



HYPEROPT

MENGENAL HYPEROPT

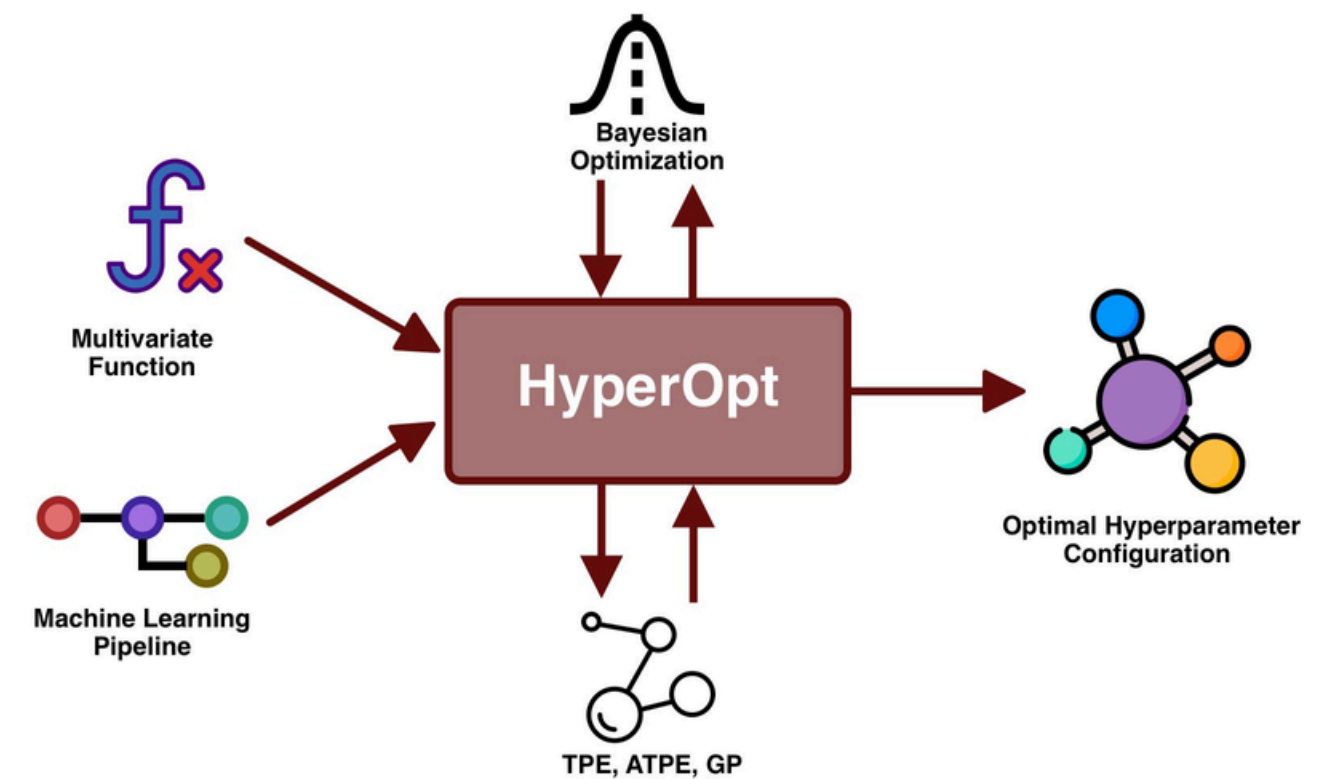
Algoritma Pintar untuk Optimasi Hyperparameter

Pengantar

Hyperopt dikembangkan oleh James Bergstra dkk pada tahun 2011 – 2013.

Dalam dunia machine learning dan deep learning, performa model sangat dipengaruhi oleh hyperparameter.

Hyperopt memungkinkan pencarian hyperparameter secara otomatis dengan berbagai metode optimasi yang lebih efisien dibandingkan pencarian manual atau grid search tradisional.

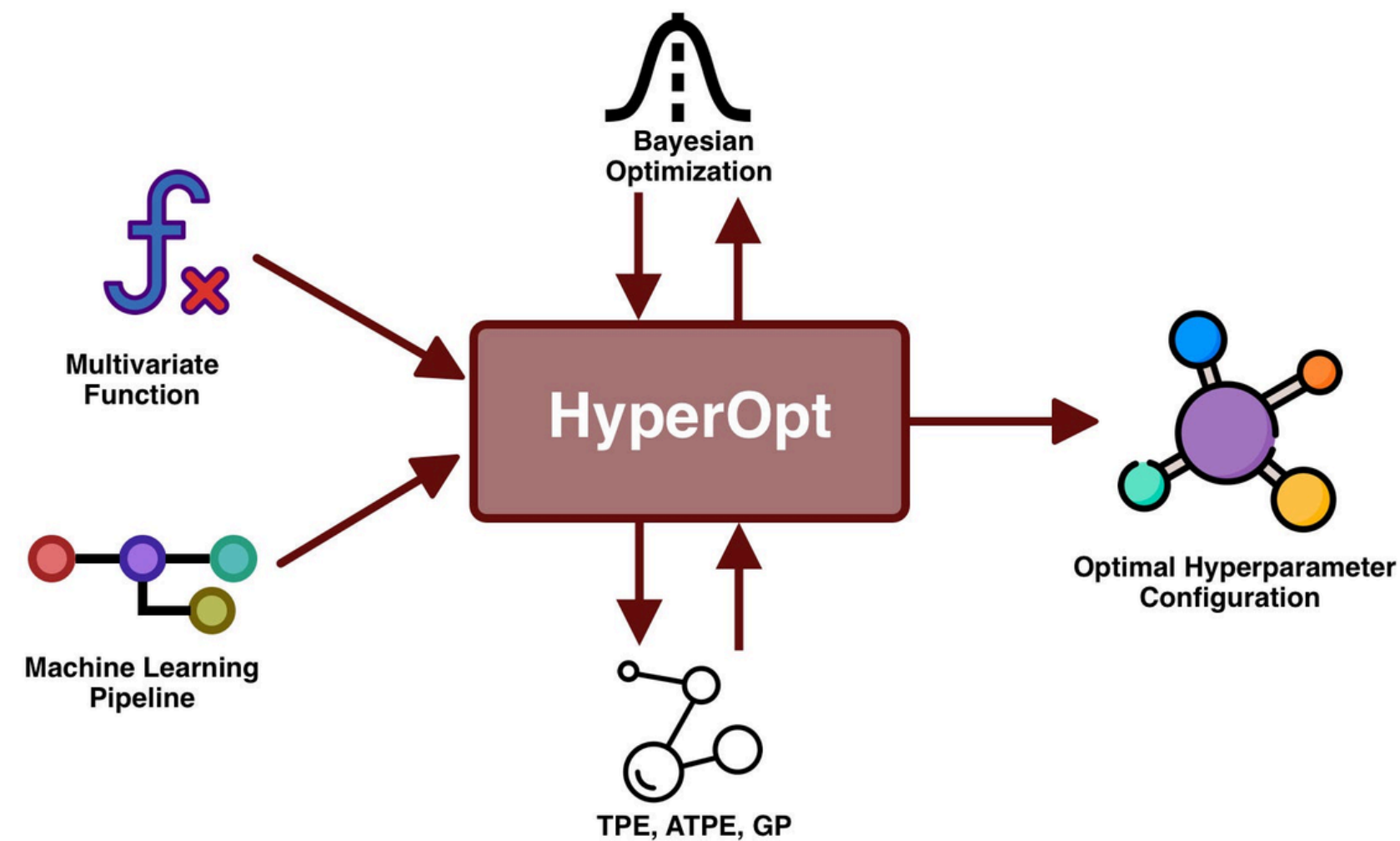


Apa itu Hyperopt ?

- Library Python untuk optimasi hyperparameter

Dapat digunakan untuk:

- ML klasik (Scikit-learn, XGBoost, LightGBM)
- Deep Learning (TensorFlow, Keras, PyTorch)
- Optimasi fungsi umum



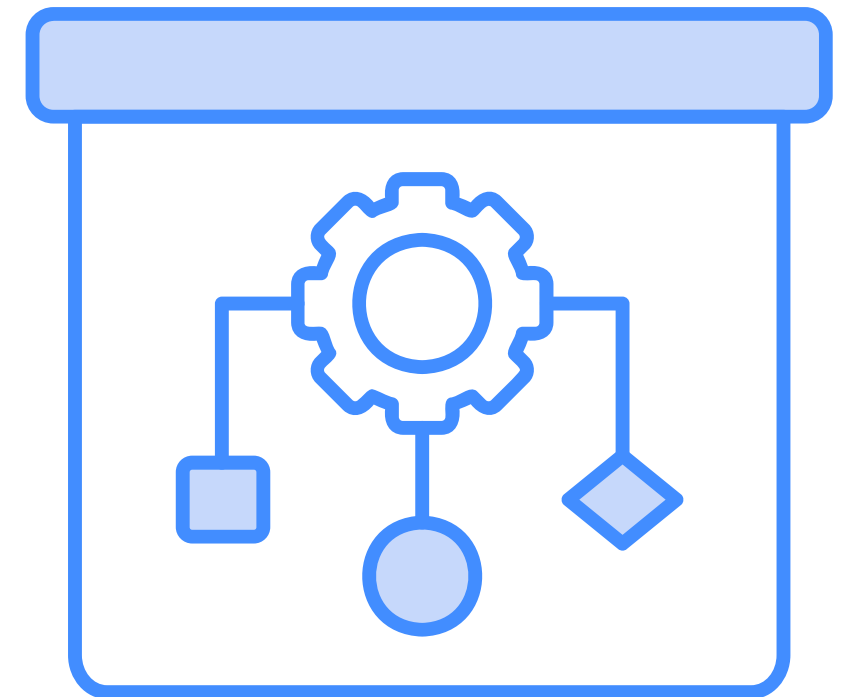
Komponen Utama Hyperopt

- `fmin()` → Fungsi utama untuk menjalankan optimasi.
- `Trials` → Objek untuk menyimpan informasi setiap percobaan
- `hp` → Modul untuk mendefinisikan ruang pencarian hyperparameter
- Algoritma optimasi (`tpe.suggest`, `rand.suggest`) → Metode pencarian hyperparameter



Optimization Algorithm Hyperopt

- Random Search → memilih kombinasi secara acak dari ruang pencarian.
- Tree of Parzen Estimators (TPE) → metode berbasis probabilistik yang membangun distribusi dari hasil sebelumnya dan menggunakan distribusi tersebut untuk memilih kandidat hyperparameter berikutnya
- Adaptive TPE/Annealing → variasi TPE dengan adaptasi suhu (simulated annealing)



Cara Kerja Hyperopt

1. Menentukan ruang pencarian hyperparameter (hp.choice, hp.uniform, dll)

```
space = {
    'max_depth': hp.choice('max_depth', range(3, 10)),
    'learning_rate': hp.uniform('learning_rate', 0.01, 0.3),
    'n_estimators': hp.quniform('n_estimators', 50, 500, 10),
}
```

2. Mendefinisikan fungsi objective

```
def objective(params):
    model = xgb.XGBClassifier(
        max_depth=int(params['max_depth']),
        learning_rate=params['learning_rate'],
        n_estimators=int(params['n_estimators']),
        eval_metric='logloss'
    )
    score = cross_val_score(model, X, y, cv=5, scoring='accuracy').mean()
    return {'loss': -score, 'status': STATUS_OK}
```

3. Menjalankan algoritma optimasi

```
trials = Trials()
best_params = fmin(
    fn=objective,
    space=space,
    algo=tpe.suggest,
    max_evals=50,
    trials=trials
)
```

4. Trials akan menyimpan nilai berdasarkan fungsi objective

```
print("Best parameters:", best_params)
```

✓ 0.0s

Best parameters: {'learning_rate': np.float64(0.26732402664794525),

Kelebihan & Kekurangan Hyperopt

Kelebihan Hyperopt

- Mendukung optimasi ruang pencarian yang kompleks.
- Lebih efisien daripada grid search dan random search biasa.
- Dapat digunakan pada berbagai jenis model (Scikit-learn, XGBoost, PyTorch, TensorFlow).

Kekurangan Hyperopt

- Dokumentasi resmi cukup ringkas, sehingga butuh eksplorasi tambahan.
- Tidak sefleksibel beberapa library lain (misalnya Optuna dalam hal visualisasi).
- Proses tuning tetap bisa memakan waktu lama pada model besar.

Kesimpulan

Hyperopt adalah library powerful untuk melakukan optimasi hyperparameter

Dengan mendefinisikan ruang pencarian, fungsi objective, dan algoritma optimasi, Hyperopt dapat secara otomatis menemukan kombinasi hyperparameter terbaik

Dibandingkan pencarian manual atau grid search, Hyperopt jauh lebih efisien dan fleksibel





**THANK YOU
FOR YOUR
ATTENTION !**