Cheminformatics in Python [Part 1.3] Predicting Solubility of Molecules using PyCaret | Data Science Project

In this Jupyter notebook, I will continue the cheminformatics by simplifying this notebook via the use of the low-code ML library PyCaret.

1. Install PyCaret

```
! pip install pycaret
```

```
Collecting pycaret
  Downloading pycaret-2.3.6-py3-none-any.whl (301 kB)
     |████████████████████████████████| 301 kB 5.2 MB/s
Collecting pyod
  Downloading pyod-0.9.7.tar.gz (114 kB)
     |████████████████████████████████| 114 kB 51.9 MB/s
Collecting kmodes>=0.10.1
  Downloading kmodes-0.11.1-py2.py3-none-any.whl (19 kB)
Requirement already satisfied: textblob in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: pyyaml<6.0.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: yellowbrick>=1.0.1 in /usr/local/lib/python3.7/dist
Collecting pyLDAvis
  Downloading pyLDAvis-3.3.1.tar.gz (1.7 MB)
     |████████████████████████████████| 1.7 MB 43.9 MB/s
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Installing backend dependencies ... done
    Preparing wheel metadata ... done
Collecting mlxtend>=0.17.0
  Downloading mlxtend-0.19.0-py2.py3-none-any.whl (1.3 MB)
     |████████████████████████████████| 1.3 MB 60.0 MB/s
Collecting scikit-learn==0.23.2
  Downloading scikit_learn-0.23.2-cp37-cp37m-manylinux1_x86_64.whl (6.8 MB)
     |████████████████████████████████| 6.8 MB 47.2 MB/s
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-package
Collecting umap-learn
  Downloading umap-learn-0.5.2.tar.gz (86 kB)
     |████████████████████████████████| 86 kB 7.0 MB/s
Requirement already satisfied: seaborn in /usr/local/lib/python3.7/dist-packages (
Collecting mlflow
  Downloading mlflow-1.23.1-py3-none-any.whl (15.6 MB)
     |████████████████████████████████| 15.6 MB 41.7 MB/s
Collecting Boruta
  Downloading Boruta-0.3-py3-none-any.whl (56 kB)
     |████████████████████████████████| 56 kB 4.8 MB/s
Collecting lightgbm>=2.3.1
  Downloading lightgbm-3.3.2-py3-none-manylinux1_x86_64.whl (2.0 MB)
     |████████████████████████████████| 2.0 MB 45.3 MB/s
Requirement already satisfied: wordcloud in /usr/local/lib/python3.7/dist-packages
Collecting scikit-plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: nltk in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: IPython in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: scipy<=1.5.4 in /usr/local/lib/python3.7/dist-packa
```

```
Collecting imbalanced-learn==0.7.0
  Downloading imbalanced_learn-0.7.0-py3-none-any.whl (167 kB)
     |████████████████████████████████| 167 kB 85.5 MB/s
Requirement already satisfied: plotly>=4.4.1 in /usr/local/lib/python3.7/dist-pack
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (f
Requirement already satisfied: cufflinks>=0.17.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: spacy<2.4.0 in /usr/local/lib/python3.7/dist-packag
Collecting pandas-profiling>=2.8.0
  Downloading pandas_profiling-3.1.0-py2.py3-none-any.whl (261 kB)
     |████████████████████████████████| 261 kB 75.9 MB/s
Requirement already satisfied: gensim<4.0.0 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: ipywidgets in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: joblib in /usr/local/lib/python3.7/dist-packages (f
```

2. Read in dataset

```
import pandas as pd
```

```
delaney_with_descriptors_url = 'https://raw.githubusercontent.com/dataprofessor/data/maste
dataset = pd.read_csv(delaney_with_descriptors_url)
```

```
dataset
```

	MolLogP	MolWt	NumRotatableBonds	AromaticProportion	logS
0	2.59540	167.850	0.0	0.000000	-2.180
1	2.37650	133.405	0.0	0.000000	-2.000
2	2.59380	167.850	1.0	0.000000	-1.740
3	2.02890	133.405	1.0	0.000000	-1.480
4	2.91890	187.375	1.0	0.000000	-3.040
...	...	...	...	...	...
1139	1.98820	287.343	8.0	0.000000	1.144
1140	3.42130	286.114	2.0	0.333333	-4.925
1141	3.60960	308.333	4.0	0.695652	-3.893
1142	2.56214	354.815	3.0	0.521739	-3.790
1143	2.02164	179.219	1.0	0.461538	-2.581

1144 rows × 5 columns

3. Model Building

3.1 Model Setup

```
from pycaret.regression import *
```

```
/usr/local/lib/python3.7/dist-packages/distributed/config.py:20: YAMLLoadWarning: cal
  defaults = yaml.load(f)
```

```
model = setup(data = dataset, target = 'logS', train_size=0.8, silent=True)
```

	Description	Value
0	session_id	172
1	Target	logS
2	Original Data	(1144, 5)

3.2. Model Comparison

4	Numeric Features	4

Subsequent blocks of codes, here I will be using the *training set* (the 80% subset) for model building

```
compare_models()
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
et	Extra Trees Regressor	0.5059	0.4887	0.6909	0.8861	0.1970	0.4637
rf	Random Forest Regressor	0.5196	0.5107	0.7076	0.8809	0.1995	0.4839
lightgbm	Light Gradient Boosting Machine	0.5454	0.5422	0.7287	0.8742	0.2062	0.5043
gbr	Gradient Boosting Regressor	0.5694	0.5505	0.7360	0.8713	0.2074	0.5092
ada	AdaBoost Regressor	0.6790	0.7454	0.8560	0.8264	0.2395	0.6042
dt	Decision Tree Regressor	0.6546	0.8740	0.9310	0.7908	0.2586	0.6321
br	Bayesian Ridge	0.7771	1.0217	1.0059	0.7567	0.2873	0.8652
ridge	Ridge Regression	0.7767	1.0219	1.0060	0.7566	0.2876	0.8646
lar	Least Angle Regression	0.7766	1.0220	1.0061	0.7566	0.2876	0.8645
lr	Linear Regression	0.7766	1.0220	1.0061	0.7566	0.2876	0.8645
huber	Huber Regressor	0.7740	1.0265	1.0083	0.7554	0.2864	0.8157
en	Elastic Net	0.8499	1.2214	1.0999	0.7140	0.2895	0.9379
lasso	Lasso Regression	0.9013	1.3810	1.1694	0.6775	0.2993	1.0137
omp	Orthogonal Matching Pursuit	0.9126	1.3978	1.1742	0.6641	0.3559	1.2094
knn	K Neighbors Regressor	0.9182	1.6377	1.2698	0.6196	0.3190	0.8813
par	Passive Aggressive Regressor	1.2775	2.6631	1.5614	0.3524	0.3925	0.9305
llar	Lasso Least Angle Regression	1.6487	4.3733	2.0796	-0.0124	0.5223	2.2574
dummy	Dummy Regressor	1.6487	4.3733	2.0796	-0.0124	0.5223	2.2574

```
ExtraTreesRegressor(bootstrap=False, ccp_alpha=0.0, criterion='mse',
                    max_depth=None, max_features='auto', max_leaf_nodes=None,
                    max_samples=None, min_impurity_decrease=0.0,
                    min_impurity_split=None, min_samples_leaf=1,
                    min_samples_split=2, min_weight_fraction_leaf=0.0,
                    n_estimators=100, n_jobs=-1, oob_score=False,
                    random_state=172, verbose=0, warm_start=False)
```

32	PCA Method	None

3.3. Model Creation

```
et = create_model('et')
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.4820	0.4334	0.6583	0.8793	0.1748	0.2359
1	0.4798	0.3620	0.6017	0.8977	0.2003	0.5676
2	0.5642	0.5649	0.7516	0.8580	0.2154	0.3474
3	0.5531	0.6503	0.8064	0.8849	0.2274	0.8153
4	0.5921	0.6924	0.8321	0.8849	0.2241	0.4793
5	0.4324	0.3232	0.5685	0.9184	0.1445	0.2563
6	0.4154	0.2832	0.5321	0.9105	0.1556	0.2966
7	0.6056	0.6470	0.8044	0.8235	0.2442	0.4453
8	0.4950	0.5781	0.7604	0.8749	0.2213	1.0027
9	0.4393	0.3520	0.5933	0.9293	0.1621	0.1911
Mean	0.5059	0.4887	0.6909	0.8861	0.1970	0.4637
SD	0.0651	0.1463	0.1065	0.0292	0.0332	0.2522

3.4. Model Tuning

The learning parameters are subjected to optimization at this phase. Here, I will use 50 iterations for the optimization process and the fitness function is the Mean Absolute Error (MAE) which is the performance metric used to judge at which learning parameter setting are optimal

```
tuned_et = tune_model(et, n_iter=50, optimize='mae')
```

	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	0.5116	0.4227	0.6501	0.8823	0.1871	0.3067
1	0.5726	0.4960	0.7043	0.8599	0.2116	0.9975
2	0.5941	0.5592	0.7478	0.8595	0.2170	0.3657

```
print(tuned_et)
```

```
ExtraTreesRegressor(bootstrap=True, ccp_alpha=0.0, criterion='mse', max_depth=9,
                    max_features=1.0, max_leaf_nodes=None, max_samples=None,
                    min_impurity_decrease=0.002, min_impurity_split=None,
                    min_samples_leaf=2, min_samples_split=7,
                    min_weight_fraction_leaf=0.0, n_estimators=240, n_jobs=-1,
                    oob_score=False, random_state=172, verbose=0,
                    warm_start=False)
```

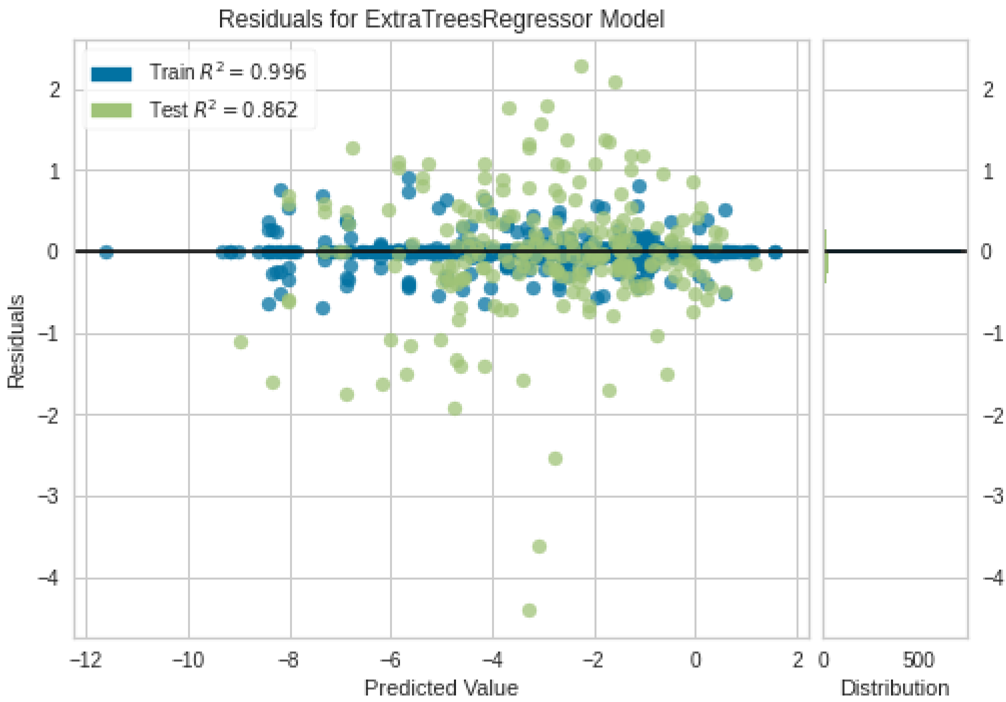
4. Model Analysis

4.1. Plot Models

In this tutorial, I will perform regression.

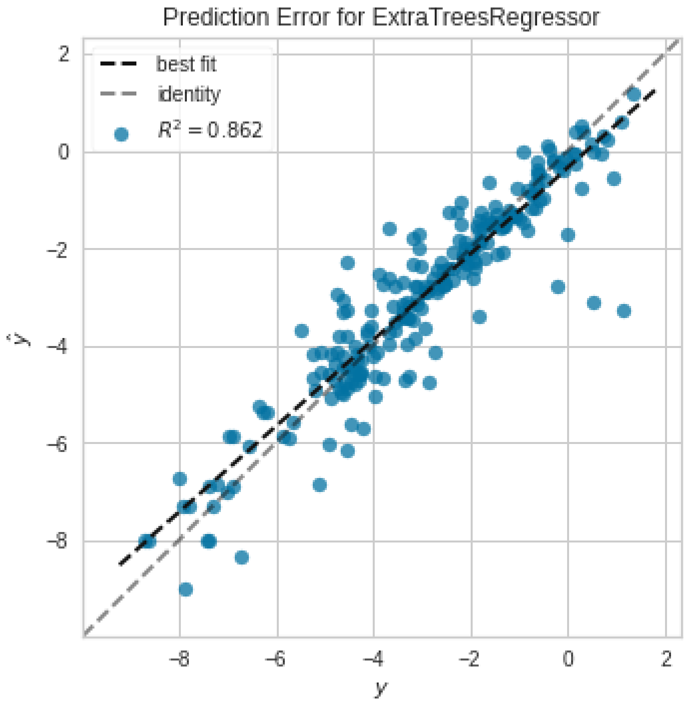
Residuals plot

```
plot_model(et, 'residuals')
```



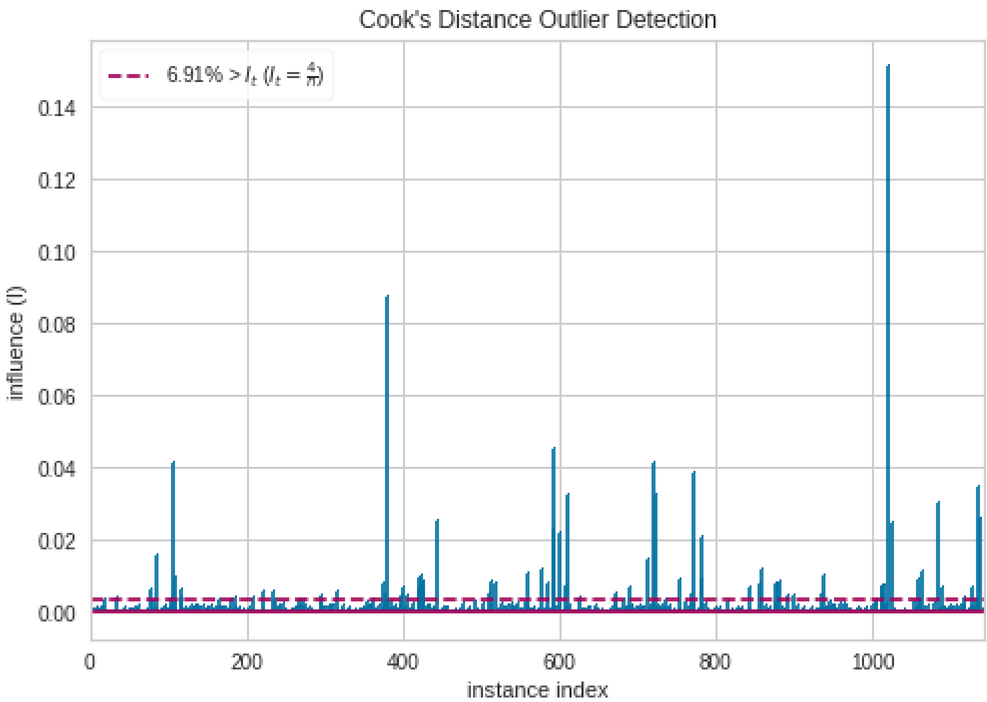
Prediction Error Plot

```
plot_model(et, 'error')
```



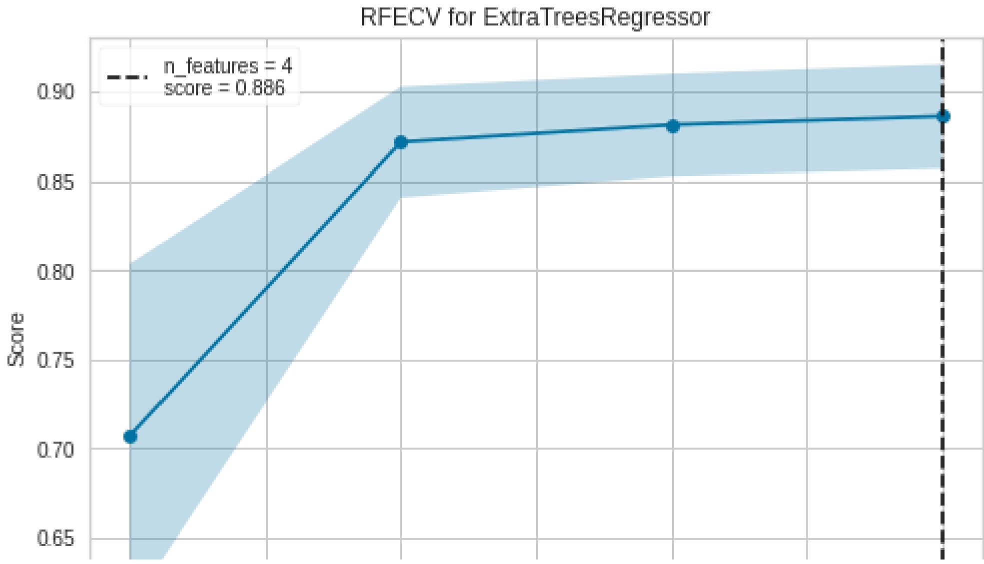
Cooks Distance Plot

```
plot_model(et, 'cooks')
```



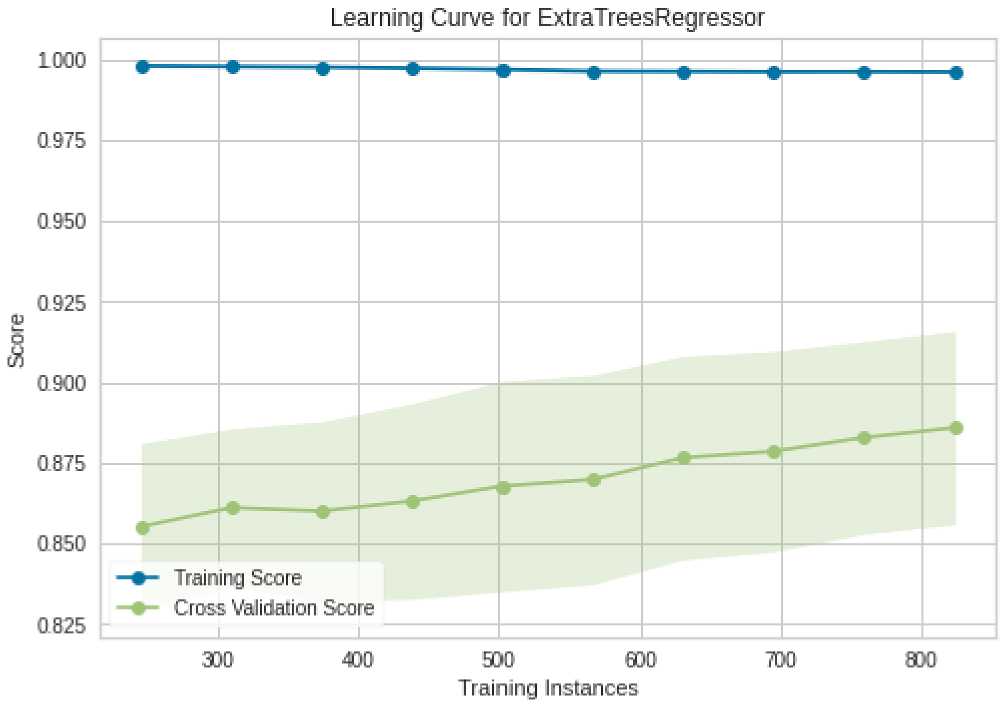
Recursive Feature Selection

```
plot_model(et, 'rfe')
```



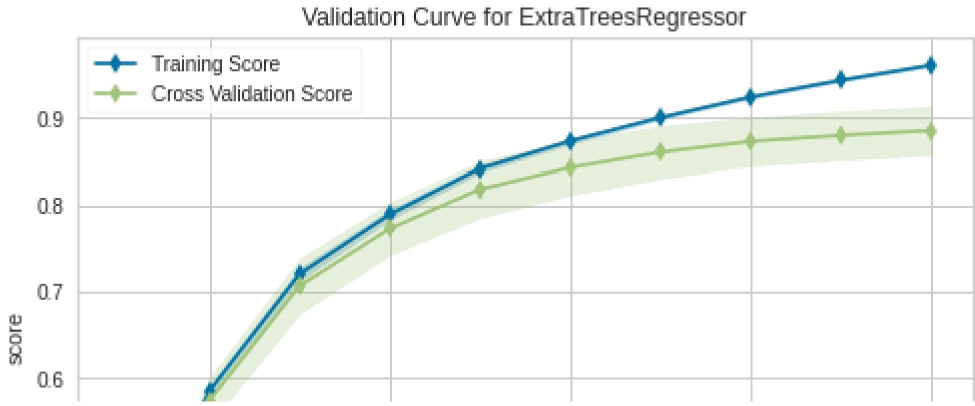
Learning Curve


```
plot_model(et, 'learning')
```



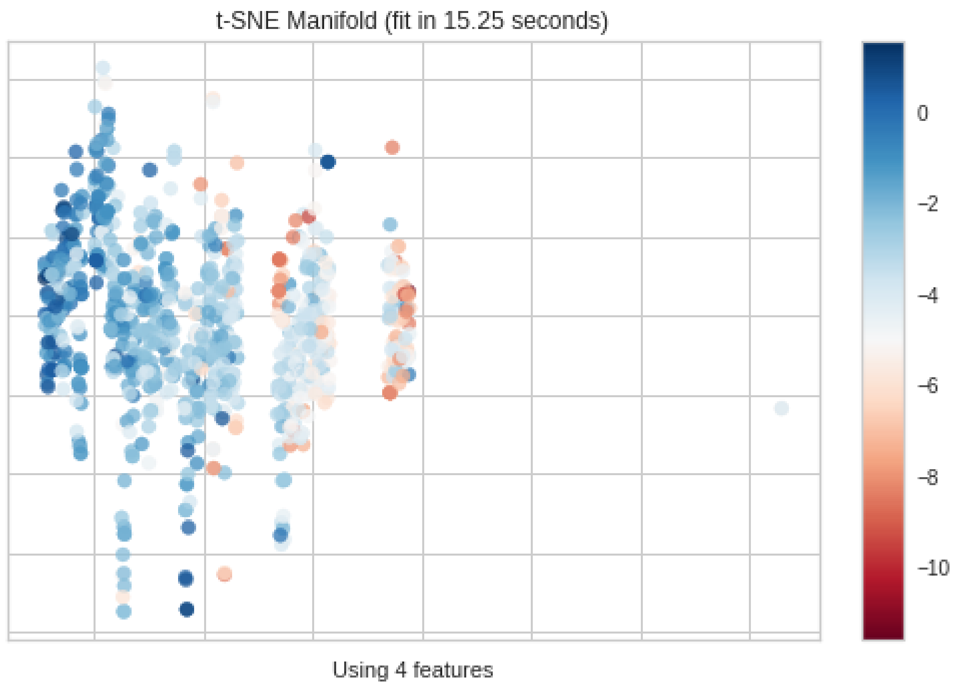
Validation Curve

```
plot_model(et, 'vc')
```



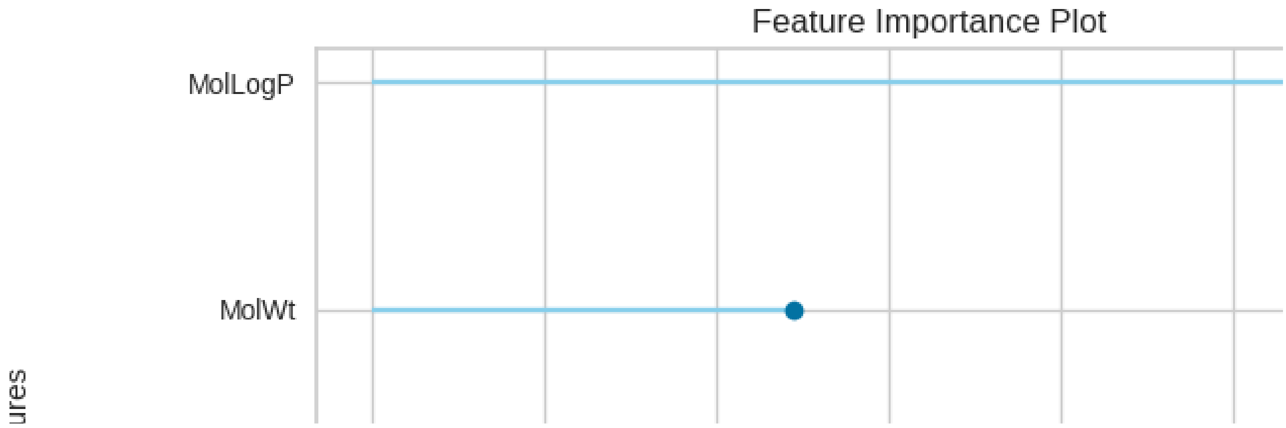
Manifold Learning

```
plot_model(et, 'manifold')
```



Feature Importance

```
plot_model(et, 'feature')
```



Model Hyperparameter

The hyperparameter of the learning model is displayed using the parameter argument in inside the plot_model() function.

```
plot_model(et, 'parameter')
```

	Parameters
bootstrap	False
ccp_alpha	0.0
criterion	mse
max_depth	None
max_features	auto
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.0
min_impurity_split	None
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
n_estimators	100
n_jobs	-1
oob_score	False
random_state	172
verbose	0
warm_start	False

Here, the hyperparameter of the tuned model is displayed below.

```
plot_model(tuned_et, 'parameter')
```

	Parameters
bootstrap	True
ccp_alpha	0.0
criterion	mse
max_depth	9
max_features	1.0
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.002
min_impurity_split	None
min_samples_leaf	2
min_samples_split	7
min_weight_fraction_leaf	0.0
n_estimators	240
n_jobs	-1
oob_score	False
random_state	172
verbose	0
warm_start	False

Show all plots

The evaulate_model() diplays all available plots here.

```
evaluate_model(tuned_et)
```

Plot Type: Hyperparameters | Residuals | Prediction Error | Cooks Distar
Feature Selection | Learning Curve | Manifold Learning | Validation Cu
Feature Importance | Feature Importance… | Decision Tree | Interactive Res

	Parameters
bootstrap	True
ccp_alpha	0.0
criterion	mse
max_depth	9
max_features	1.0
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.002
min_impurity_split	None
min_samples_leaf	2
warm_start	False

4.2. Model Interpretain

The interpret_model() function of Pycaret leverages the use of the SHAP library to produce stunning plots for depicting the *SHapley additive exPlanations (SHAP)* values that was originally proposed by Lundberg and Lee in 2016. In a nutshell, SHAP plots adds interpretability to constructed models so that the contribution of each features to the prediction can be elucidated.

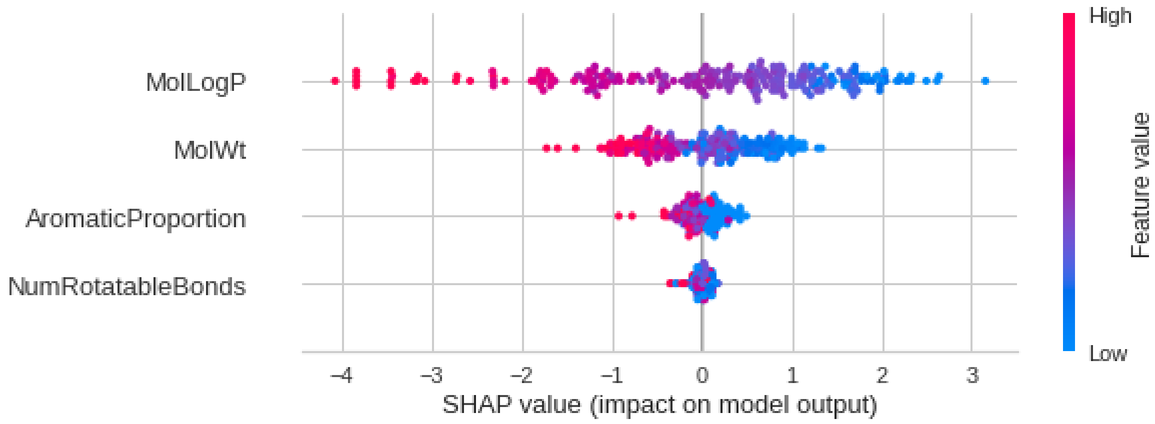
Summary plot

```
! pip install shap
```

```
Collecting shap
  Downloading shap-0.40.0-cp37-cp37m-manylinux2010_x86_64.whl (564 kB)
     |████████████████████████████████| 564 kB 5.0 MB/s
Requirement already satisfied: tqdm>4.25.0 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: cloudpickle in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/dist-packages
Collecting slicer==0.0.7
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: packaging>20.9 in /usr/local/lib/python3.7/dist-packag
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (fror
Requirement already satisfied: numba in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/c
Requirement already satisfied: llvmlite<0.35,>=0.34.0.dev0 in /usr/local/lib/python3
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-packages (
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dis
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages
```

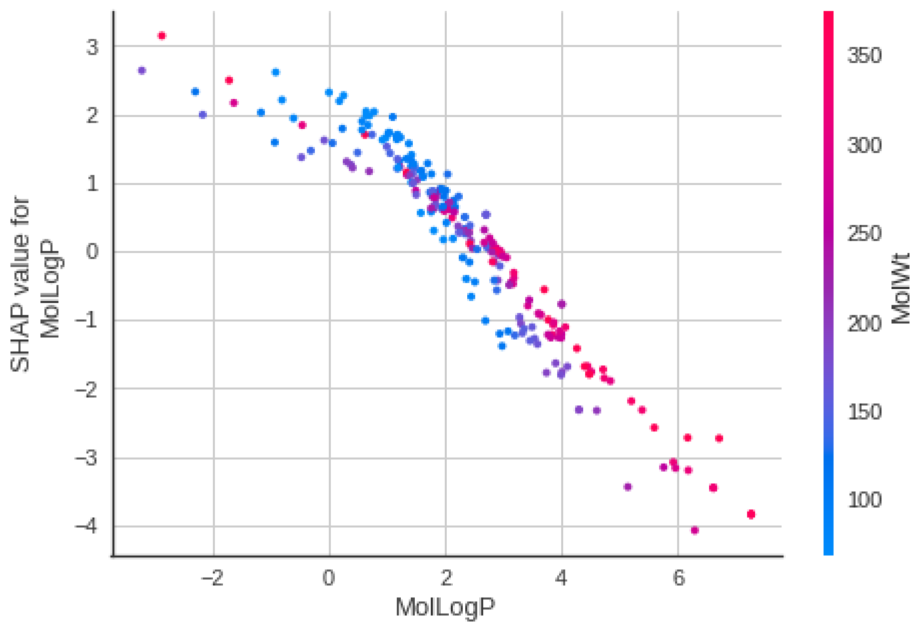
```
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7/dist-
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-package:
Installing collected packages: slicer, shap
Successfully installed shap-0.40.0 slicer-0.0.7
```

```
interpret_model(et)
```



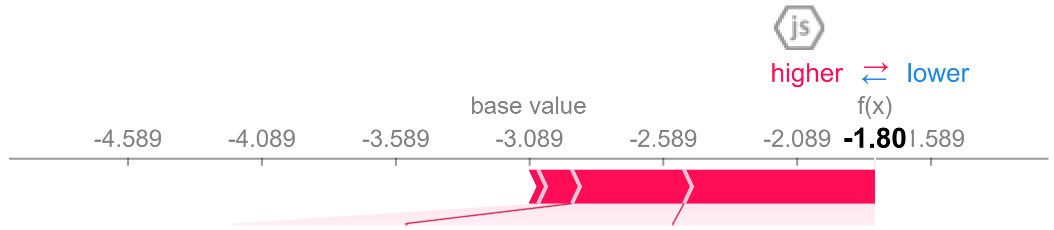
Correlation Plot

```
interpret_model(et, plot='correlation')
```



Reason Plot at Observation Level

```
interpret_model(et, plot='reason', observation=10)
```



6.6. External Testing

We will now apply the trained model (build with 80% subset) to evaulate on the so-called "**hold-out**" testing set (the 20% subset) that serves as the unseen data.

```
prediction_holdout=predict_model(et)
```

	Model	MAE	MSE	RMSE	R2	RMSLE	MAPE
0	Extra Trees Regressor	0.5208	0.6193	0.787	0.8618	0.2168	1.1997

```
prediction_holdout.head()
```

	MolLogP	MolWt	NumRotatableBonds	AromaticProportion	logS	Label
0	1.85272	209.292999	5.0	0.400000	-3.028	-2.38132
1	1.75940	90.191002	2.0	0.000000	-1.340	-2.08080
2	1.33860	588.562012	5.0	0.285714	-3.571	-3.51642
3	6.62060	326.437012	1.0	0.705882	-7.320	-7.32000
4	1.82980	169.992996	0.0	0.000000	-2.090	-1.97185