featexp

Feature exploration for supervised learning. Helps with feature understanding, identifying noisy features, feature debugging, leakage detection and model monitoring.

Installation

```
pip install featexp
```

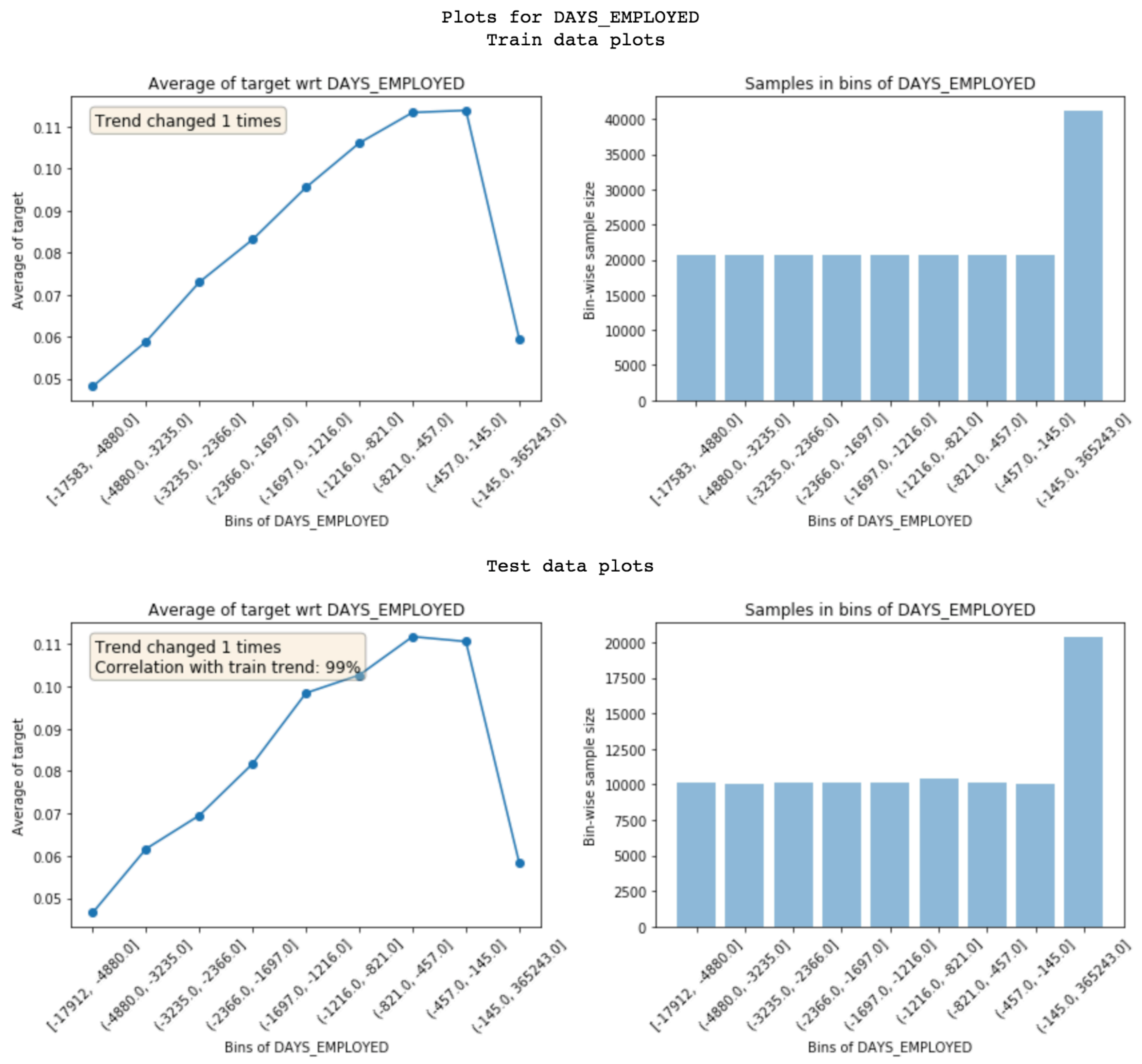
Using featexp

Detailed Medium post on using featexp. Translations from web: Chinese, Russian

featexp draws plots similar to partial dependence plots, but directly from data instead of using a trained model like current implementations of pdp do. Since it draws plots from data directly, it helps with understanding the features well and building better ML models.

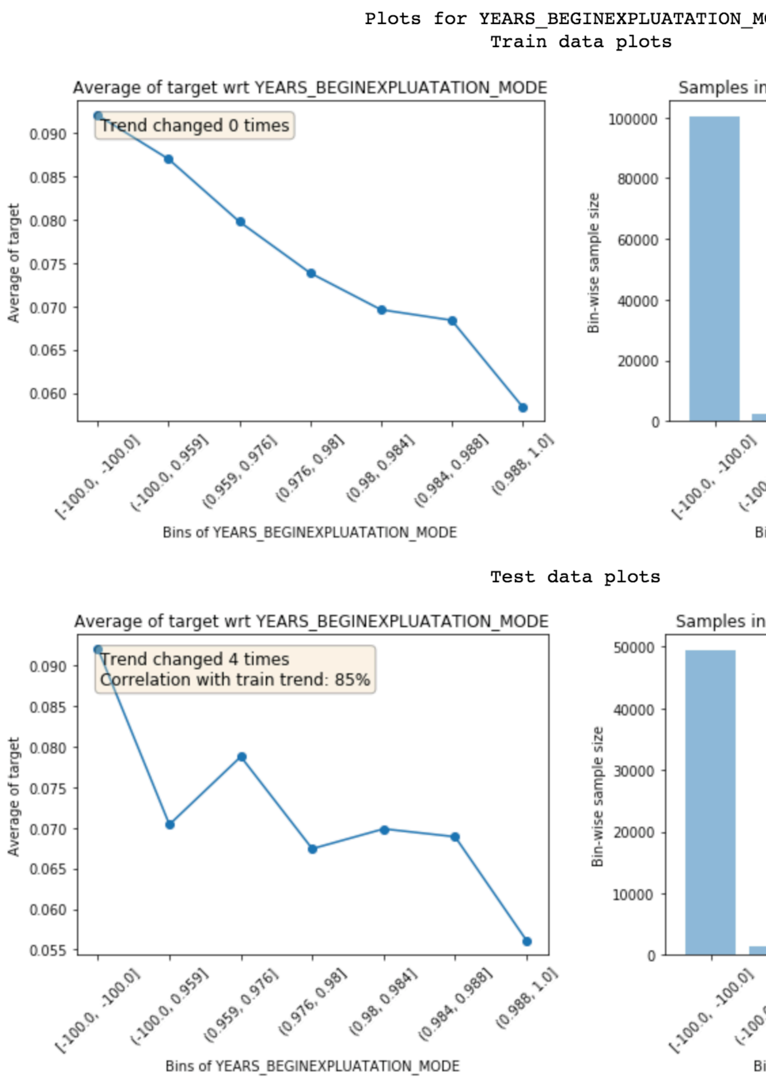
```python
from featexp import get_univariate_plots
get_univariate_plots(data=data_train, target_col='target', data_test=
    data_test, features_list=['DAYS_EMPLOYED'])

# data_test and features_list are optional.
# Draws plots for all columns if features_list not passed
# Draws only train data plots if no test_data passed
```



featexp bins a feature into equal population bins and shows the mean value of the dependent variable (target) in each bin. Here's how to read these plots:

1. The trend plot on the left helps you understand the relationship between target and feature.
2. Population distribution helps you make sure the feature is correct.
3. Also, shows the number of trend direction changes and the correlation between train and test trend which can be used to identify noisy features. A high number of trend changes or low trend correlation implies high noise.



Example of a noisy feature: Has low trend correlation

Getting binned feature stats

Returns mean target and population in each bin of a feature

```
from featexp import univariate_plotter
binned_data_train, binned_data_test = univariate_plotter(data=
    data_train, target_col='target', feature='DAYS_EMPLOYED', data_test=
    data_test)
# For only train data
binned_data_train = univariate_plotter(data=data_train, target_col='
    target', feature='DAYS_EMPLOYED')
```

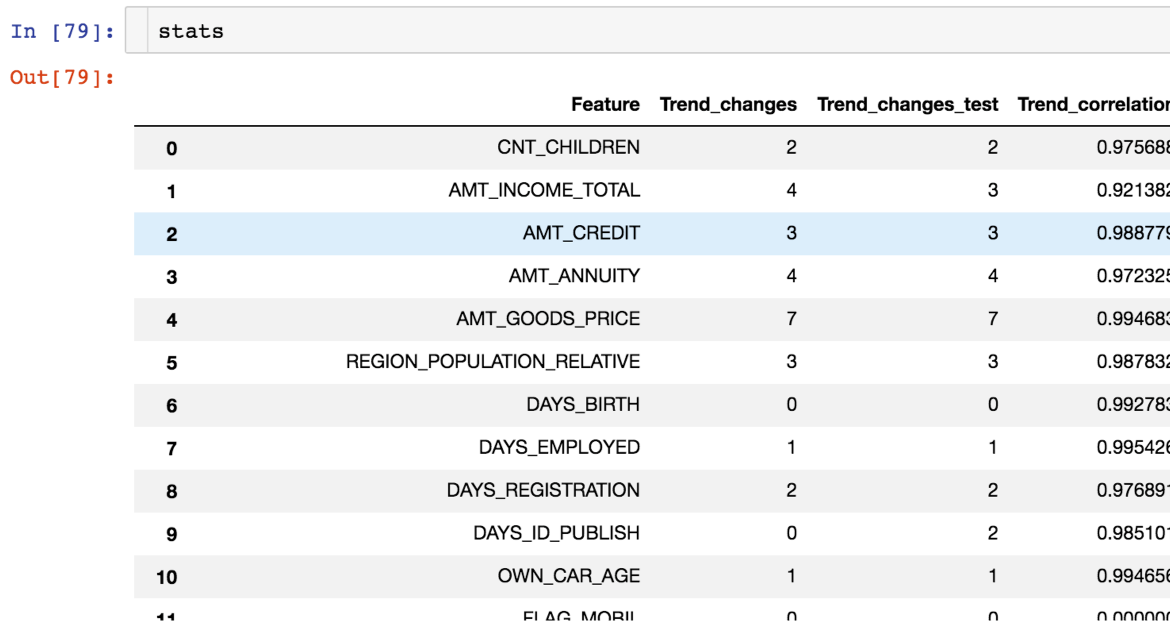
Getting stats for all features

Returns trend changes and trend correlation for all features in a dataframe

```python
from featexp import get_trend_stats_feature
stats = get_trend_stats(data=data_train, target_col='target', data_test
    =data_test)

# data_test is optional. If not passed, trend correlations aren't
    calculated.
```

Returns a dataframe with trend changes and trend correlation which can be used for dropping the noisy features, etc.

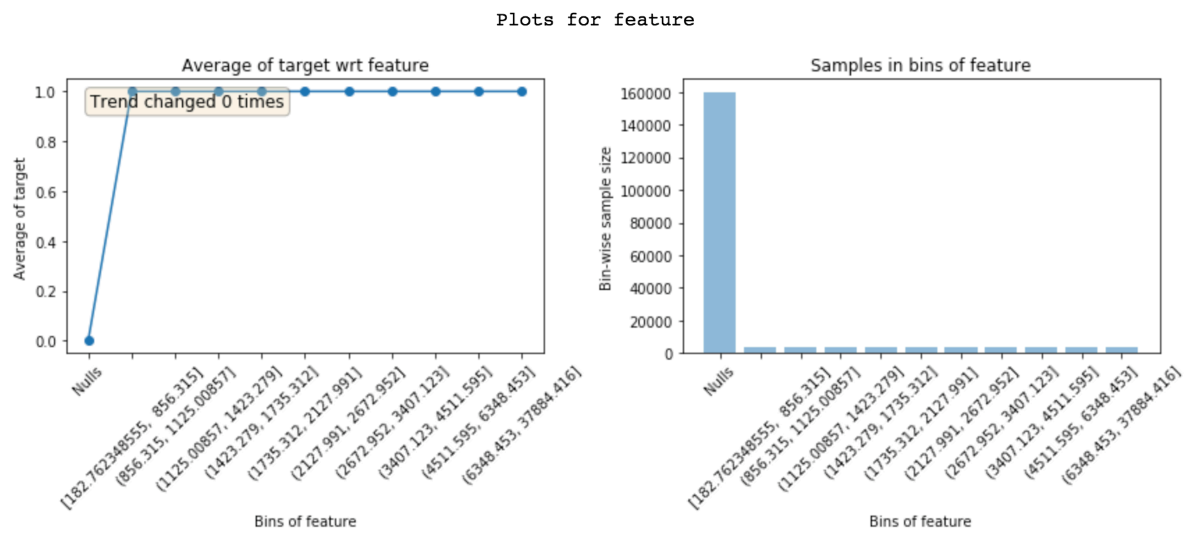


```
In [79]: stats
Out[79]:
```

	Feature	Trend_changes	Trend_changes_test	Trend_correlatio
0	CNT_CHILDREN	2	2	0.97568
1	AMT_INCOME_TOTAL	4	3	0.92138
2	AMT_CREDIT	3	3	0.98877
3	AMT_ANNUITY	4	4	0.97232
4	AMT_GOODS_PRICE	7	7	0.99468
5	REGION_POPULATION_RELATIVE	3	3	0.98783
6	DAYS_BIRTH	0	0	0.99278
7	DAYS_EMPLOYED	1	1	0.99542
8	DAYS_REGISTRATION	2	2	0.97689
9	DAYS_ID_PUBLISH	0	2	0.98510
10	OWN_CAR_AGE	1	1	0.99465
11	FLAG_MOBIL	0	0	0.00000

Leakage detection

It helps with identifying why a feature is leaky which helps with debugging.



Nulls have 0% mean target and 100% mean target in other bins. Implies this feature is populated only for target = 1.

Citing featexp

DOI 10.5281/zenodo.2616757