A. Teori

1. Jelaskan yang dimaksud dengan machine learning?
2. Berikan satu contoh pemanfaatan machine learning (selain yang disebutkan di slide)?
3. Pada soal nomor 3 ini, kita mengeksplorasi hubungan *cosine similarity* dengan *correlation* serta *euclidean distance* untuk data dalam ruang dimensi R^n:
   a. Berapa rentang nilai yang dapat dihasilkan dari *cosine similarity*?
   b. Jika skor *cosine similarity* antar dua data sama dengan 1, apakah kedua data tersebut identic (sama persis)? Berikanlah penjelasan atas jawaban anda.
   c. Apa hubungan antara cosine similarity dengan correlation (jika ada)? (petunjuk: perhatikan ukuran-ukuran statistic seperti rata-rata dan standar deviasi dimana cosine dan correlation keduanya bisa sama dan bisa juga berbeda)
   d. Jelaskan/rumuskan hubungan secara matematik antara cosine similarity dan Euclidean distance ketika setiap data memiliki panjang $L_2 = 1$. Panjang L_2 disebut juga sebagai norm.
   e. Jelaskan/rumuskan hubungan secara matematik antara correlation dan Euclidean distance ketika setiap data telah distandarkan yaitu telah dikurangi dengan mean dan dibagi dengan standar deviasi.
4. Proximity biasanya digunakan untuk pasangan data.
   a. Berikan dua cara untuk mendefinisikan *proximity* (kedekatan) antar data pada sekumpulan data (jumlah data pada kumpulan tersebut lebih dari dua). Misal: suatu rumus/ukuran untuk mengukur seberapa mirip sejumlah data terhadap satu sama lain.
   b. Bagaimana cara anda mendefinisikan jarak antara dua set/kumpulan data dalam ruang *Euclidian*? (Note: jarak antara dua kumpulan data, bukan jarak antar dua data)
   c. Bagaimana cara anda mendefinisikan proximity/kedekatan antara dua kumpulan data? (Note: kedekatan antara dua kumpulan data, bukan kedekatan antar dua data. Hindari penggunakan asumsi tambahan tentang data set).
5. Diberikan matriks dokumen, dimana tf_{ij} merupakan jumlah kemunculan kata ke-*i* dalam dokumen ke-*j,* sedangkan *m* adalah total dokumen. Anggap transformasi variabel yang didefinisikan oleh *m* adalah

$$tf'_{ij} = tf_{ij} \ log \frac{m}{df_i}$$

Dimana df_i merupakan jumlah dokumen yang mengandung kata ke-*i* (disebut juga sebagai frekuensi dokumen terhadap suatu term/kata). Transformasi tersebut dikenal sebagai transformasi *inverse document frequency (IDF)*.

a. Apa efek dari transformasi tersebut jika suatu kata muncul hanya pada satu dokumen (hanya dimiliki oleh satu dokumen)?
b. Apa efek dari transformasi tersebut jika suatu kata muncul pada setiap dokumen (terdapat pada semua dokumen)?
c. Apa efek secara keseluruhan dan apa tujuan dari transformasi tersebut?
d. Dapatkah anda menemukan data lain (non dokumen) yang dapat menggunakan perhitungan transformasi tersebut?

B. Programming

6. Pada problem ini akan dilakukan implementasi perhitungan *similarity* antar data film pada dataset *Movielens.* Download dataset *Movielens* yang telah dikirimkan melalui

email dan Telegram. Pada dataset tersebut terdapat juga beberapa fungsi yang mempermudah proses *load* data untuk bahasa pemrograman Matlab, Octave dan R serta beberapa contoh *codingan* yang dapat digunakan. Lihat file README untuk keterangan lebih lanjut.

a. Melakukan perhitungan *similarity* terhadap *movie*. Pertama, lakukan perhitungan yang mudah dengan tidak menggunakan nilai rating (eksplisit) yang diperoleh dari seluruh user dan juga *time stamp* dari rating. Namun, gunakan hanya informasi tentang apakah suatu movie sudah diberi rating atau tidak oleh seorang user.
    i. Buatlah suatu fungsi dimana inputnya adalah dua ID movie yang berbeda, sedangkan outputnya adalah **koefisien Jaccard** (yaitu jumlah user yang memberikan rating untuk kedua movie tersebut dibagi jumlah user yang setidaknya telah memberikan rating terhadap salah satu movie tersebut). Sebagai contoh, untuk movie Toy Story dan Golden Eye nilai koefisiennya adalah 0.217.
    ii. Berapakah nilai koefisien Jaccard untuk movie 'Three Colors: Red' dan movie 'Three Colors: Blue'?
    iii. Sebutkan 5 movie yang memiliki nilai koefisien Jaccard tertinggi terhadap movie 'Taxi Driver'?
    iv. Pilih salah satu movie (yang anda ketahui) dan tentukan 5 movie yang memiliki nilai koefisien Jaccard tertinggi terhadap movie tersebut. Apakah hasilnya masuk akal?

b. Perhitungan similarity dibawah ini menggunakan eksplisit rating
    I. Buatlah fungsi dimana inputnya adalah dua ID movie yang berbeda, sedangkan outputnya adalah **koefisien korelasi** dari rating yang diberikan untuk kedua movie tersebut oleh semua user yang telah memberikan rating pada kedua movie tersebut. (Perhatikan: fungsi harus dapat mengembalikan nilai 0 jika jumlah user yang memberikan rating untuk kedua movie sangatlah kecil sehingga perhitungan terhadap koefisien korelasi tidak memungkinkan).
    II. Berapakah nilai similarity (koefisien korelasi) antara movie 'Toy Story' dan 'Golden Eye'?
    III. Berapakah nilai similarity (koefisien korelasi) antara movie 'Three Colors: Red' dan 'Three Colors: Blue'?
    IV. Sebutkan 5 movie dengan nilai similarity (koefisien korelasi) tertinggi terhadap movie 'Taxi Driver'?
    V. Pilih salah satu movie (yang anda ketahui) dan tentukan 5 movie yang memiliki nilai similarity (koefisien korelasi) tertinggi terhadap movie tersebut.

c. Berdasarkan poin 6a dan 6b diatas, berikan penjelasan yang menerangkan perhitungan similarity mana yang lebih baik, dimana perhitungan similarity tersebut sesuai dengan intuisi anda tentang similarity. Mengapa anda berfikir demikian? Jelaskan.

7. Pada latihan ini, anda diharuskan mengimplementasikan proses klasifikasi menggunakan *classifier* yang sangat sederhana berbasiskan prototype, disebut sebagai *prototype-based classifier*, untuk mengklasifikasi tulisan digit menggunakan dataset MNIST, kemudian membandingkan hasilnya dengan *nearest-neighbor classifier*.

Download dataset MNIST yang telah dikirimkan melalui email dan Telegram. Pada dataset tersebut terdapat juga beberapa fungsi yang mempermudah proses *load* data untuk bahasa pemrograman Matlab, Octave dan R serta beberapa contoh *codingan* yang dapat digunakan. Lihat file README untuk keterangan lebih lanjut.

a. Load 5000 image pertama menggunakan fungsi yang telah disediakan. Gunakan fungsi yang telah disediakan untuk melakukan plot terhadap 100 data image yang anda pilih secara acak dan tunjukkan label-label dari image tersebut. Lakukan verifikasi bahwa label-label tersebut sesuai dengan masing-masing gambar digitnya. (Verfikasi ini adalah suatu keharusan untuk membuktikan bahwa data yang anda miliki sudah dalam format yang benar.)
b. Bagilah data menjadi dua bagian, 'data training' terdiri atas 2500 gambar (beserta label) dan 'data testing' terdiri atas 2500 gambar (beserta label). Untuk setiap 10 kelas (digit 0-9), buatlah prototype dari masing-masing kelas dengan cara menggunakan nilai rata-rata dari seluruh gambar pada data training untuk kelas yang sama. Sebagai contoh, lakukan pengambilan seluruh gambar dari kelas 0 (pada data training) kemudian hitung rata-rata dari gambar tersebut. Nilai rata-rata inilah yang kita sebut sebagai prototype. Lakukan hal tersebut untuk seluruh kelas dan plot hasilnya. Apakah hasil tersebut sesuai dengan yang anda inginkan?
c. Untuk setiap gambar di data testing, hitunglah jarak Euclidean-nya terhadap 10 prototype yang telah dihasilkan dari soal 7b, kemudian klasifikasikan gambar tersebut kedalam kelas yang memiliki jarak Euclidean terkecil. Oleh karena itu, jika suatu gambar di data testing memiliki jarak yang paling dekat dengan prototype '3', maka kelas dari gambar tersebut adalah kelas 3. Hitung dan tampilkan hasil klasifikasi tersebut dalam bentuk *confusion matrix*.
d. Klasifikasikan setiap data dari data testing menggunakan *nearest neighbor classifier*. Caranya: untuk setiap data testing, hitunglah jarak Euclidean-nya terhadap seluruh data training (2500 gambar), kemudian kelas prediksi dari gambar testing tersebut ditentukan oleh kelas dari gambar (di data training) yang paling dekat dengan data testing tersebut. Hitung dan tampilkan hasil klasifikasi dalam bentuk confusion matrix.
e. Hitung dan bandingkan nilai error dari kedua classifier tersebut (*prototype-based classifier* dan *nearest neighbor classifier*). Classifier mana yang memberikan hasil yang terbaik? Berdasarkan confusion matrix yang telah diperoleh, digit mana yang paling banyak salah diklasifikasikan kedalam digit lain? Mengapa demikian?