Muhammad Aizar Yazid / 1103223097

Chapter 1: The Machine Learning Landscape

1. Ringkasan Teori

Chapter pertama ini menyajikan gambaran umum tentang Machine Learning (ML), memperkenalkan konsep-konsep fundamental, terminologi, jenis-jenis sistem ML, dan tantangan utama yang dihadapi.

- **Definisi Machine Learning** adalah ilmu yang memprogram komputer agar dapat belajar dari data. Definisi yang lebih berorientasi pada rekayasa menyebutkan bahwa sebuah program komputer dikatakan belajar dari pengalaman E sehubungan dengan tugas T dan ukuran kinerja P, jika kinerjanya pada T, yang diukur dengan P, meningkat seiring dengan pengalaman E. Contoh sederhananya adalah filter spam, yang belajar untuk menandai email spam berdasarkan contoh email spam (pengalaman) untuk tugas menyaring email baru (tugas) dengan tujuan meningkatkan rasio email yang diklasifikasikan dengan benar (kinerja).
- **Jenis-Jenis Sistem Machine Learning** Sistem ML dapat diklasifikasikan berdasarkan beberapa kriteria utama:
    1. **Berdasarkan Pengawasan Manusia (Human Supervision):**
        - **Supervised Learning:** Algoritma diberi data latihan yang sudah diberi label, artinya setiap data sudah memiliki solusi atau target yang diinginkan. Dua tugas paling umum adalah **klasifikasi** (seperti filter spam) dan **regresi** (seperti memprediksi harga rumah).
        - **Unsupervised Learning:** Data latihan tidak memiliki label, dan sistem mencoba belajar tanpa "guru". Tugas-tugas umumnya meliputi **clustering** (misalnya, mengelompokkan pengunjung blog serupa) , **visualization** dan **dimensionality reduction** (menyederhanakan data tanpa kehilangan banyak informasi) , **anomaly detection** (mendeteksi transaksi tidak biasa) , dan **association rule learning** (menemukan hubungan menarik antar atribut).
        - **Semisupervised Learning:** Menangani data yang sebagian besar tidak berlabel dengan sedikit data berlabel. Contohnya adalah Google Photos yang mengelompokkan foto orang yang sama secara otomatis (bagian unsupervised) dan kemudian hanya membutuhkan satu label dari pengguna untuk menamai semua foto orang tersebut (bagian supervised).
        - **Reinforcement Learning (RL):** Sistem yang disebut *agent* belajar dengan mengamati lingkungan, memilih dan melakukan tindakan, dan mendapatkan *rewards* atau *penalties* sebagai imbalannya. Tujuannya adalah untuk belajar strategi terbaik (disebut *policy*) untuk mendapatkan *reward* terbesar dari waktu ke waktu.
    2. **Berdasarkan Kemampuan Belajar secara Inkremental:**

        - **Batch Learning:** Sistem harus dilatih menggunakan semua data yang tersedia sekaligus. Sistem ini tidak dapat belajar secara bertahap. Untuk belajar dari data baru, sistem harus dilatih ulang dari awal dengan dataset penuh.
        - **Online Learning:** Sistem dilatih secara bertahap dengan memberinya data secara sekuensial, baik satu per satu atau dalam kelompok kecil yang disebut *mini-batches*. Ini sangat berguna untuk sistem yang perlu beradaptasi dengan cepat terhadap data yang berubah.
    3. **Berdasarkan Cara Generalisasi:**
        - **Instance-Based Learning:** Sistem mempelajari contoh-contoh data dengan "menghafalnya", kemudian menggeneralisasi ke kasus-kasus baru dengan membandingkannya dengan contoh yang telah dipelajari menggunakan ukuran kesamaan (*similarity measure*).
        - **Model-Based Learning:** Sistem membangun sebuah model dari contoh-contoh data, kemudian menggunakan model tersebut untuk membuat prediksi. Proses ini melibatkan pemilihan model, melatihnya pada data latihan (yaitu, algoritma pembelajaran mencari nilai parameter model yang meminimalkan *cost function*), dan akhirnya menerapkan model untuk membuat prediksi pada kasus baru (disebut *inference*).
- **Tantangan Utama Machine Learning** Tantangan utama dalam ML dapat diringkas menjadi dua kategori: "algoritma yang buruk" dan "data yang buruk".
    - **Data yang Buruk:**
        - **Kuantitas Data Pelatihan yang Tidak Cukup:** Sebagian besar algoritma ML membutuhkan banyak data untuk bekerja dengan baik.
        - **Data Pelatihan yang Tidak Representatif:** Data pelatihan harus mewakili kasus-kasus baru yang ingin digeneralisasi. Sampel yang kecil dapat menimbulkan *sampling noise*, sedangkan metode sampling yang cacat dapat menyebabkan *sampling bias*.
        - **Data Berkualitas Buruk:** Kesalahan, *outliers*, dan *noise* dalam data pelatihan akan mempersulit sistem untuk mendeteksi pola yang mendasarinya.
        - **Fitur yang Tidak Relevan:** Sistem hanya akan mampu belajar jika data pelatihan berisi cukup fitur yang relevan dan tidak terlalu banyak fitur yang tidak relevan. Proses ini disebut *feature engineering*.
    - **Algoritma yang Buruk:**
        - **Overfitting:** Model berkinerja baik pada data pelatihan, tetapi tidak dapat menggeneralisasi dengan baik untuk data baru. Ini terjadi ketika model terlalu kompleks dibandingkan dengan jumlah dan tingkat *noise* pada data pelatihan.
        - **Underfitting:** Model terlalu sederhana untuk mempelajari struktur data yang mendasarinya.

- **Pengujian dan Validasi** Untuk mengetahui seberapa baik sebuah model akan menggeneralisasi ke kasus baru, data dibagi menjadi dua set: **training set** dan **test set**. Model dilatih menggunakan training set dan diuji menggunakan test set. Untuk memilih model terbaik dan menyetel *hyperparameter*, sebagian dari training set disisihkan sebagai **validation set** (atau *dev set*).

2. Latihan (Exercises)

Berikut adalah jawaban dari soal-soal latihan pada akhir Chapter 1.

1. Bagaimana Anda mendefinisikan Machine Learning? Machine Learning adalah ilmu (dan seni) memprogram komputer sehingga mereka dapat belajar dari data. Definisi yang lebih formal adalah bidang studi yang memberikan komputer kemampuan untuk belajar tanpa diprogram secara eksplisit.

2. Dapatkah Anda menyebutkan empat jenis masalah di mana ML bersinar? Machine Learning sangat berguna untuk:

- Masalah yang solusinya memerlukan banyak penyesuaian manual atau daftar aturan yang panjang (satu algoritma ML seringkali dapat menyederhanakan kode dan berkinerja lebih baik).
- Masalah kompleks yang tidak memiliki solusi baik dengan pendekatan tradisional (teknik ML terbaik mungkin dapat menemukan solusi).
- Lingkungan yang berfluktuasi (sistem ML dapat beradaptasi dengan data baru).
- Mendapatkan wawasan tentang masalah kompleks dan data dalam jumlah besar (*data mining*).

3. Apa itu *labeled training set*? *Labeled training set* adalah set data pelatihan di mana setiap instance data disertai dengan solusi yang diinginkan, yang disebut "label". Misalnya, dalam klasifikasi email spam, setiap email dalam set pelatihan akan diberi label "spam" atau "bukan spam".

4. Apa dua tugas supervised yang paling umum? Dua tugas *supervised learning* yang paling umum adalah **regresi** dan **klasifikasi**. Tugas regresi adalah memprediksi nilai numerik target, seperti harga mobil. Tugas klasifikasi adalah untuk mengkategorikan instance ke dalam kelas tertentu, seperti mengklasifikasikan email sebagai spam atau bukan spam.

5. Dapatkah Anda menyebutkan empat tugas unsupervised yang umum? Empat tugas *unsupervised learning* yang umum adalah:

- **Clustering:** Mengelompokkan instance yang serupa ke dalam grup atau *cluster*.
- **Visualization and Dimensionality Reduction:** Menggambarkan data dalam representasi 2D atau 3D yang mudah dipahami, atau menyederhanakan data tanpa kehilangan banyak informasi.
- **Anomaly Detection:** Mendeteksi instance yang tidak normal atau *outliers*.
- **Association Rule Learning:** Menemukan hubungan menarik antara atribut dalam data dalam jumlah besar.

6. Jenis algoritma Machine Learning apa yang akan Anda gunakan untuk memungkinkan robot berjalan di berbagai medan yang tidak diketahui? Untuk memungkinkan robot berjalan di medan yang tidak diketahui, jenis algoritma yang paling cocok adalah **Reinforcement Learning (RL)**. Dalam RL, *agent* (robot) belajar dengan mengamati lingkungan, melakukan tindakan, dan menerima *rewards* atau *penalties*. Robot akan belajar *policy* (strategi) terbaik untuk memaksimalkan *reward*-nya dari waktu ke waktu, yang dalam kasus ini adalah berhasil berjalan tanpa jatuh.

7. Jenis algoritma apa yang akan Anda gunakan untuk mengelompokkan pelanggan Anda ke dalam beberapa grup? Jika Anda ingin mengelompokkan pelanggan ke dalam beberapa grup, Anda akan menggunakan algoritma **clustering** (unsupervised learning). Algoritma ini dapat mencoba mendeteksi grup pelanggan yang serupa berdasarkan data seperti riwayat pembelian dan aktivitas di situs web.

8. Apakah Anda akan membingkai masalah deteksi spam sebagai masalah *supervised learning* atau *unsupervised learning*? Masalah deteksi spam biasanya dibingkai sebagai masalah **supervised learning**. Hal ini karena sistem dapat dilatih dengan banyak contoh email yang telah diberi label oleh pengguna sebagai "spam" atau "bukan spam" (ham).

9. Apa itu sistem *online learning*? Sistem *online learning* adalah sistem yang dapat belajar secara bertahap (*incrementally*) dari aliran data yang masuk, baik secara individual maupun dalam kelompok kecil yang disebut *mini-batches*. Setiap langkah pembelajaran cepat dan murah, memungkinkan sistem untuk belajar dari data baru secara *on-the-fly* saat data tersebut tiba.

10. Apa itu *out-of-core learning*? *Out-of-core learning* adalah penggunaan algoritma *online learning* untuk melatih sistem pada dataset yang sangat besar yang tidak muat dalam memori utama satu mesin. Algoritma memuat sebagian data, menjalankan langkah pelatihan pada data tersebut, dan mengulangi proses tersebut hingga semua data telah diproses.

11. Jenis algoritma pembelajaran apa yang mengandalkan ukuran kesamaan (*similarity measure*) untuk membuat prediksi? Algoritma pembelajaran yang mengandalkan ukuran kesamaan untuk membuat prediksi adalah **instance-based learning**. Sistem ini mempelajari contoh-contoh data dengan "menghafalnya", lalu menggeneralisasi ke kasus-kasus baru dengan membandingkannya dengan contoh yang telah dipelajari menggunakan ukuran kesamaan.

12. Apa perbedaan antara *model parameter* dan *hyperparameter* algoritma pembelajaran?

- **Model parameter** adalah parameter dari sebuah model yang nilainya ditentukan dari data selama proses pembelajaran. Parameter ini menentukan apa yang akan diprediksi oleh model pada instance baru (contohnya adalah kemiringan dan perpotongan pada model linear).
- **Hyperparameter** adalah parameter dari algoritma pembelajaran itu sendiri, bukan dari model. Nilainya tidak dipelajari oleh algoritma pembelajaran, melainkan harus diatur sebelum pelatihan dan tetap konstan selama pelatihan (contohnya adalah *learning rate* pada *gradient descent* atau jumlah *regularization* yang akan diterapkan).

13. Apa yang dicari oleh algoritma *model-based learning*? Apa strategi paling umum yang mereka gunakan untuk berhasil? Bagaimana cara mereka membuat prediksi? Algoritma *model-based learning* mencari **nilai optimal untuk parameter model** sehingga model tersebut

dapat menggeneralisasi dengan baik pada instance baru. Strategi paling umum yang mereka gunakan untuk berhasil adalah dengan **meminimalkan sebuah *cost function*** yang mengukur seberapa buruk model dalam membuat prediksi pada data pelatihan. Untuk membuat prediksi, mereka memasukkan fitur dari instance baru ke dalam **fungsi prediksi model**, menggunakan parameter yang telah ditemukan oleh algoritma pembelajaran.

14. Dapatkah Anda menyebutkan empat tantangan utama dalam Machine Learning? Empat tantangan utama dalam Machine Learning adalah:

- **Kuantitas data pelatihan yang tidak mencukupi** (*Insufficient Quantity of Training Data*).
- **Data pelatihan yang tidak representatif** (*Nonrepresentative Training Data*).
- **Data berkualitas buruk** (*Poor-Quality Data*).
- **Fitur yang tidak relevan** (*Irrelevant Features*).
- (Tantangan lain termasuk *overfitting* dan *underfitting*).

15. Jika model Anda berkinerja baik pada data pelatihan tetapi menggeneralisasi dengan buruk pada instance baru, apa yang sedang terjadi? Dapatkah Anda menyebutkan tiga solusi yang mungkin? Jika model berkinerja baik pada data pelatihan tetapi buruk pada instance baru, model tersebut kemungkinan besar mengalami **overfitting** pada data pelatihan. Tiga solusi yang mungkin adalah:

- **Menyederhanakan model:** Ini bisa dilakukan dengan memilih model dengan parameter yang lebih sedikit (misalnya model linear daripada model polinomial tingkat tinggi), mengurangi jumlah atribut dalam data pelatihan, atau membatasi model (*regularization*).
- **Mengumpulkan lebih banyak data pelatihan.**
- **Mengurangi noise dalam data pelatihan** (misalnya, memperbaiki kesalahan data dan menghapus *outliers*).

16. Apa itu *test set*, dan mengapa Anda ingin menggunakannya? *Test set* adalah sebagian dari data yang Anda sisihkan dan tidak pernah Anda gunakan untuk melatih model. Anda ingin menggunakannya untuk **mengestimasi *generalization error*** (atau *out-of-sample error*) dari model Anda pada instance yang belum pernah dilihat sebelumnya. Ini memberi Anda gambaran seberapa baik kinerja model Anda di dunia nyata sebelum Anda meluncurkannya.

17. Apa tujuan dari *validation set*? Tujuan dari *validation set* adalah untuk **membandingkan beberapa model kandidat dan memilih yang terbaik**. Anda melatih beberapa model dengan berbagai *hyperparameter* pada *training set* yang dikurangi (set pelatihan penuh dikurangi *validation set*), dan Anda memilih model yang berkinerja terbaik pada *validation set*. Proses ini disebut *hyperparameter tuning* dan *model selection*.

18. Apa itu *train-dev set*, kapan Anda membutuhkannya, dan bagaimana Anda menggunakannya? *Train-dev set* adalah bagian dari set pelatihan yang disisihkan (model tidak dilatih di atasnya). Anda membutuhkannya ketika ada risiko ketidaksesuaian (*mismatch*) antara data pelatihan dengan data yang digunakan dalam set validasi dan set tes. Cara menggunakannya adalah:

1. Model dilatih pada sisa set pelatihan (bukan pada *train-dev set*).
2. Model dievaluasi pada *train-dev set* dan *validation set*.
3. Jika model berkinerja baik pada set pelatihan tetapi tidak pada *train-dev set*, maka model tersebut kemungkinan *overfitting*.
4. Jika model berkinerja baik pada set pelatihan dan *train-dev set* tetapi tidak pada *validation set*, maka kemungkinan ada ketidaksesuaian data yang signifikan antara data pelatihan dan data validasi/tes.

19. Apa yang bisa salah jika Anda menyetel *hyperparameter* menggunakan *test set*? Jika Anda menyetel *hyperparameter* menggunakan *test set*, Anda berisiko **mengalami *overfitting* pada *test set***. Akibatnya, estimasi *generalization error* yang Anda ukur akan terlalu optimis, dan model yang Anda luncurkan kemungkinan akan berkinerja lebih buruk dari yang diharapkan saat dihadapkan pada data baru yang sesungguhnya.