



FAKULTAS  
**ILMU  
KOMPUTER**

CSCE604135 • Perolehan Informasi  
Semester Gasal 2023/2024  
Fakultas Ilmu Komputer, Universitas Indonesia

**Tugas Pemrograman 3: Learning-to-Rank**  
**Tenggat Waktu: 23:55, 20 November 2023, 23.55 WIB**

**Ketentuan:**

1. Tugas Pemrograman 3 ini terdiri dari 1 buah tutorial LETOR.
2. Dataset dokumen dan qrels dapat kalian akses di [sini](#).
3. Buatlah program sesuai dengan petunjuk pengerjaan tugas yang disediakan.
4. Seluruh program (file .py) yang telah dibuat dikumpulkan dalam satu folder dan dikonversi ke dalam format .zip dengan format penamaan TugasX\_NPM.zip. Contoh: Tugas3\_1906262623.zip
5. Tugas ini dirancang sebagai tugas mandiri. Plagiarisme tidak diperkenankan dalam bentuk apapun. Adapun kolaborasi berupa diskusi (tanpa menyalin maupun mengambil jawaban orang lain) dan literasi masih diperbolehkan dengan mencantumkan kolaborator dan sumber.
6. Anda boleh berkonsultasi terkait tugas ini dengan 2 asisten dosen berikut. Asisten dosen diperbolehkan membantu anda dengan memberikan petunjuk.
  - a. Gibran Brahmana  
Email: [gibranbrahmanta@gmail.com](mailto:gibranbrahmanta@gmail.com)  
Whatsapp: +6281219578792  
Line: gibranbrahmanta
  - b. Hendrico Kristiawan  
Email: [hendricokristiawan@gmail.com](mailto:hendricokristiawan@gmail.com)  
Whatsapp: 089655978450  
Line: hendrico\_k

# Learning To Rank: LambdaMART

## Pengertian

Learning-to-rank atau yang umum disebut letor, merupakan salah satu task yang dapat diselesaikan menggunakan machine learning. Berikut adalah perbedaan antara beberapa task yang umum ditemukan:

- Klasifikasi

Klasifikasi adalah tugas di mana model machine learning mencoba memprediksi kategori atau label dari suatu instance berdasarkan inputnya. Misalnya, memprediksi apakah email masuk ke dalam kategori spam atau bukan. Output dari model klasifikasi berupa label atau kategori yang bersifat diskrit.

- Regresi

Regresi melibatkan prediksi nilai kontinu berdasarkan input. Berbeda dengan klasifikasi yang memprediksi kategori, regresi bertujuan untuk memprediksi angka atau nilai yang kontinu. Misalnya, memprediksi harga rumah berdasarkan fitur-fiturnya.

- Letor

Sedangkan permasalahan letor, dapat menjadi permasalahan regresi (pointwise method) maupun klasifikasi (pair wise method). Namun inti dari masalah ini adalah model harus mempelajari cara memberikan peringkat (ranking) pada sejumlah item berdasarkan preferensi atau relevansinya terhadap suatu kriteria. Ini umumnya digunakan dalam masalah seperti penyaringan informasi dan sistem rekomendasi.

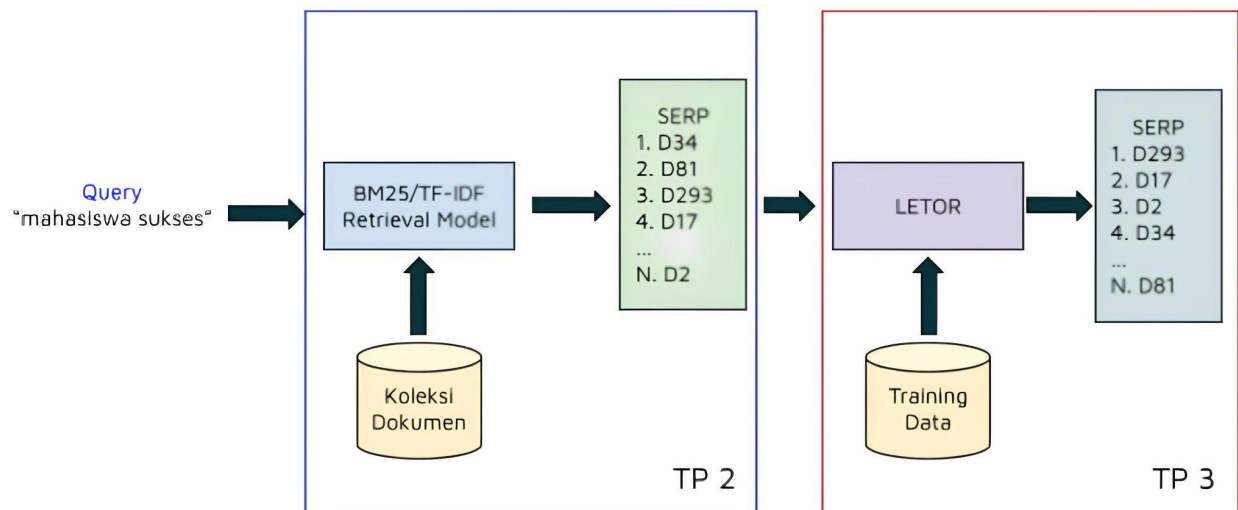
## LambdaMART

LambdaMART bersama dengan RankNet dan LambdaRank merupakan algoritma Learning to Rank yang ditemukan oleh Chris Burges dan rekannya di Microsoft Research. Ketiganya mentransformasikan ranking ke bentuk pairwise classification atau regression problem. LambdaMART merupakan perpaduan dari LambdaRank dan MART (Multiple Additive Regression Trees). Terdapat dua implementasi populer LambdaMART:

- RankLib yang merupakan bagian dari Lemur Project.
- LightGBM yang disediakan oleh Microsoft.

Pada tutorial yang diberikan, akan menggunakan LightGBM sebagai contoh implementasi. Cara Kerja LambdaMART menggunakan LightGBM dapat dilihat pada link [ini](#)

## Petunjuk Pengerjaan Tugas



Sebelumnya di Tugas Pemrograman 2, Anda diminta mengimplementasikan ranked retrieval model dengan pembobotan menggunakan sparse retrieval model, yaitu TF-IDF dan BM25. Output yang dihasilkan dari Tugas Pemrograman 2 adalah top-100 dokumen terbaik dari suatu query.

Pada Tugas Pemrograman 3 ini, anda akan menggunakan kembali program dari Tugas Pemrograman 2 sebagai basis. Jadi, model yang dikembangkan pada Tugas Pemrograman 2 akan mengembalikan top-100 dokumen yang harapannya mengandung dokumen-dokumen yang relevan terhadap query (walaupun mungkin saja dokumen-dokumen yang relevan tersebut berada pada posisi ranking bawah). Tugas Pemrograman 3 kemudian menerima input top-100 dokumen tersebut beserta query awal, dan melakukan re-ranking untuk memperbaiki kualitas SERP yang dihasilkan sebelumnya. Anda diminta untuk mempelajari tutorial Learning-to-rank yang telah disediakan dan mengimplementasikannya ke kode milik anda sendiri.

Berikut cara mengimplementasikan LETOR:

1. Persiapkan data yang akan dilakukan re-ranking
2. Membuat LSI/LSA Model
3. Train LightGBM LambdaMART Model
4. Melakukan Prediksi

Untuk memudahkan Anda, berikut walkthrough singkat terkait pengerjaan TP-3:

1. Implementasi LETOR toolkit dengan mengadaptasi salah satu toolkit LETOR yang dapat diakses melalui tutorial berikut: [ristek.link/TutorialLETOR](https://ristek.link/TutorialLETOR) ([link alternatif](#)). Kemas code tersebut dalam suatu class atau fungsi dengan format .py. Kode di notebook biasanya kurang rapih dan tidak modular. Anda perlu merapihkan kode di notebook tersebut dengan menggunakan class atau fungsi.
2. Bandingkan hasil ranking dengan menggunakan LETOR dan tidak, tuliskan hasil evaluasi dalam file .txt

## Pengumpulan

Kumpulkan sebuah *file* .zip dengan format **Tugas3\_NPM.zip** (contoh: **Tugas3\_2006123456.zip**) yang berisi:

- util.py
- compression.py
- index.py
- bsbi.py
- search.py (opsional)
- experiment.py
- File py yang digunakan untuk letor
- *folder* index
- *file* laporan (txt atau pdf)
- *file* terkait pengerjaan bonus bila ada

Anda tidak perlu mengumpulkan *folder* collections.

Note:

Pada eksperiment.py di TP2, kalian melakukan inisialisasi qrels dengan membuat dictionary terlebih dahulu sebanyak max query id dan max doc id. Untuk TP3, kalian disarankan untuk menggunakan [default dict](#) agar semua value memiliki nilai awal, pada kasus ini 0 jika pasangan query dan dokumen tidak relevan. Berikut contoh kode yang memanfaatkan default dict.

```
from collections import defaultdict
```

```
def load_qrels(qrel_file = "test_qrels.txt"):
    qrels = defaultdict(lambda: defaultdict(lambda: 0))
    with open(qrel_file) as file:
        for line in file:
            parts = line.strip().split()
            qid = parts[0]
            did = int(parts[1])
            qrels[qid][did] = 1
    return qrels
```

## Bonus

Sebelumnya Anda telah melakukan implementasi re-ranking menggunakan LETOR. Terkait bonus, silahkan lakukan eksperimen yang dapat meningkatkan hasil menjadi lebih baik, seperti menambahkan fitur-fitur baru, melakukan hyperparameter tuning pada library LambdaMART, atau coba cari dan implementasikan library LETOR lain sebagai perbandingan. Tuangkan hasil evaluasi mu dalam file berformat .txt. Kemudian, lakukan perbandingan terkait hasil dari sebelum dan setelah eksperimen dan tuliskan hasil evaluasi perbandingan tersebut.

### Poin penilaian:

- letor.py 90 poin
- evaluasi.txt 10 poin
- Bonus 10 poin

### Catatan:

- Gunakan set test pada qrels saat melakukan evaluasi akhir
- Untuk metrics, scorenya cukup 2 angka di belakang koma.
- Tugas akan dinilai bukan berbasis output, melainkan berbasis ide yang dikerjakan.
- Mohon untuk menuliskan kode dengan rapi agar penilai dapat mengerti ide dan kode

yang dituliskan.

**Referensi & Kredit:**

- Intuitive explanation of Learning to Rank (and RankNet, LambdaRank and LambdaMART)
- A Practical Guide to LambdaMART in LightGbm

**Selamat mengerjakan!**