



GeekOut / CoWork / ScaleUp

Introduction to TPOT

(Automated Machine Learning in Python)



25 Mei 2019

20 Ramadhan 1440

Radical Rakhman Wahid

[linkedin.com/in/rakhid16/](https://www.linkedin.com/in/rakhid16/)
github.com/rakhid16/



Juli 2017 - Sekarang



Maret 2018 - Sekarang



Oktober 2018 - Sekarang

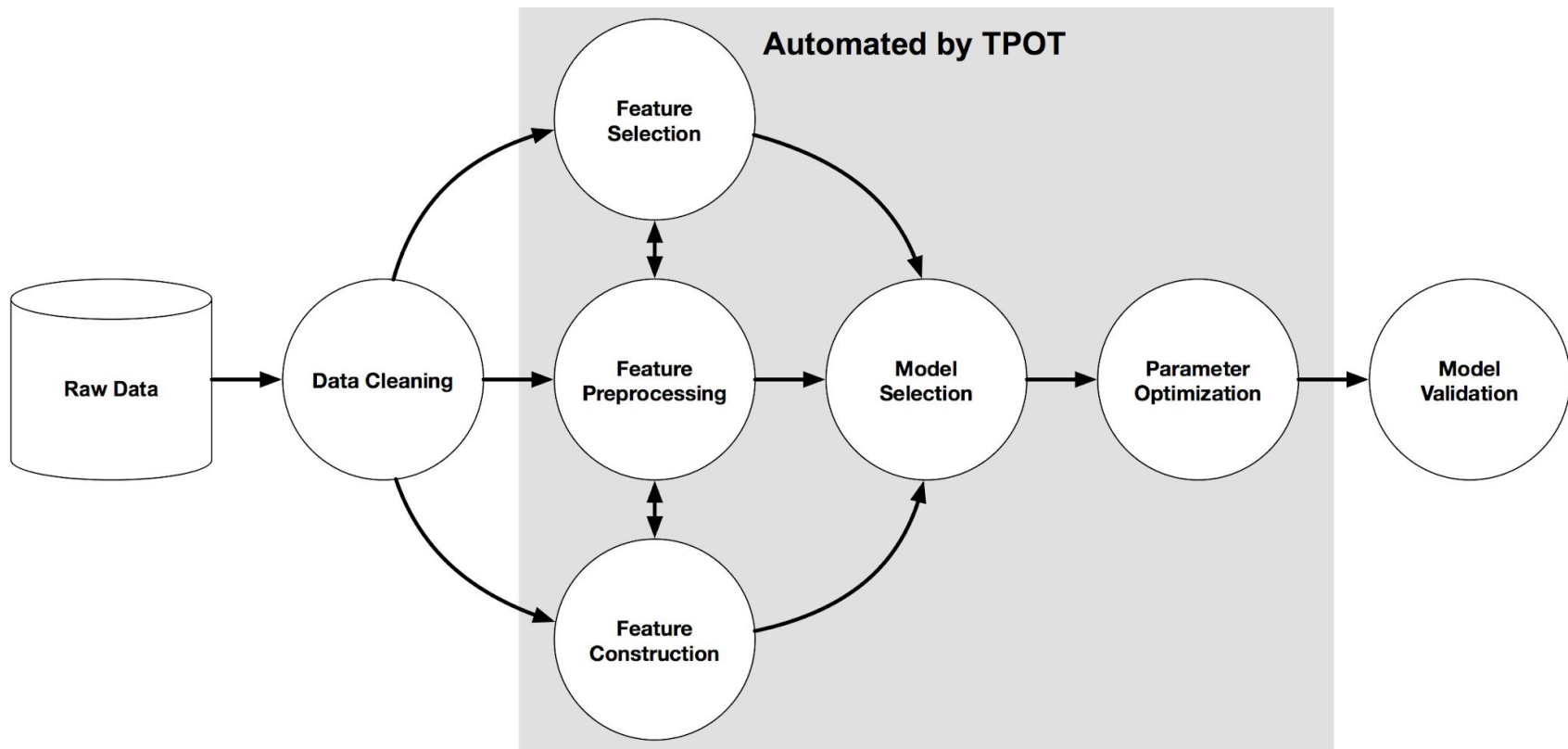
Yang hendak dibahas :

1. *TPOT Overview*
2. *Machine Learning Pipeline*
3. Cara kerja TPOT
4. *Genetic Algorithm 101*
5. Instalasi TPOT
6. Cara menggunakan TPOT (menentukan *classifier* dan *regressor* terbaik)

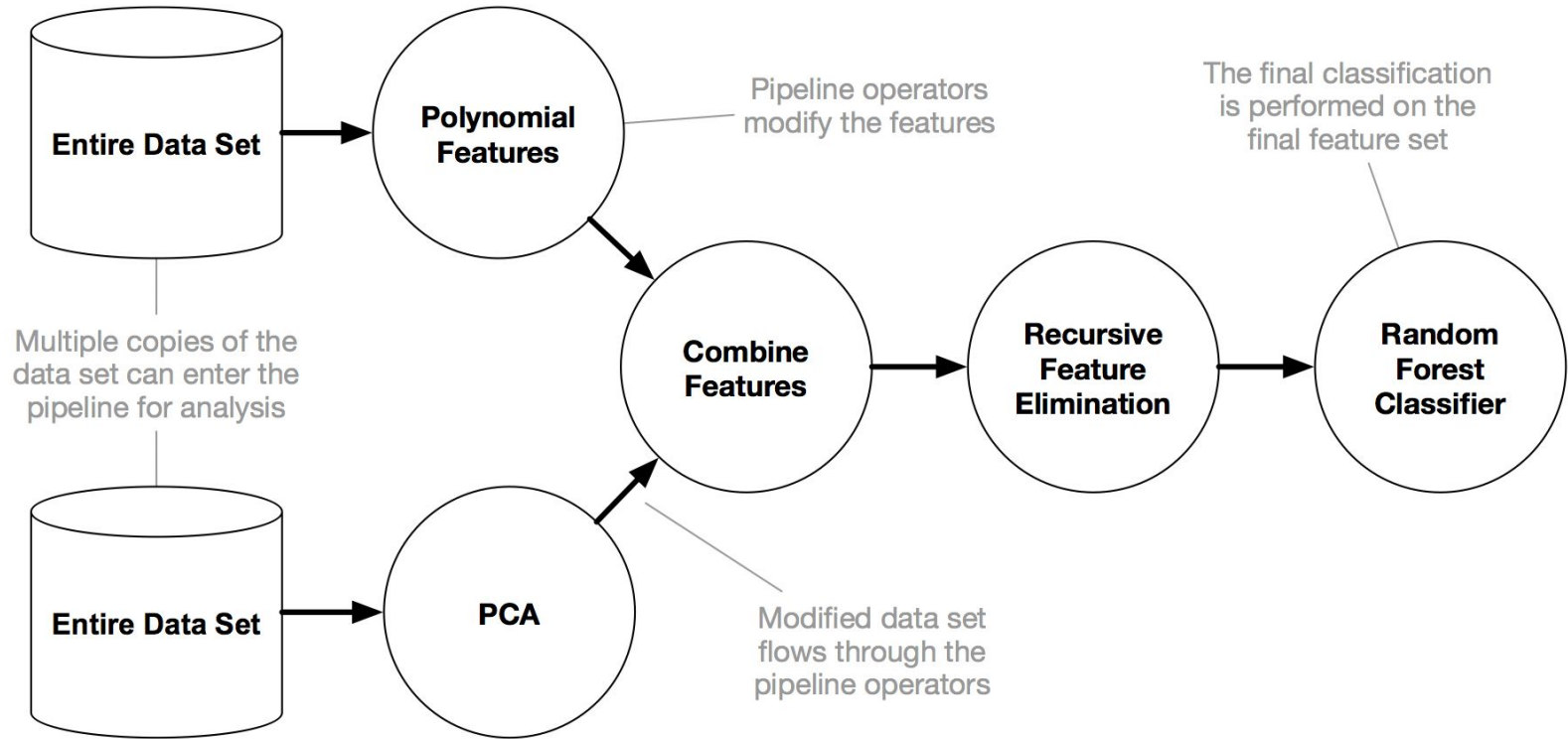
TPOT Overview

- Digunakan untuk mengoptimalkan/menentukan algoritma pembelajaran mesin secara otomatis
- Memanfaatkan algoritma genetika(metaheuristik)
- Bekerja di atas library scikit-learn
- Dilatar belakangi oleh
“Oke *dataset*’nya udah bersih algoritma pembelajaran mesin mana nih yang cocok?”
- Sangat mudah digunakan(*high level*)

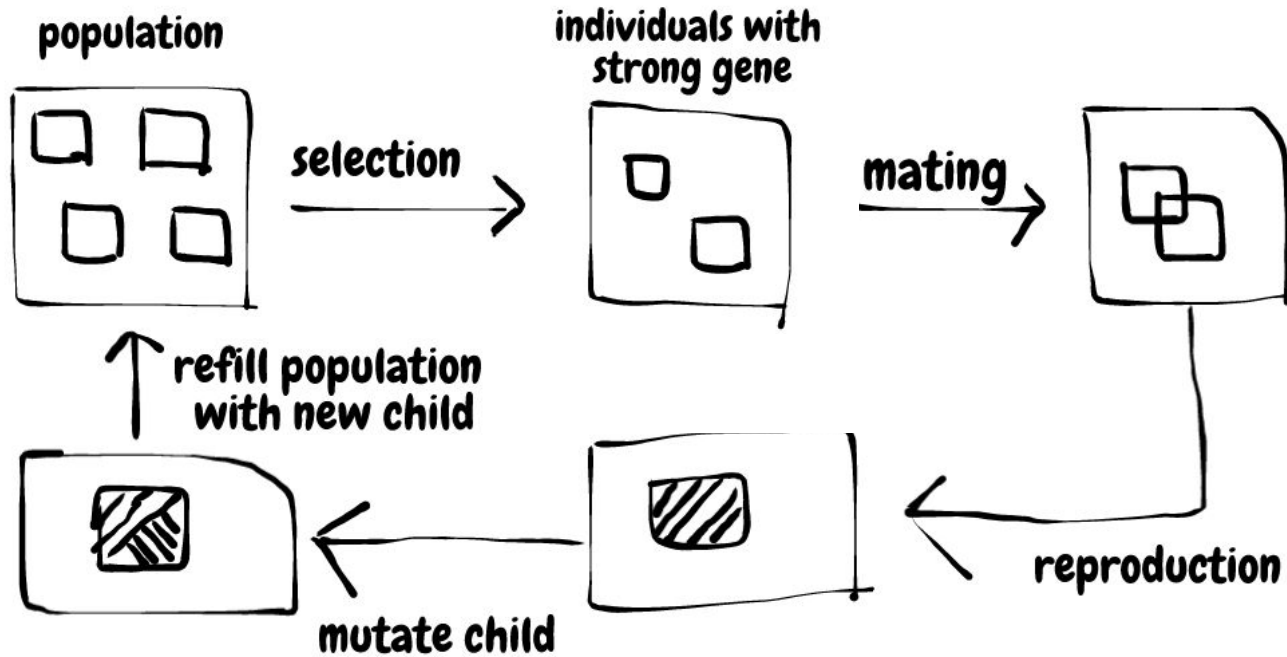
Machine Learning Pipeline



Cara kerja TPOT



Genetic Algorithm 101



Instalasi TPOT

Install python (windows)

Install pip (windows, linux, mac)

pip install tpot atau **pip3 install tpot**

Cara menggunakan TPOT untuk menentukan *classifier* terbaik

```
from tpot import TPOTClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import pandas as pd
import numpy as np

data = pd.read_csv('iris.csv')

column = ['class']
for feature in column:
    if feature in data.columns.values:
        data[feature] = LabelEncoder().fit_transform(data[feature])

X_train, X_test, y_train, y_test = train_test_split(data, data['class'].ravel(), train_size=2/3,
test_size=1/3)

tpot = TPOTClassifier(generations=5, population_size=50, verbosity=2)
tpot.fit(X_train, y_train)

print(tpot.score(X_test, y_test))

tpot.export("classifier.py")
```

Hasil ekspor(classifier.py)

```
import numpy as np
import pandas as pd
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.model_selection import train_test_split

# NOTE: Make sure that the class is labeled 'target' in the data file
tpot_data = pd.read_csv('PATH/TO/DATA/FILE', sep='COLUMN_SEPARATOR', dtype=np.float64)
features = tpot_data.drop('target', axis=1).values
training_features, testing_features, training_target, testing_target = \
    train_test_split(features, tpot_data['target'].values, random_state=None)

# Average CV score on the training set was:1.0
exported_pipeline = ExtraTreesClassifier(bootstrap=False, criterion="gini", max_features=0.1,
min_samples_leaf=12, min_samples_split=19, n_estimators=100)

exported_pipeline.fit(training_features, training_target)
results = exported_pipeline.predict(testing_features)
```

Cara menggunakan TPOT untuk menentukan *regressor* terbaik

```
from tpot import TPOTRegressor
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np

data = pd.read_csv('yacht_hydrodynamics.csv')

X_train, X_test, y_train, y_test = train_test_split(data, data['residuary'].ravel(), train_size=0.7,
test_size=0.3)

tpot = TPOTRegressor(generations=5, population_size=50, verbosity=2)
tpot.fit(X_train, y_train)

print(tpot.score(X_test, y_test))

tpot.export("regressor.py")
```

Hasil ekspor(regressor.py)

```
import numpy as np
import pandas as pd
from sklearn.cluster import FeatureAgglomeration
from sklearn.feature_selection import VarianceThreshold
from sklearn.linear_model import LassoLarsCV
from sklearn.model_selection import train_test_split
from sklearn.pipeline import make_pipeline

# NOTE: Make sure that the class is labeled 'target' in the data file
tpot_data = pd.read_csv('PATH/TO/DATA/FILE', sep='COLUMN_SEPARATOR', dtype=np.float64)
features = tpot_data.drop('target', axis=1).values
training_features, testing_features, training_target, testing_target = \
    train_test_split(features, tpot_data['target'].values, random_state=None)

# Average CV score on the training set was:-3.5543171952759125e-29
exported_pipeline = make_pipeline(
    VarianceThreshold(threshold=0.05),
    FeatureAgglomeration(affinity="manhattan", linkage="average"),
    LassoLarsCV(normalize=True)
)

exported_pipeline.fit(training_features, training_target)
results = exported_pipeline.predict(testing_features)
```

Terima kasih!



materi, *notebook*, dan *slides* presentasi :
github.com/Rakhid16/Python-TPOT