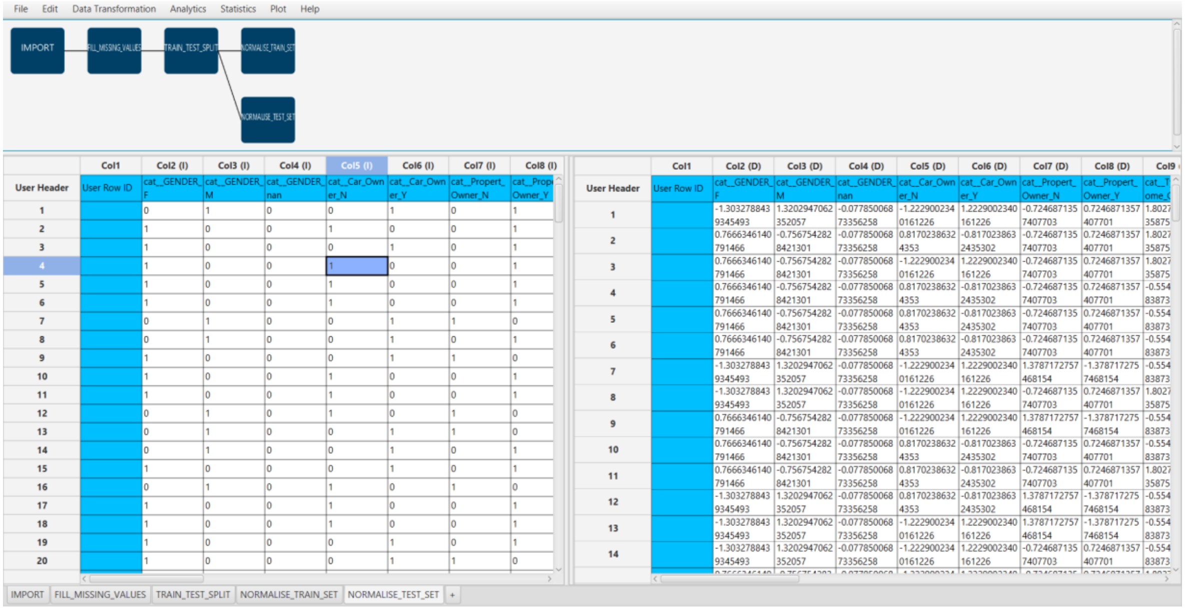
Description

Model Input

Header -> Datatype
cat__GENDER_F -> Double
cat__GENDER_M -> Double
cat__GENDER_nan -> Double
cat__Car_Owner_N -> Double
cat__Car_Owner_Y -> Double
cat__Propert_Owner_N -> Double
cat__Propert_Owner_Y -> Double
cat__Type_Income_Commercial associate -> Double

□ Transfer Column(s) to Output

The results will appear on the output spreadsheet.



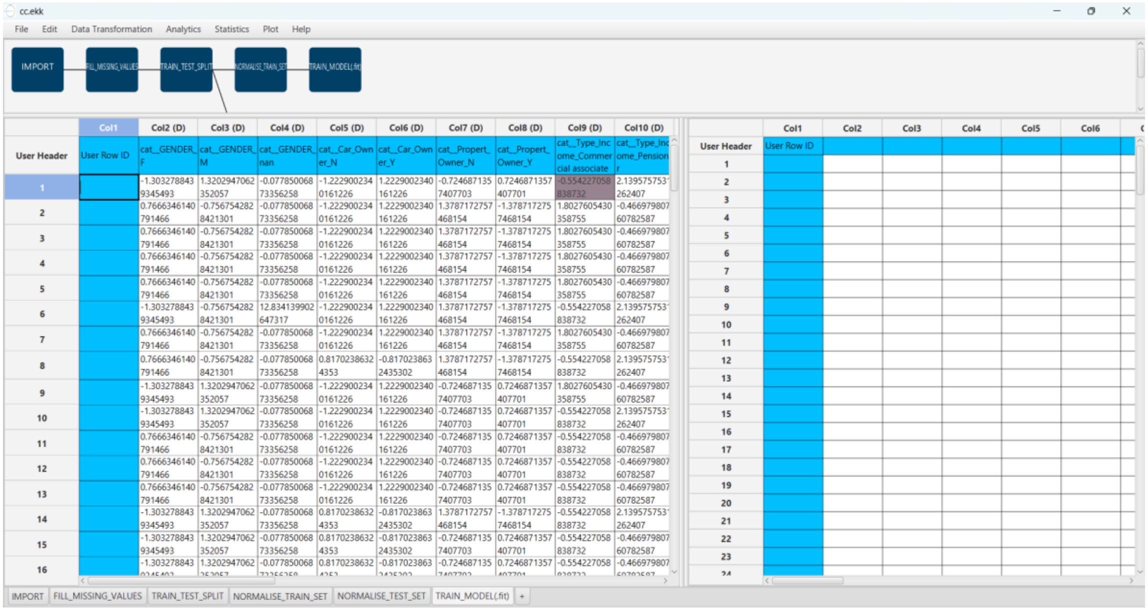
	Col1	Col2 (I)	Col3 (I)	Col4 (I)	Col5 (I)	Col6 (I)	Col7 (I)	Col8 (I)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propei Owner_Y
1		0	1	0	0	1	0	1
2		1	0	0	1	0	0	1
3		1	0	0	0	1	0	1
4		1	0	0	1	0	0	1
5		1	0	0	1	0	0	1
6		1	0	0	1	0	0	1
7		0	1	0	0	1	1	0
8		0	1	0	0	1	0	1
9		1	0	0	0	1	1	0
10		1	0	0	1	0	0	1
11		1	0	0	1	0	0	1
12		0	1	0	1	0	1	0
13		0	1	0	0	1	1	0
14		0	1	0	0	1	0	1
15		1	0	0	0	1	0	1
16		0	1	0	1	0	1	0
17		1	0	0	1	0	0	1
18		1	0	0	1	0	0	1
19		1	0	0	0	1	0	1
20		1	0	0	0	1	1	0

	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (D)	Col9
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat__T ome_(
1		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	1.802735875
2		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	1.802735875
3		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	1.802735875
4		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
5		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
6		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
7		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	-0.55483873
8		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	1.802735875
9		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	-0.55483873
10		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
11		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	1.802735875
12		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	0.81702386324353	-0.8170238632435302	1.3787172757468154	-1.3787172757468154	-0.55483873
13		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	-0.55483873
14		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	-0.55483873

Step 7: Train the model

Create a new tab by pressing the "+" button on the bottom of the page with the name "TRAIN_MODEL(.fit)".

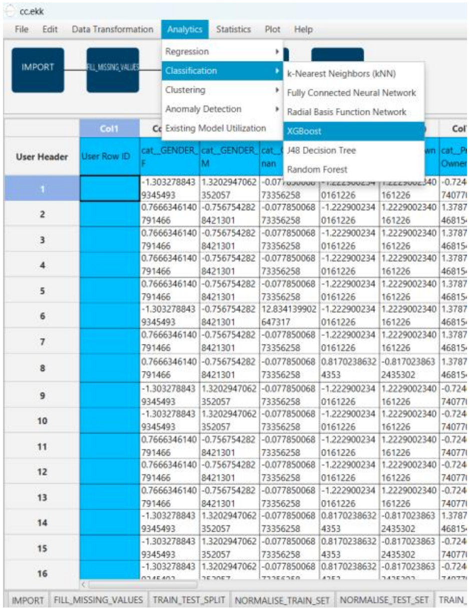
Import data into the input spreadsheet of the "TRAIN_MODEL(.fit)" tab from the output of the "NORMALISE_TRAIN_SET" tab by right-clicking on the input spreadsheet and then choosing "Import from SpreadSheet".

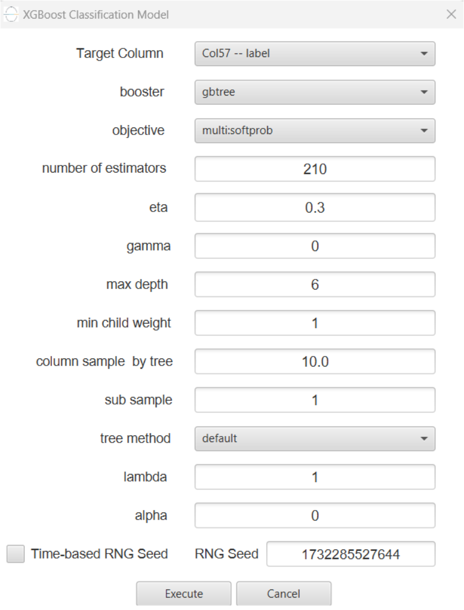


	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (D)	Col9 (D)	Col10 (D)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat__Type_Income_Commercial associate	cat__Type_Income_Pensioner
1		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	-0.554227058838732	2.139575753262407
2		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	1.8027605430358755	-0.46697980760782587
3		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	1.8027605430358755	-0.46697980760782587
4		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	1.8027605430358755	-0.46697980760782587
5		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	1.8027605430358755	-0.46697980760782587
6		-1.3032788439345493	-0.7567542828421301	12.834139902647317	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	-0.554227058838732	2.139575753262407
7		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	1.8027605430358755	-0.46697980760782587
8		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	1.3787172757468154	-1.3787172757468154	-0.554227058838732	2.139575753262407
9		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	1.8027605430358755	-0.46697980760782587
10		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	-0.554227058838732	2.139575753262407
11		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	-0.554227058838732	-0.46697980760782587
12		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	-0.554227058838732	-0.46697980760782587
13		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	-0.554227058838732	-0.46697980760782587
14		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	0.81702386324353	-0.8170238632435302	1.3787172757468154	-1.3787172757468154	-0.554227058838732	2.139575753262407
15		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.554227058838732	-0.46697980760782587
16		-1.303278843	1.3202947062	-0.077850068	0.8170238632	-0.817023863	-0.724687135	0.7246871357	-0.554227058	-0.466979807

	Col1	Col2	Col3	Col4	Col5	Col6
User Header	User Row ID					

Use the XGBoost Method to train and fit the model by browsing: "Analytics" → "Classification" → "XGBoost" and set the "number of estimators" as 210, the "column sample by tree" as 10, the "Target Column" as the column corresponding to "Label" and use the following "RNG Seed": 1732285527644.




XGBoost Classification Model

Target Column: Col57 -- label

booster: gbtree

objective: multi:softprob

number of estimators: 210

eta: 0.3

gamma: 0

max depth: 6

min child weight: 1

column sample by tree: 10.0

sub sample: 1

tree method: default

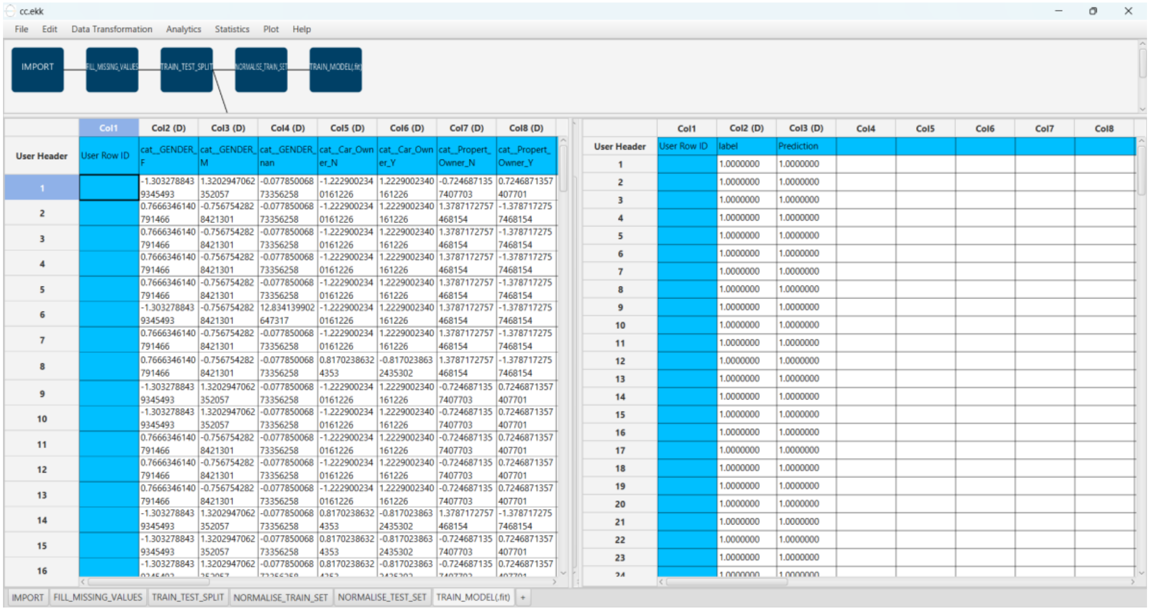
lambda: 1

alpha: 0

☐ Time-based RNG Seed

RNG Seed: 1732285527644

The predictions will appear on the output spreadsheet.

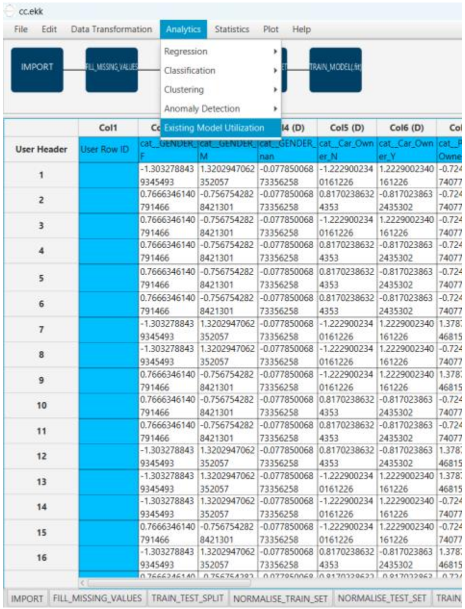

cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
IMPORT
FILL_MISSING_VALUES
TRAIN_TEST_SPLIT
NORMALISE_TRAIN_SET
TRAIN_MODEL(.fit)
User Header
User Row ID
label
Prediction
IMPORT | FILL_MISSING_VALUES | TRAIN_TEST_SPLIT | NORMALISE_TRAIN_SET | NORMALISE_TEST_SET | TRAIN_MODEL(.fit) | +

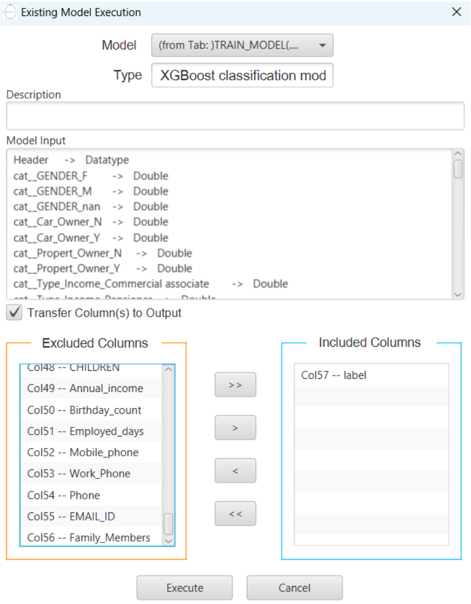
Step 8: Validate the model

Create a new tab by pressing the "+" button on the bottom of the page with the name "VALIDATE_MODEL(.predict)".

Import data into the input spreadsheet of the "VALIDATE_MODEL(.predict)" tab from the output of the "NORMALISE _TEST_SET" tab by right-clicking on the input spreadsheet and then choosing "Import from SpreadSheet".

To validate the model: "Analytics" → "Existing Model Utilization". Then choose Model "(from Tab:) TRAIN_MODEL (.fit)". and transfer the "label" column to the output.


cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
Regression
Classification
Clustering
Anomaly Detection
Existing Model Utilization
IMPORT
FILL_MISSING_VALUES
TRAIN_MODEL(.fit)
User Header
User Row ID
IMPORT | FILL_MISSING_VALUES | TRAIN_TEST_SPLIT | NORMALISE_TRAIN_SET | NORMALISE_TEST_SET | TRAIN


Existing Model Execution

Model: (from Tab:)TRAIN_MODEL(....

Type: XGBoost classification mod

Description

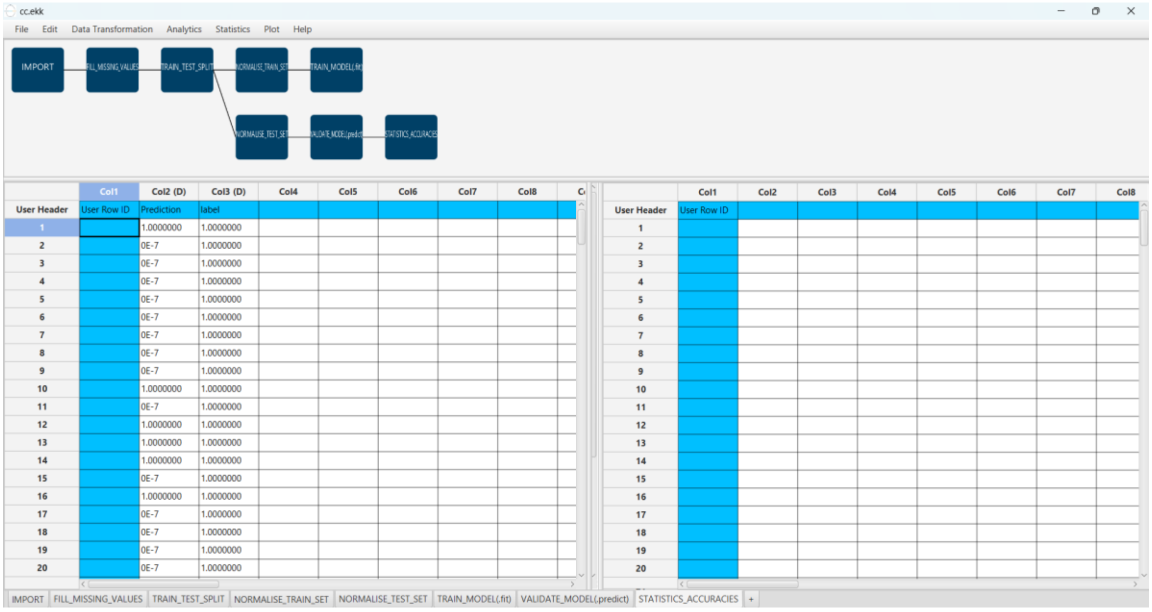
Model Input

```
Header    ->  Datatype
cat__GENDER_F      ->  Double
cat__GENDER_M      ->  Double
cat__GENDER_nan    ->  Double
cat__Car_Owner_N   ->  Double
cat__Car_Owner_Y   ->  Double
cat__Propert_Owner_N   ->  Double
cat__Propert_Owner_Y   ->  Double
cat__Type_Income_Commercial associate    ->  Double
```

☑ Transfer Column(s) to Output

Excluded Columns:
- Col48 -- CHILDREN
- Col49 -- Annual_income
- Col50 -- Birthday_count
- Col51 -- Employed_days
- Col52 -- Mobile_phone
- Col53 -- Work_Phone
- Col54 -- Phone
- Col55 -- EMAIL_ID
- Col56 -- Family_Members

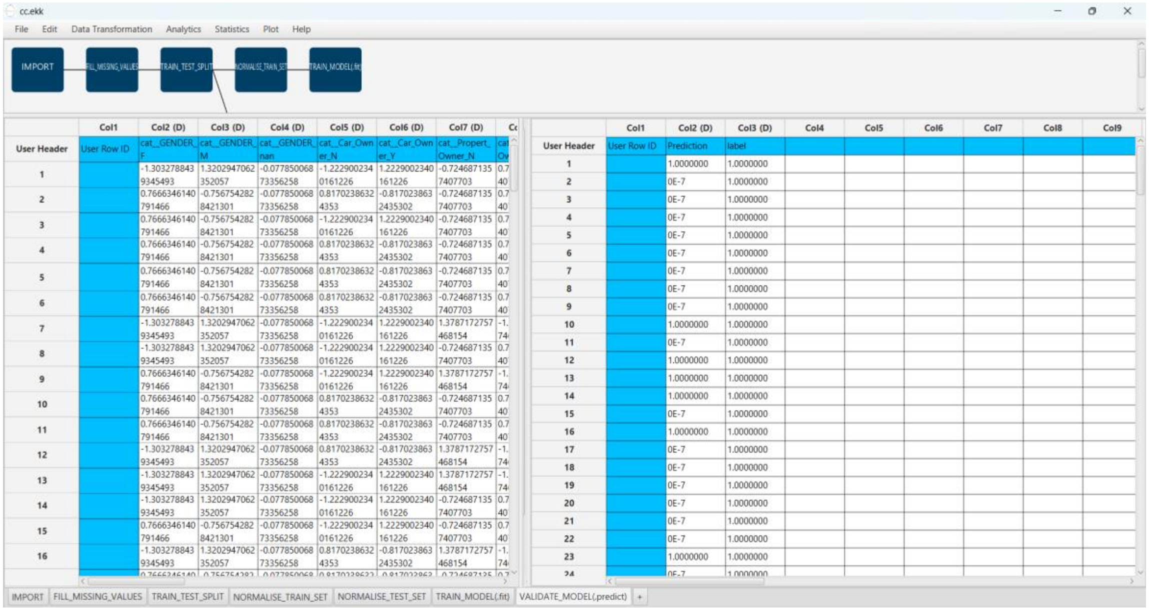
Included Columns:
- Col57 -- label


cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
IMPORT
FILL_MISSING_VALUES
TRAIN_TEST_SPLIT
NORMALISE_TRAIN_SET
TRAIN_MODEL(.fit)
NORMALISE_TEST_SET
VALIDATE_MODEL(.predict)
STATISTICS_ACCURACIES
IMPORT FILL_MISSING_VALUES TRAIN_TEST_SPLIT NORMALISE_TRAIN_SET NORMALISE_TEST_SET TRAIN_MODEL(.fit) VALIDATE_MODEL(.predict) STATISTICS_ACCURACIES +

	Col1	Col2 (D)	Col3 (D)	Col4	Col5	Col6	Col7	Col8
User Header	User Row ID	Prediction	label					
1		1.0000000	1.0000000					
2		0E-7	1.0000000					
3		0E-7	1.0000000					
4		0E-7	1.0000000					
5		0E-7	1.0000000					
6		0E-7	1.0000000					
7		0E-7	1.0000000					
8		0E-7	1.0000000					
9		0E-7	1.0000000					
10		1.0000000	1.0000000					
11		0E-7	1.0000000					
12		1.0000000	1.0000000					
13		1.0000000	1.0000000					
14		1.0000000	1.0000000					
15		0E-7	1.0000000					
16		1.0000000	1.0000000					
17		0E-7	1.0000000					
18		0E-7	1.0000000					
19		0E-7	1.0000000					
20		0E-7	1.0000000					

	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8
User Header	User Row ID							
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								

The predictions will appear on the output spreadsheet.


cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
IMPORT
FILL_MISSING_VALUES
TRAIN_TEST_SPLIT
NORMALISE_TRAIN_SET
TRAIN_MODEL(.fit)
IMPORT FILL_MISSING_VALUES TRAIN_TEST_SPLIT NORMALISE_TRAIN_SET NORMALISE_TEST_SET TRAIN_MODEL(.fit) VALIDATE_MODEL(.predict) +

	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Cc
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat Ov
1		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7 40
2		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7 40
3		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7 40
4		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7 40
5		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7 40
6		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7 40
7		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1. 74
8		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7 40
9		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1. 74
10		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7 40
11		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7 40
12		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	0.81702386324353	-0.8170238632435302	1.3787172757468154	-1. 74
13		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1. 74
14		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7 40
15		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7 40
16		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	0.81702386324353	-0.8170238632435302	1.3787172757468154	-1. 74

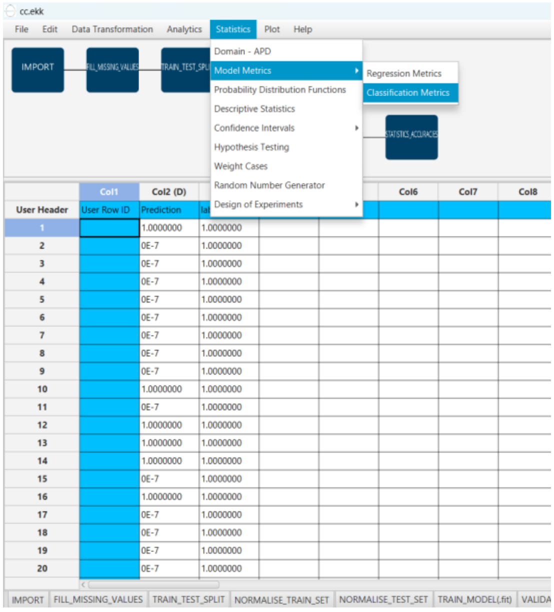
	Col1	Col2 (D)	Col3 (D)	Col4	Col5	Col6	Col7	Col8	Col9
User Header	User Row ID	Prediction	label						
1		1.0000000	1.0000000						
2		0E-7	1.0000000						
3		0E-7	1.0000000						
4		0E-7	1.0000000						
5		0E-7	1.0000000						
6		0E-7	1.0000000						
7		0E-7	1.0000000						
8		0E-7	1.0000000						
9		0E-7	1.0000000						
10		1.0000000	1.0000000						
11		0E-7	1.0000000						
12		1.0000000	1.0000000						
13		1.0000000	1.0000000						
14		1.0000000	1.0000000						
15		0E-7	1.0000000						
16		1.0000000	1.0000000						
17		0E-7	1.0000000						
18		0E-7	1.0000000						
19		0E-7	1.0000000						
20		0E-7	1.0000000						
21		0E-7	1.0000000						
22		0E-7	1.0000000						
23		1.0000000	1.0000000						
24		0E-7	1.0000000						

Step 9: Statistics calculation

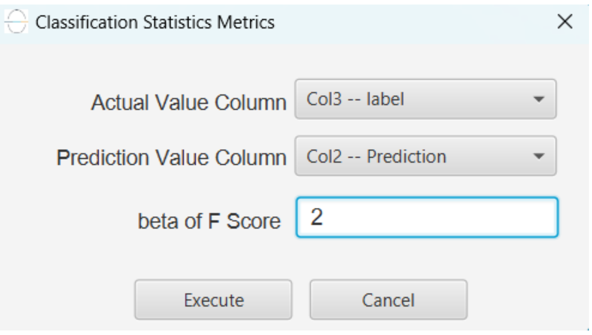
Create a new tab by pressing the "+" button on the bottom of the page with the name "STATISTICS_ACCURACIES".

Import data into the input spreadsheet of the "STATISTICS_ACCURACIES" tab from the output of the "VALIDATE_MODEL(.predict)" tab by right-clicking on the input spreadsheet and then choosing "Import from SpreadSheet".

Calculate the statistical metrics for the classification by browsing: "Statistics" → "Model Metrics" → "Classification Metrics".



	Col1	Col2 (D)				Col6	Col7	Col8
User Header	User Row ID	Prediction	la					
1		1.0000000	1.0000000					
2		0E-7	1.0000000					
3		0E-7	1.0000000					
4		0E-7	1.0000000					
5		0E-7	1.0000000					
6		0E-7	1.0000000					
7		0E-7	1.0000000					
8		0E-7	1.0000000					
9		0E-7	1.0000000					
10		1.0000000	1.0000000					
11		0E-7	1.0000000					
12		1.0000000	1.0000000					
13		1.0000000	1.0000000					
14		1.0000000	1.0000000					
15		0E-7	1.0000000					
16		1.0000000	1.0000000					
17		0E-7	1.0000000					
18		0E-7	1.0000000					
19		0E-7	1.0000000					
20		0E-7	1.0000000					


Classification Statistics Metrics

Actual Value Column: Col3 -- label

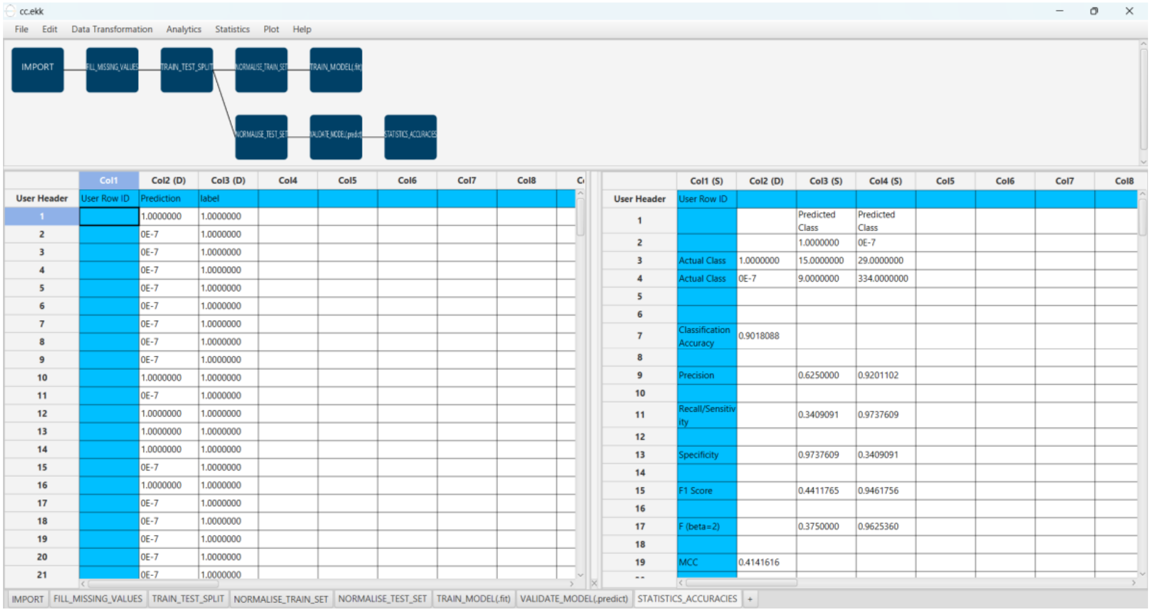
Prediction Value Column: Col2 -- Prediction

beta of F Score: 2

The results will appear on the output spreadsheet.

Accuracy: 0.902

F1-Score = 0.694



	Col1	Col2 (D)	Col3 (D)	Col4	Col5	Col6	Col7	Col8
User Header	User Row ID	Prediction	label					
1		1.0000000	1.0000000					
2		0E-7	1.0000000					
3		0E-7	1.0000000					
4		0E-7	1.0000000					
5		0E-7	1.0000000					
6		0E-7	1.0000000					
7		0E-7	1.0000000					
8		0E-7	1.0000000					
9		0E-7	1.0000000					
10		1.0000000	1.0000000					
11		0E-7	1.0000000					
12		1.0000000	1.0000000					
13		1.0000000	1.0000000					
14		1.0000000	1.0000000					
15		0E-7	1.0000000					
16		1.0000000	1.0000000					
17		0E-7	1.0000000					
18		0E-7	1.0000000					
19		0E-7	1.0000000					
20		0E-7	1.0000000					
21		0E-7	1.0000000					

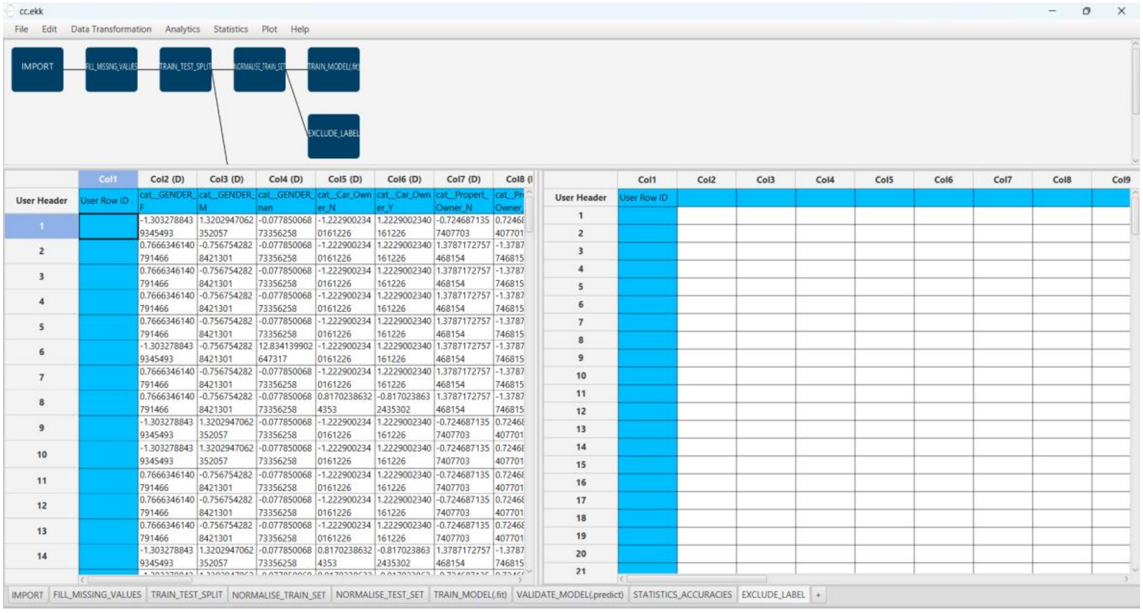
	Col1 (S)	Col2 (D)	Col3 (S)	Col4 (S)	Col5	Col6	Col7	Col8
User Header	User Row ID							
1			Predicted Class	Predicted Class				
2			1.0000000	0E-7				
3	Actual Class	1.0000000	15.0000000	29.0000000				
4	Actual Class	0E-7	9.0000000	334.0000000				
5								
6								
7	Classification Accuracy	0.9018088						
8								
9	Precision		0.6250000	0.9201102				
10								
11	Recall/Sensitivity		0.3409091	0.9737609				
12								
13	Specificity		0.9737609	0.3409091				
14								
15	F1 Score		0.4411765	0.9461756				
16								
17	F (beta=2)		0.3750000	0.9625360				
18								
19	MCC	0.4141616						

Step 10: Reliability check of each record of the test set

Step 10.a: Create the domain

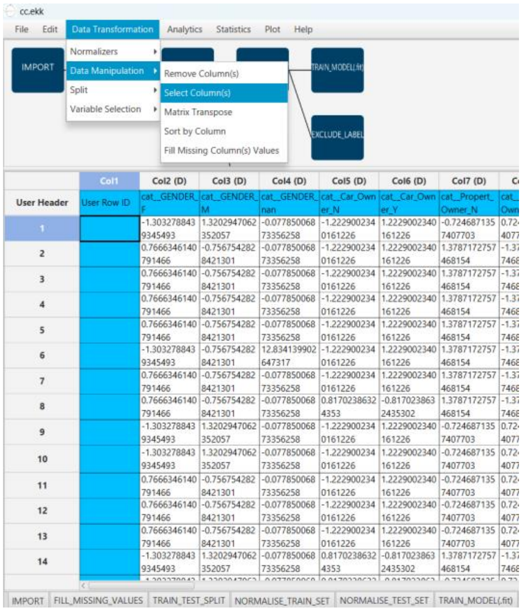
Create a new tab by pressing the "+" button on the bottom of the page with the name "EXCLUDE_LABEL".

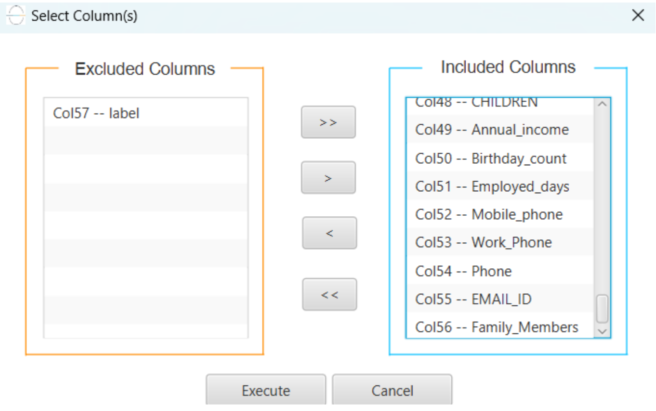
Import data into the input spreadsheet of the "EXCLUDE_LABEL" tab from the output of the "NORMALISE_TRAIN_SET" tab by right-clicking on the input spreadsheet and then choosing "Import from SpreadSheet".


cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
IMPORT
FILL_MISSING_VALUES
TRAIN_TEST_SPLIT
NORMALISE_TRAIN_SET
TRAIN_MODEL(.fit)
EXCLUDE_LABEL
User Header
User Row ID
Col1 Col2 Col3 Col4 Col5 Col6 Col7 Col8 Col9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
IMPORT | FILL_MISSING_VALUES | TRAIN_TEST_SPLIT | NORMALISE_TRAIN_SET | NORMALISE_TEST_SET | TRAIN_MODEL(.fit) | VALIDATE_MODEL(.predict) | STATISTICS_ACCURACIES | EXCLUDE_LABEL | +

	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (I
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Pr Owner_
1		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246 407701
2		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787 746815
3		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787 746815
4		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787 746815
5		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787 746815
6		-1.3032788439345493	-0.7567542828421301	12.834139902647317	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787 746815
7		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787 746815
8		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	1.3787172757468154	-1.3787 746815
9		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246 407701
10		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246 407701
11		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246 407701
12		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246 407701
13		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246 407701
14		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	0.81702386324353	-0.8170238632435302	1.3787172757468154	-1.3787 746815

Manipulate the data to exclude the column that corresponds to the "label" by browsing: "Data Transformation" → "Data Manipulation" → "Select Columns". Then select all the columns except the "label".


cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
Normalizers
Data Manipulation
Split
Variable Selection
Remove Column(s)
Select Column(s)
Matrix Transpose
Sort by Column
Fill Missing Column(s) Values
IMPORT
TRAIN_MODEL(.fit)
EXCLUDE_LABEL
IMPORT | FILL_MISSING_VALUES | TRAIN_TEST_SPLIT | NORMALISE_TRAIN_SET | NORMALISE_TEST_SET | TRAIN_MODEL(.fit)



Excluded Columns

- Col57 -- label

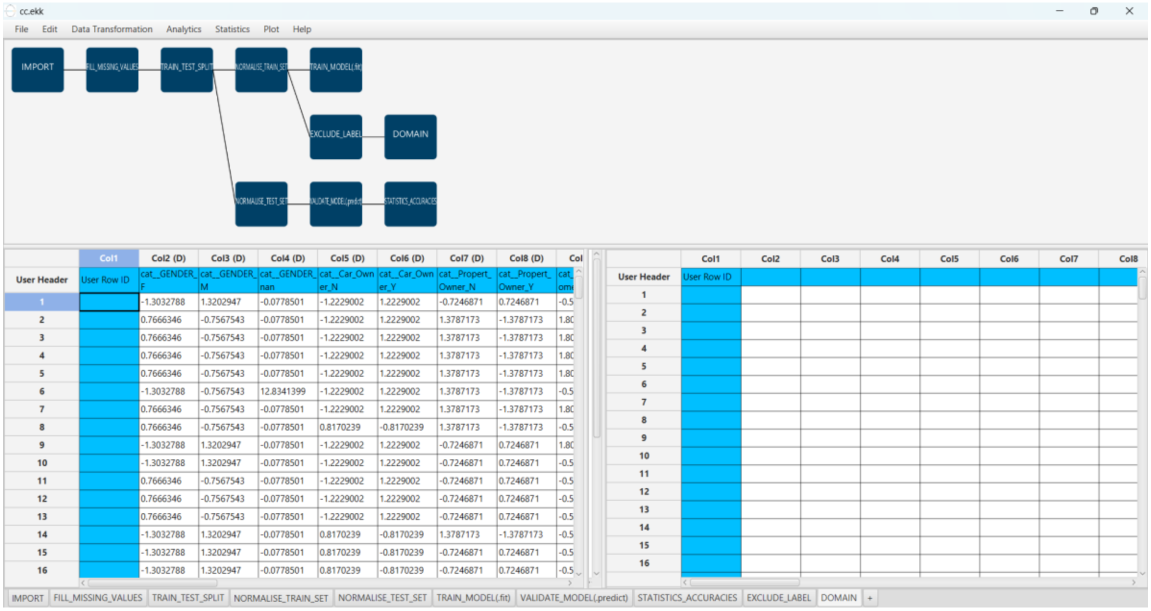
Included Columns

- Col48 -- CHILDREN
- Col49 -- Annual_income
- Col50 -- Birthday_count
- Col51 -- Employed_days
- Col52 -- Mobile_phone
- Col53 -- Work_Phone
- Col54 -- Phone
- Col55 -- EMAIL_ID
- Col56 -- Family_Members

The results will appear on the output spreadsheet.

Create a new tab by pressing the "+" button on the bottom of the page with the name "DOMAIN".

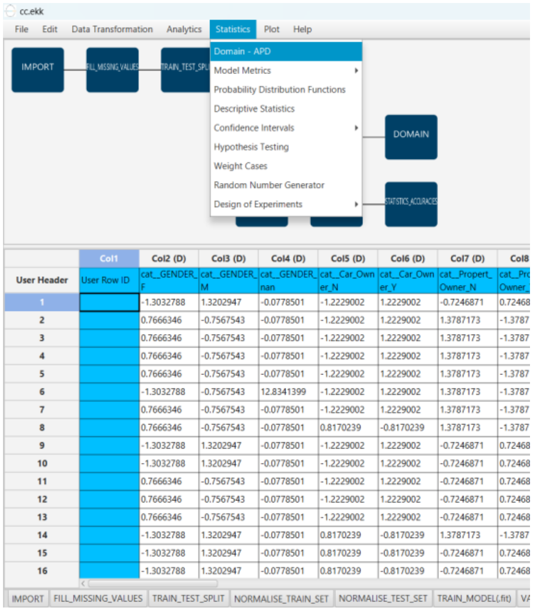
Import data into the input spreadsheet of the "DOMAIN" tab from the output of the "EXCLUDE_LABEL" tab by right-clicking on the input spreadsheet and then choosing "Import from SpreadSheet".



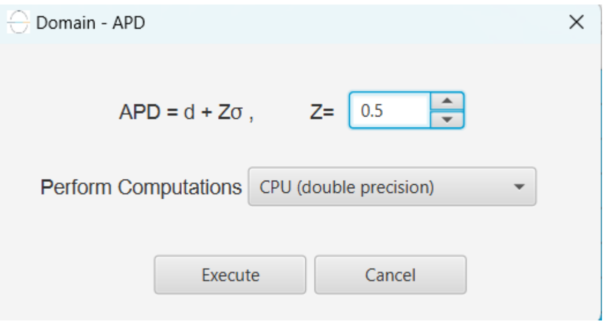
	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (D)	Col
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat_ome
1		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
2		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
3		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
4		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
5		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
6		-1.3032788	-0.7567543	12.8341399	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.5
7		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
8		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.5
9		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.80
10		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
11		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
12		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
13		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
14		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.5
15		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.5
16		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.5

	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8
User Header	User Row ID							
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

Create the domain by browsing: "Statistics" → "Domain APD".



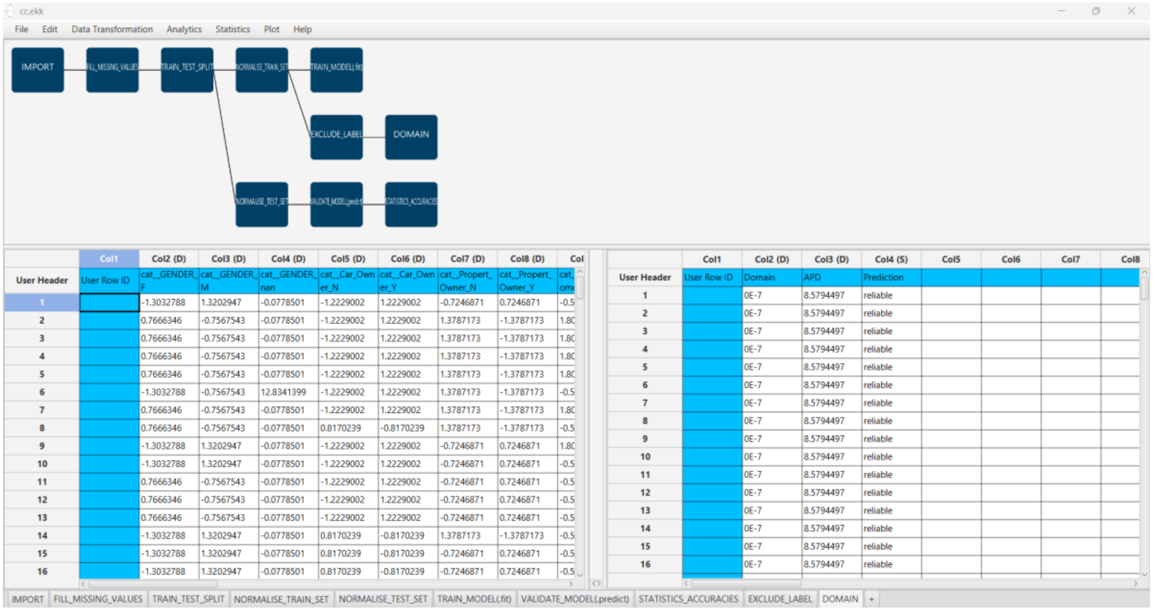
	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Pr Owner_
1		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.72468
2		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787
3		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787
4		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787
5		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787
6		-1.3032788	-0.7567543	12.8341399	-1.2229002	1.2229002	1.3787173	-1.3787
7		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787
8		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787
9		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.72468
10		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.72468
11		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.72468
12		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.72468
13		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.72468
14		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787
15		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.72468
16		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.72468



APD = d + Zσ , Z= 0.5

Perform Computations CPU (double precision)

The results will appear on the output spreadsheet.


cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
IMPORT
FILL_MISSING_VALUES
TRAIN_TEST_SPLIT
NORMALISE_TRAIN_SET
TRAIN_MODEL(.fit)
EXCLUDE_LABEL
DOMAIN
NORMALISE_TEST_SET
VALIDATE_MODEL(.predict)
STATISTICS_ACCURACIES
IMPORT | FILL_MISSING_VALUES | TRAIN_TEST_SPLIT | NORMALISE_TRAIN_SET | NORMALISE_TEST_SET | TRAIN_MODEL(.fit) | VALIDATE_MODEL(.predict) | STATISTICS_ACCURACIES | EXCLUDE_LABEL | DOMAIN | +

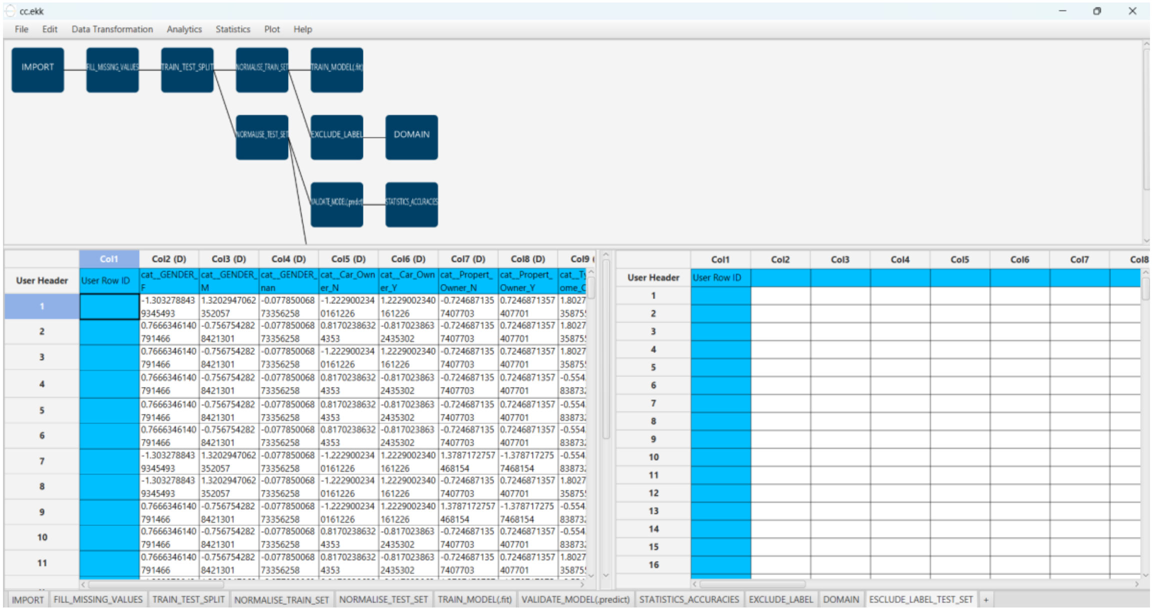
	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (D)	Col
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat_ome
1		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
2		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
3		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
4		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
5		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
6		-1.3032788	-0.7567543	12.8341399	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.5
7		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	1.80
8		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.5
9		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.80
10		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
11		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
12		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
13		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.5
14		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.5
15		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.5
16		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.5

	Col1	Col2 (D)	Col3 (D)	Col4 (S)	Col5	Col6	Col7	Col8
User Header	User Row ID	Domain	APD	Prediction				
1		0E-7	8.5794497	reliable				
2		0E-7	8.5794497	reliable				
3		0E-7	8.5794497	reliable				
4		0E-7	8.5794497	reliable				
5		0E-7	8.5794497	reliable				
6		0E-7	8.5794497	reliable				
7		0E-7	8.5794497	reliable				
8		0E-7	8.5794497	reliable				
9		0E-7	8.5794497	reliable				
10		0E-7	8.5794497	reliable				
11		0E-7	8.5794497	reliable				
12		0E-7	8.5794497	reliable				
13		0E-7	8.5794497	reliable				
14		0E-7	8.5794497	reliable				
15		0E-7	8.5794497	reliable				
16		0E-7	8.5794497	reliable				

Step 10.b: Check the test set reliability

Create a new tab by pressing the "+" button on the bottom of the page with the name "EXCLUDE_LABEL_TEST_SET".

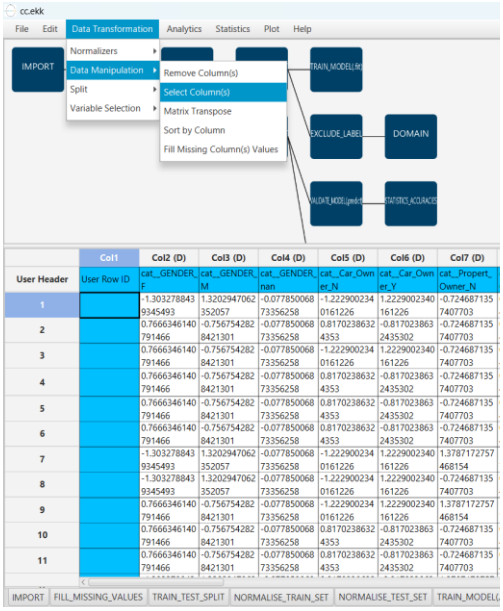
Import data into the input spreadsheet of the "EXCLUDE_LABEL_TEST_SET" tab from the output of the "NORMALISE _TEST_SET" tab by right-clicking on the input spreadsheet and then choosing "Import from SpreadSheet".


cc.ekk
File Edit Data Transformation Analytics Statistics Plot Help
IMPORT
FILL_MISSING_VALUES
TRAIN_TEST_SPLIT
NORMALISE_TRAIN_SET
TRAIN_MODEL(.fit)
NORMALISE_TEST_SET
EXCLUDE_LABEL
DOMAIN
VALIDATE_MODEL(.predict)
STATISTICS_ACCURACIES
IMPORT | FILL_MISSING_VALUES | TRAIN_TEST_SPLIT | NORMALISE_TRAIN_SET | NORMALISE_TEST_SET | TRAIN_MODEL(.fit) | VALIDATE_MODEL(.predict) | STATISTICS_ACCURACIES | EXCLUDE_LABEL | DOMAIN | ESCLUDE_LABEL_TEST_SET | +

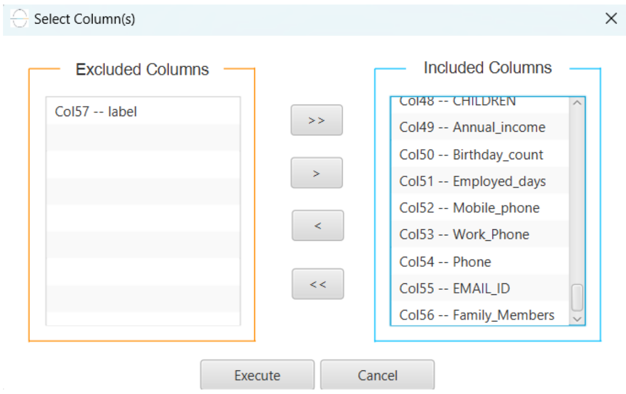
	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (D)	Col9
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat__Tyome_C
1		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	1.802735875
2		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	1.802735875
3		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	1.802735875
4		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
5		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
6		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
7		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	-0.55483873
8		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703	0.7246871357407701	1.802735875
9		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154	-1.3787172757468154	-0.55483873
10		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	-0.55483873
11		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703	0.7246871357407701	1.802735875

	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8
User Header	User Row ID							
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								

Filter the data to exclude the column that corresponds to the "label" by browsing: "Data Transformation" → "Data Manipulation" → "Select Columns". Then select all the columns except "label".



	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N
1		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703
2		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703
3		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703
4		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703
5		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703
6		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703
7		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154
8		-1.3032788439345493	1.3202947062352057	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	-0.7246871357407703
9		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	-1.2229002340161226	1.2229002340161226	1.3787172757468154
10		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703
11		0.7666346140791466	-0.7567542828421301	-0.07785006873356258	0.81702386324353	-0.8170238632435302	-0.7246871357407703



Excluded Columns

Col57 -- label

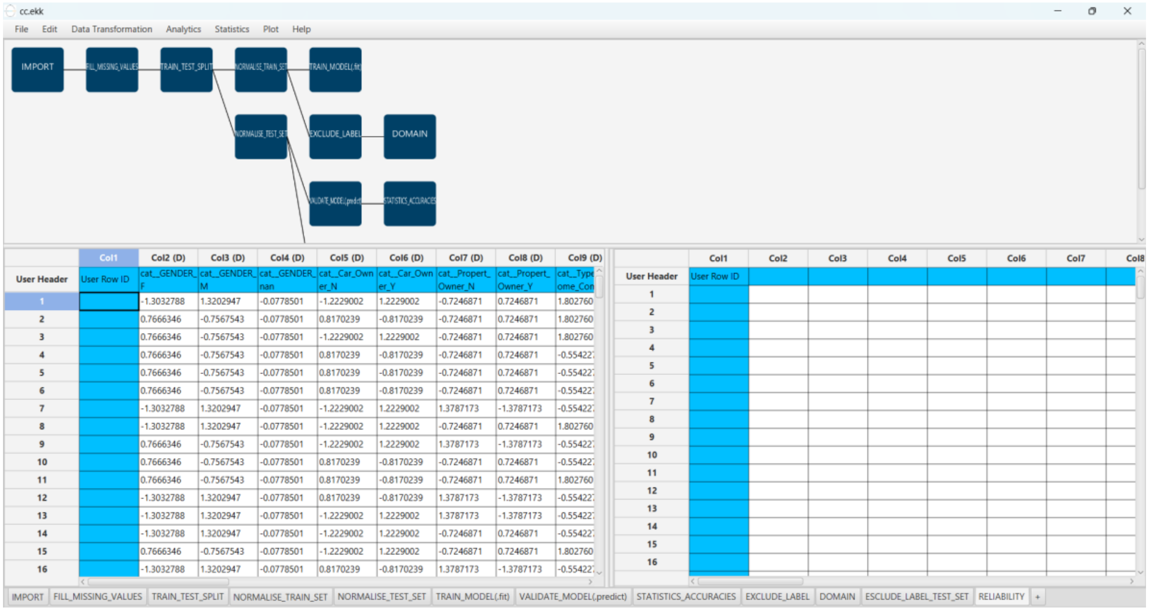
Included Columns

Col48 -- CHILDREN
Col49 -- Annual_income
Col50 -- Birthday_count
Col51 -- Employed_days
Col52 -- Mobile_phone
Col53 -- Work_Phone
Col54 -- Phone
Col55 -- EMAIL_ID
Col56 -- Family_Members

The results will appear on the output spreadsheet.

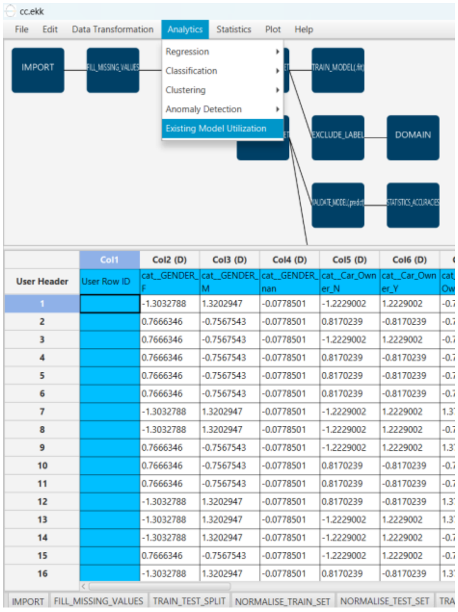
Create a new tab by pressing the "+" button on the bottom of the page with the name "RELIABILITY".

Import data into the input spreadsheet of the "RELIABILITY" tab from the output of the "EXCLUDE_LABEL_TEST_SET" tab by right-clicking on the input spreadsheet and then choosing "Import from SpreadSheet".

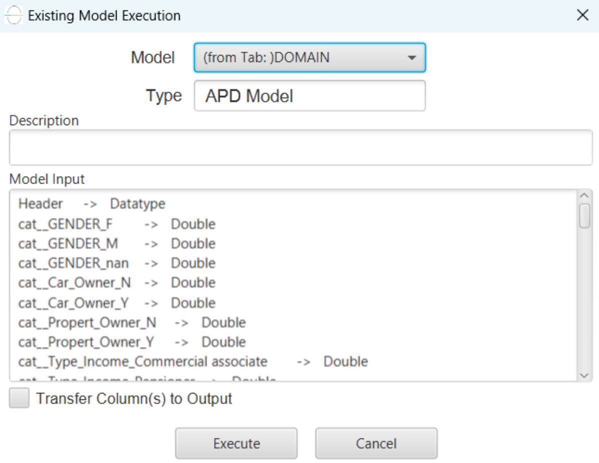


	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (D)	Col9 (D)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat__Type ome_Con
1		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
2		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	1.802760
3		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
4		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.554227
5		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.554227
6		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.554227
7		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.554227
8		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
9		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.554227
10		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.554227
11		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	1.802760
12		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.554227
13		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.554227
14		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.554227
15		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
16		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.554227

Check the Reliability by browsing: "Analytics" → "Existing Model Utilization". Then select as Model "(from Tab:) DOMAIN".



	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y
1		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002
2		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239
3		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002
4		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239
5		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239
6		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239
7		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002
8		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002
9		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002
10		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239
11		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239
12		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239
13		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002
14		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002
15		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002
16		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239



Model: (from Tab:)DOMAIN

Type: APD Model

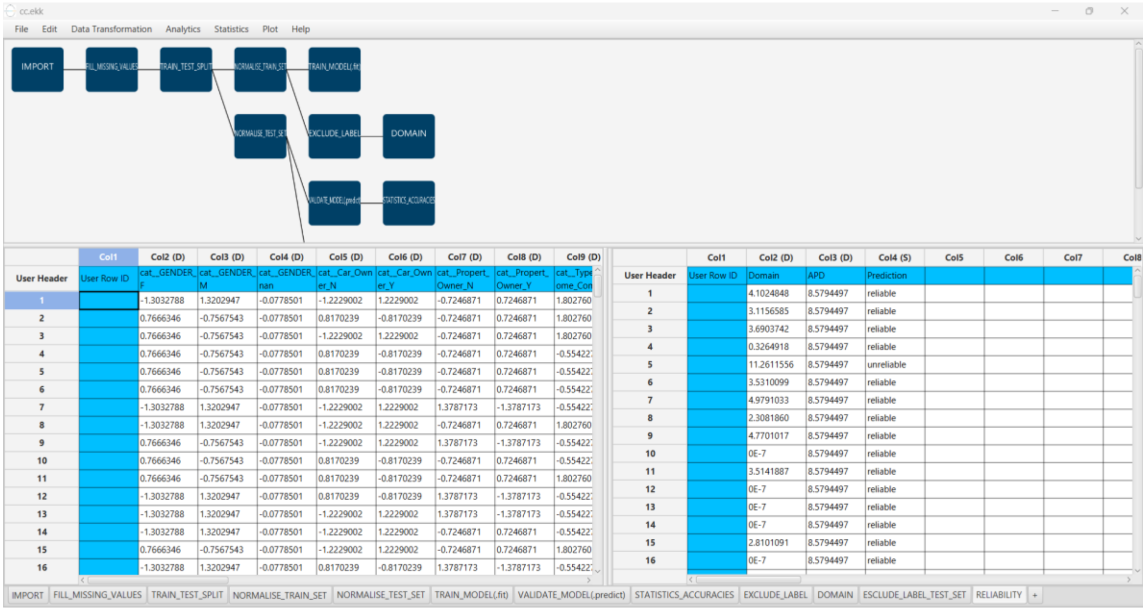
Description

Model Input

```
Header    ->  Datatype
cat__GENDER_F      ->  Double
cat__GENDER_M      ->  Double
cat__GENDER_nan    ->  Double
cat__Car_Owner_N   ->  Double
cat__Car_Owner_Y   ->  Double
cat__Propert_Owner_N    ->  Double
cat__Propert_Owner_Y    ->  Double
cat__Type_Income_Commercial associate     ->  Double
```

□ Transfer Column(s) to Output

The results will appear on the output spreadsheet.



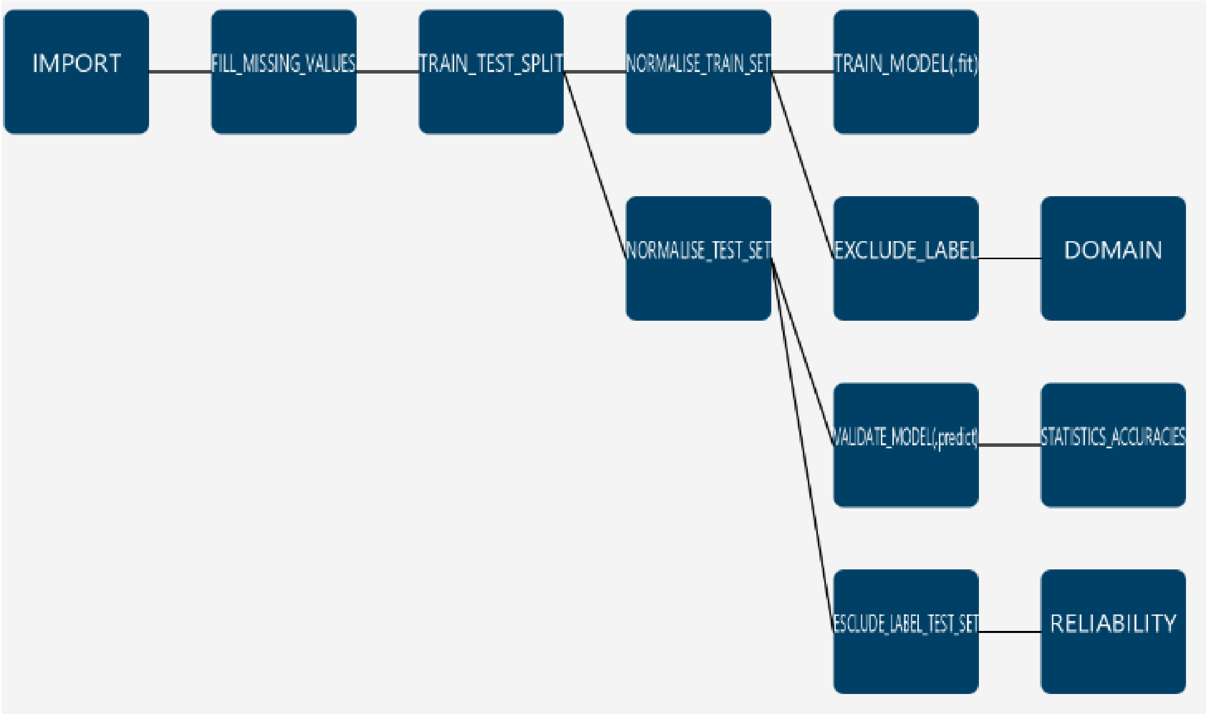
	Col1	Col2 (D)	Col3 (D)	Col4 (D)	Col5 (D)	Col6 (D)	Col7 (D)	Col8 (D)	Col9 (D)
User Header	User Row ID	cat__GENDER_F	cat__GENDER_M	cat__GENDER_nan	cat__Car_Owner_N	cat__Car_Owner_Y	cat__Propert_Owner_N	cat__Propert_Owner_Y	cat__Type ome_Con
1		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
2		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	1.802760
3		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
4		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.55422
5		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.55422
6		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.55422
7		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.55422
8		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
9		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.55422
10		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	-0.55422
11		0.7666346	-0.7567543	-0.0778501	0.8170239	-0.8170239	-0.7246871	0.7246871	1.802760
12		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.55422
13		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	1.3787173	-1.3787173	-0.55422
14		-1.3032788	1.3202947	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	-0.55422
15		0.7666346	-0.7567543	-0.0778501	-1.2229002	1.2229002	-0.7246871	0.7246871	1.802760
16		-1.3032788	1.3202947	-0.0778501	0.8170239	-0.8170239	1.3787173	-1.3787173	-0.55422

	Col1	Col2 (D)	Col3 (D)	Col4 (S)	Col5	Col6	Col7	Col8
User Header	User Row ID	Domain	APD	Prediction				
1		4.1024848	8.5794497	reliable				
2		3.1156585	8.5794497	reliable				
3		3.6903742	8.5794497	reliable				
4		0.3264918	8.5794497	reliable				
5		11.2611556	8.5794497	unreliable				
6		3.5310099	8.5794497	reliable				
7		4.9791033	8.5794497	reliable				
8		2.3081860	8.5794497	reliable				
9		4.7701017	8.5794497	reliable				
10		0E-7	8.5794497	reliable				
11		3.5141887	8.5794497	reliable				
12		0E-7	8.5794497	reliable				
13		0E-7	8.5794497	reliable				
14		0E-7	8.5794497	reliable				
15		2.8101091	8.5794497	reliable				
16		0E-7	8.5794497	reliable				

There are four unreliable samples in the test set.

Final Isalos Workflow

Following the above-described steps, the final workflow on Isalos will look like this:


IMPORT
FILL_MISSING_VALUES
TRAIN_TEST_SPLIT
NORMALISE_TRAIN_SET
TRAIN_MODEL(.fit)
NORMALISE_TEST_SET
EXCLUDE_LABEL
DOMAIN
VALIDATE_MODEL(.predict)
STATISTICS_ACCURACIES
ESCLUDE_LABEL_TEST_SET
RELIABILITY