Nama : Bagus Mahardika Santoso

NIM : 1103204028

## Understanding 3 link StatQuest

Principal Component Analysis (PCA) adalah teknik pengurangan dimensi yang populer dalam bidang data sains dan pembelajaran mesin. Tujuan utama dari PCA adalah untuk mengurangi jumlah variabel dalam dataset sambil tetap mempertahankan sebanyak mungkin informasi dari data aslinya.

Konsep Dasar PCA:

- Variabilitas: Dalam PCA, komponen yang memiliki variabilitas (varians) yang tinggi dianggap penting karena mereka menangkap sebagian besar informasi dalam data.
- Orthogonalitas: Komponen yang dihasilkan oleh PCA adalah orthogonal (saling tegak lurus), yang berarti mereka tidak berkorelasi satu sama lain.
- Transformasi: PCA mengubah data asli ke koordinat sistem baru yang disebut principal components (komponen utama).

menerapkan Principal Component Analysis:

1) Standarisasi Data: Skala fitur penting dalam PCA. Pertama-tama, kita perlu menstandarisasi data, yang berarti setiap fitur harus memiliki rata-rata 0 dan deviasi standar 1.
2) Hitung Matriks Kovariansi: Ini akan mengukur bagaimana perubahan dalam satu fitur mempengaruhi perubahan dalam fitur lain.
3) Hitung Eigenvalues dan Eigenvectors: Dari matriks kovariansi, kita dapat menemukan eigenvalues dan eigenvectors-nya. Eigenvectors akan menentukan arah komponen utama, sementara eigenvalues akan menentukan besar variabilitas yang ditangkap oleh masing-masing komponen.
4) Urutkan Eigenvalues: Urutkan eigenvalues dalam urutan menurun dan pilih n eigenvectors teratas, di mana n adalah jumlah komponen utama yang ingin Anda tetapkan.
5) Transformasi Data: Gunakan eigenvectors yang dipilih untuk mengubah data asli ke koordinat sistem baru.

K-Nearest Neighbors Algorithm atau biasa dipanggil K-NN merupakan cara yang sangat sederhana untuk mengklasifikasikan data. Algoritma ini mencoba mengklasifikasikan data baru dengan cara mencari k-nearest neighbors (k tetangga terdekat) dari data baru tersebut di dalam ruang fitur. Algoritma ini mencoba mengklasifikasikan data baru dengan cara mencari data menggunakan algoritma ini dari data baru tersebut di dalam ruang fitur. Cara kerja K-NN sebagai berikut:

- Menentukan Nilai: K adalah jumlah tetangga terdekat yang akan digunakan untuk memutuskan kelas dari data baru. Nilai K ini harus dipilih dengan hati-hati, karena dapat mempengaruhi kinerja algoritma.
- Mengukur Jarak: Algoritma ini mengukur jarak antara data baru dan semua titik data dalam set pelatihan. Jarak ini dapat diukur dengan berbagai metrik, seperti jarak Euclidean atau jarak Manhattan.
- Memilih Tetangga Terdekat: Algoritma memilih K tetangga terdekat dengan jarak terpendek dari data baru.
- Voting: Ada dua pendekatan umum untuk menentukan kelas dari data baru, Voting berarti kelas yang paling umum di antara tetangga terpilih dipilih sebagai kelas untuk data baru.
  Weighted Voting, Jarak dapat digunakan sebagai bobot untuk mempengaruhi keputusan kelas.
- Menentukan Kelas Data Baru: Data baru diklasifikasikan berdasarkan mayoritas kelas dari tetangga terdekat.

K-NN memiliki keunggulan dalam kemudahan implementasi dan interpretasi. Namun, metodenya dapat menjadi lambat jika datasetnya sangat besar, dan pemilihan parameter K yang tepat adalah kunci untuk mendapatkan hasil yang baik.

K-NN sering digunakan dalam pemodelan klasifikasi dan regresi, dan dapat menjadi salah satu langkah awal yang baik dalam eksplorasi data sebelum mencoba algoritma pembelajaran mesin yang lebih kompleks.

Decision Trees merupakan algoritma yang memungkinkan pengambilan keputusan berbasis serangkaian aturan dan percabangan. Algoritma ini dapat digunakan untuk melakukan klasifikasi atau regresi.

Classification Trees adalah tipe spesifik dari Decision Trees yang digunakan untuk tugas klasifikasi, di mana tujuannya adalah memprediksi kelas atau label dari sampel data.

Gini Impurity adalah salah satu metrik yang digunakan untuk mengukur seberapa *impur*e data yang sedang kita olah. Pada setiap simpul dalam pohon, Gini Impurity mengukur kemungkinan salah satu sampel acak dipilih dari kumpulan data akan salah diklasifikasikan berdasarkan distribusi kelas di dalam simpul tersebut.

Impurity ini merujuk pada tingkat ketidakmurnian simpul atau cabang dalam pohon. Semakin rendah impuritynya, semakin baik atau bersih simpul tersebut dalam memisahkan kelas. Impurity dapat diukur dengan metrik seperti Gini Impurity atau Entropi.

Overfitting adalah masalah umum yang terkait dengan Decision Trees Overfitting terjadi karena data atau model terlalu kompleks sehingga tidak dapat melakukan prediksi yang baik pada data baru. Terdapat 2 cara mencegah overfitting yang disebutkan pada video tersebut, yaitu:

- Pruning: Menghilangkan cabang-cabang yang tidak memberikan kontribusi signifikan pada performa prediksi. Ada dua jenis pruning: Pre-pruning (pemangkasan sebelum membangun pohon) dan Post-pruning (pemangkasan setelah pohon dibangun).
- Cross-Validation: Menguji performa model pada set validasi terpisah dari data latih dapat membantu mengidentifikasi overfitting.