알테어 날리지 스튜디오 2021.2 버전 출시

날리지 스튜디오란?

업계 최고의 특허 받은 의사 결정 트리, 전략 트리, 워크플로우 및 마법사 중심의 그래픽 사용자 인터페이스로 널리 알려진 날리지 스튜디오는 고급 데이터 마이닝 및 예측 분석 워크 벤치입니다. 날리지 스튜디오는 고급 예측 모델링 기능을 통해 모델 개발 및 구현에 중점을 두고 데이터 마이닝 주기의 모든 단계에 대한 포괄적인 예측 분석을 제공합니다.

왜 날리지 스튜디오인가?

비용 효율적이고 사용하기 쉬운 날리지 스튜디오는 보편성과 효율성면에서 독보적입니다. 고성능의 비주얼 환경은 직관적인 워크플로우와 마법사 중심의 그래픽 사용자 인터페이스를 제공하여 사용자가 코딩하지 않고도 효과적인 모델을 구축할 수 있게 합니다.

날리지 스튜디오는 누가 사용하는가?

날리지 스튜디오는 신용 위험, 사기, 마케팅, 영업, CRM 분석, 엔지니어링과 같은 다양한 산업과 부서의 고객이 사용합니다. 비즈니스 분석가와 계량 분석가(퀀트)는 어떤 복잡한 모델도 수용할 수 있도록 고급 모델 파라미터를 세밀하게 조정하기 위해, 구성 가능한 설정을 통해 보다 복잡한 기능에 액세스할 수 있습니다.

이번 버전에서 꼭 확인해야할 사항

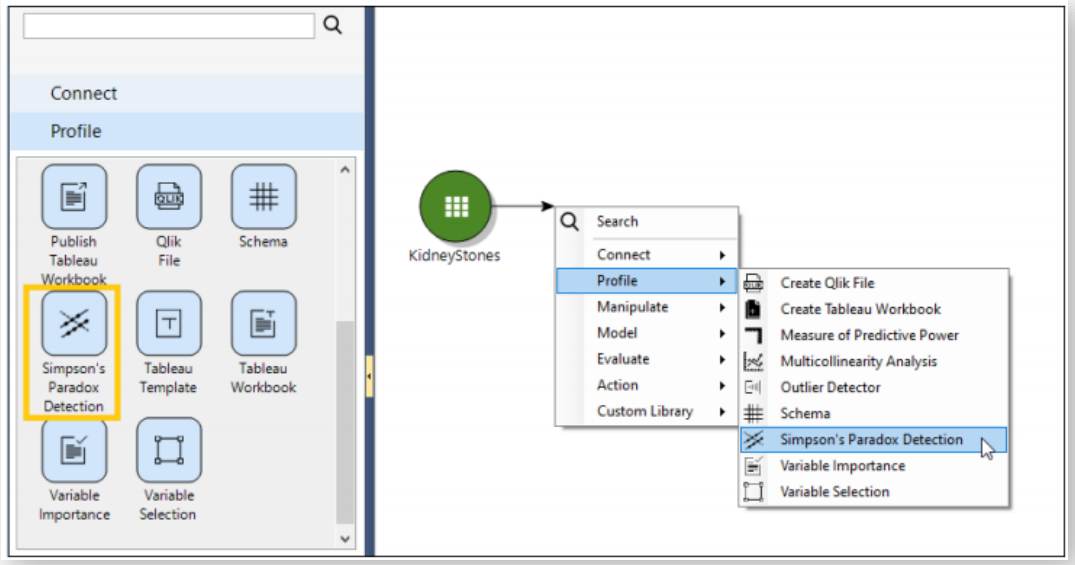
새로운 기능 및 향상된 기능

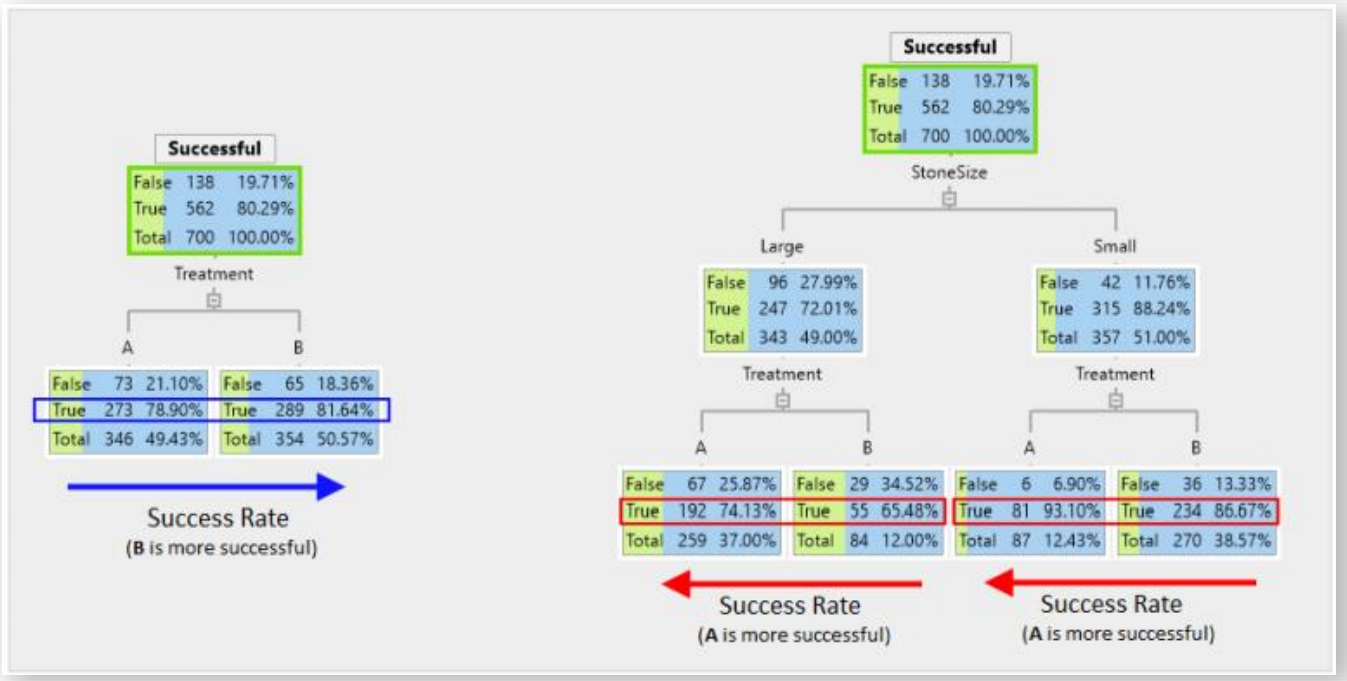
- Simpson's Paradox Detection
- 결측치 대체
- 클래스 불균형 핸들링
- Sklearn GLM

Simpson's Paradox Detection 기능

대상 변수의 특정 추세가 하위 집합에선 관찰되지만 전체 데이터 집합에서는 그 반대의 추세가 관찰되는 Simpson's Paradox를 탐지하는 데이터 프로파일링 기능이 추가되었습니다.

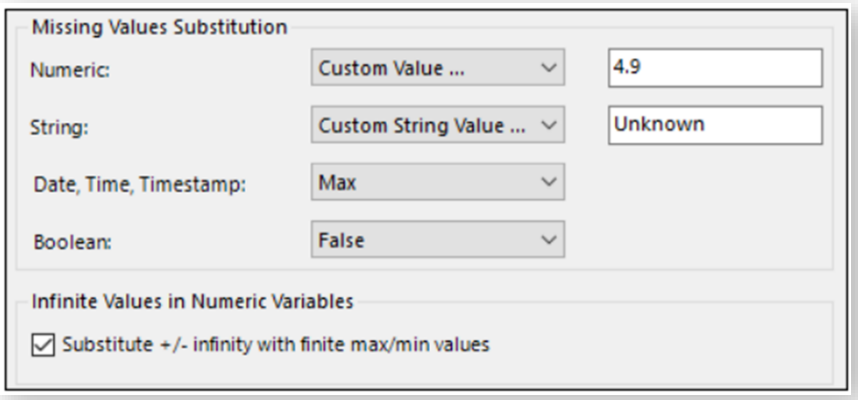
이를 통해 추세에 대한 잘못된 해석을 피하고, 변수 간의 편향이나 인과관계에 대한 잘못된 결론을 도출하는 것을 방지할 수 있습니다.





Substitute Missing Values 기능

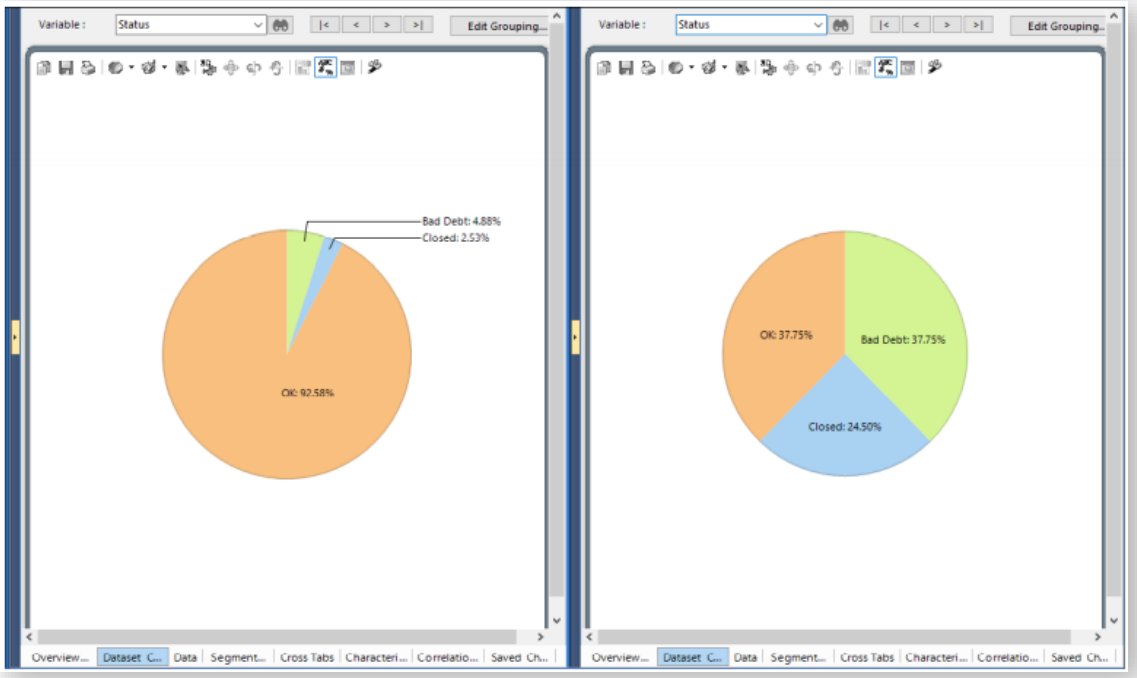
기존에는 Variable Transformation 노드 내에서 helper 기능을 통해서 결측치를 처리한 새로운 컬럼을 생성했습니다. 이 기능은 노드 형태로 따로 존재하고, 원래의 변수값을 대체하는 방법으로 동작합니다.



Handle Class Imbalance 기능: 오버샘플링을 통한 클래스 불균형 핸들링

이 기능은 일부 클래스를 오버샘플링하여 왜곡된 분포를 가진 변수 클래스의 균형을 맞춥니다.

클래스 불균형 처리에 많이 사용되는 Python 라이브러리인 Imbalanced-Learn(imblearn)을 사용합니다.

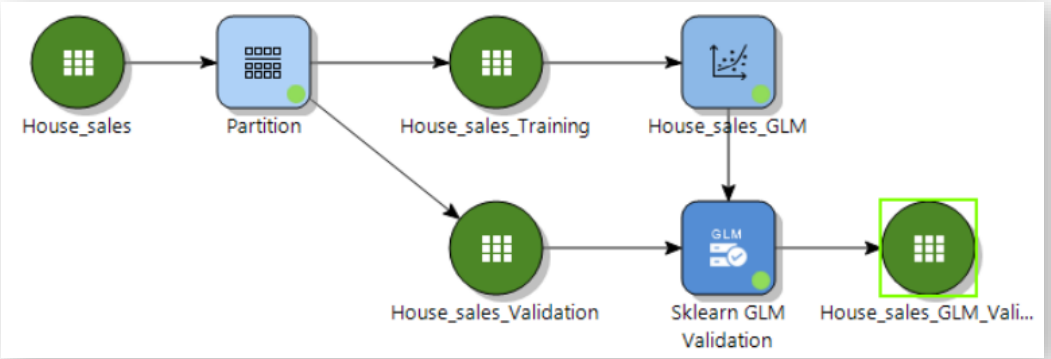


Sklearn GLM(Generalized Linear Model)

학습데이터 셋에 연결하여 새로 모델을 생성하거나 학습된 선형회귀모델에 연결할 수 있는 'Sklearn GLM ' 이 Model 탭에, 이를 검증하고 스코어링할 수 있는 'Sklearn GLM Validation' 노드와 'Sklearn GLM Scoring' 노드가 생성되었습니다.

이를 이용하여 정규 분포를 반드시 따를 필요는 없는 종속변수로 회귀 모델을 구축할 수 있습니다.

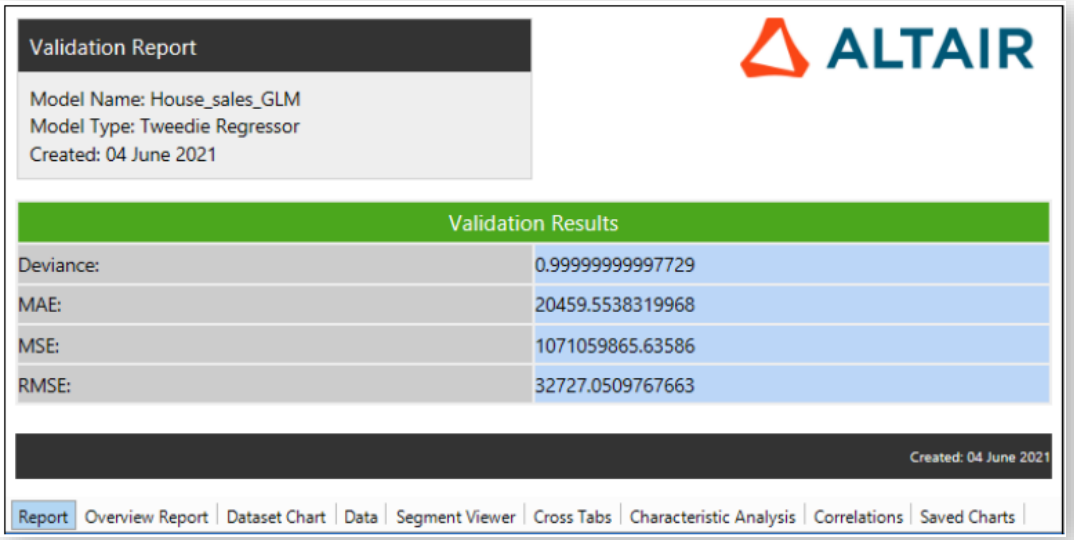
관련 노드들은 머신러닝을 위한 Python 라이브러리인 scikit-learn의 선형 모델 방법을 사용하고 Python 코드로 바로 출력이 가능합니다.



Generalized Linear Model - Model Parameters

```
#! python

import pandas as pd
import numpy as np
from sklearn import preprocessing
import pickle
from sklearn.linear_model import TweedieRegressor, PoissonRegressor, GammaRegressor
from sklearn import metrics
from sklearn.metrics import mean_tweedie_deviance
from sklearn.model_selection import train_test_split
import json

random_state = 1
target_variable = 'Selling_price'
weight = None
alpha = 0
fit_intercept = True
max_iter = 2000
tol = 0.0001

tweedie_powers = [0]
eval_Metric = 'mean_tweedie_deviance'

# load dataset
import DataUtilities
directory = 'House_Sales_GLM'

source = DataUtilities.NativeToDataframe(directory, 'House_sales_Training')
variables = ['Bathrooms', 'Bedrooms', 'Constr_material', 'Depth', 'Frontage', 'Garage_area', 'Garage_type', 'Half_bathrooms',
'Latitude', 'Longitude', 'Lot_size', 'Rooms', 'Sale_date', 'Stories', 'Total_living_area', 'Year_built', 'Selling_price']

train = source[variables]

class NpEncoder(json.JSONEncoder):
    def default(self, obj):
        if isinstance(obj, np.integer):
            return int(obj)
        elif isinstance(obj, np.floating):
            return float(obj)
        elif isinstance(obj, np.ndarray):
            return obj.tolist()
        else:
            return super(NpEncoder, self).default(obj)

def preProcess(train, target_variable, save=True):
    # impute
```

Sklearn GLM Validation



Validation Report

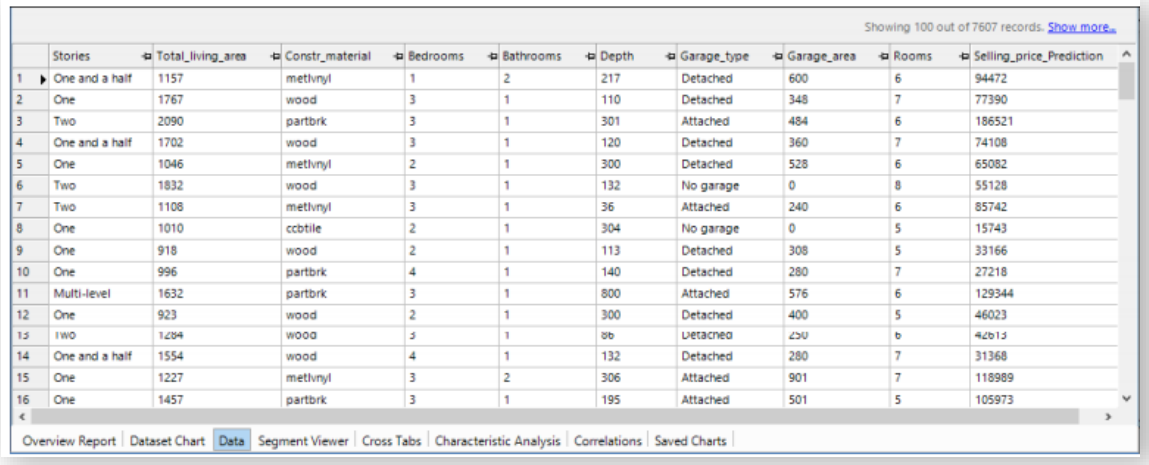
Model Name: House_sales_GLM
Model Type: Tweedie Regressor
Created: 04 June 2021

Validation Results	
Deviance:	0.99999999997729
MAE:	20459.5538319968
MSE:	1071059865.63586
RMSE:	32727.0509767663

Created: 04 June 2021

Report | Overview Report | Dataset Chart | Data | Segment Viewer | Cross Tabs | Characteristic Analysis | Correlations | Saved Charts

Sklearn GLM Scoring



Showing 100 out of 7607 records. Show more...

	Stories	Total_living_area	Constr_material	Bedrooms	Bathrooms	Depth	Garage_type	Garage_area	Rooms	Selling_price_Prediction
1	One and a half	1157	metlvnyl	1	2	217	Detached	600	6	94472
2	One	1767	wood	3	1	110	Detached	348	7	77390
3	Two	2090	partbrk	3	1	301	Attached	484	6	186521
4	One and a half	1702	wood	3	1	120	Detached	360	7	74108
5	One	1046	metlvnyl	2	1	300	Detached	528	6	65082
6	Two	1832	wood	3	1	132	No garage	0	8	55128
7	Two	1108	metlvnyl	3	1	36	Attached	240	6	85742
8	One	1010	ccbtile	2	1	304	No garage	0	5	15743
9	One	918	wood	2	1	113	Detached	308	5	33166
10	One	996	partbrk	4	1	140	Detached	280	7	27218
11	Multi-level	1632	partbrk	3	1	800	Attached	576	6	129344
12	One	923	wood	2	1	300	Detached	400	5	46023
13	Two	1284	wood	3	1	86	Detached	250	6	42613
14	One and a half	1554	wood	4	1	132	Detached	280	7	31368
15	One	1227	metlvnyl	3	2	306	Attached	901	7	118989
16	One	1457	partbrk	3	1	195	Attached	501	5	105973

Overview Report | Dataset Chart | Data | Segment Viewer | Cross Tabs | Characteristic Analysis | Correlations | Saved Charts

기타

- 회귀모델, 정규화, 딥러닝, PCA 등에 대해 Code Generation 노드에서 R 코드 생성이 활성화 되었습니다.
- R 버전 3.4.4 이하 버전은 더 이상 지원되지 않습니다.
- 프로그램 설치 시 자동 설치 옵션을 이용하여 프로그램 기본 설정을 저장하고 사용할 수 있습니다.

날리지 스튜디오에 대한 추가 정보

이 제품의 새 버전에 대한 소프트웨어 및 문서는 Altair Connect에서 이용할 수 있습니다. Altair Connect 의 다운로드 섹션에서 **Data Analytics Products**를 확인하십시오.