**Mempelajari Machine Learning dengan Python**

Nama : I Putu Ananta Wijaya
Program : Introduction to Python for Data Science

Instruktor : Raka Ardhi

1) Apa itu Machine Learning (ML) ?
   - ML bisa dialokasikan ke hampir semua industri. ML adalah AI yang menyediakan sebuah sistem untuk belajar dari experience tanpa kita harus 'ngoding' directly. Kita ngasi data, mesin ini yang akan berpikir untuk kita sehingga save time.
   - ML ada di sekitar kita, membedakan email spam, rekomendasi yang dibeli user, mobil tanpa awak. Komputer mempelajari data yang ada, properti dan sebagainya, dan berdasarkan itu bisa membedakan/memberi saran ke user.
   - Traditional Programming perlu logic code: if, case, while/for. Sedangkan, Machine Learning perlu data, algoritma, analisis, dan model.
   - Dua tipe machine learning: Supervised Learning & Unsupervised. Supervised contohnya prediksi rumah. Unsupervised mendapatkan cluster data berdasarkan kesamaan tiap data.
2) Persiapan untuk ML ?
   - Jupyter Notebook, Python 3, dan Anaconda.
3) Machine Learning Workflow
   - Permasalahan → Persiapan Data → Memilih Algoritma → Melatih Model → Uji Model.
   - Memilih algoritma berdasarkan permasalahan. Ditaruh ke-3 karena kita perlu data juga. Kalau hasilnya jelek, boleh direfine ke step sebelumnya-sebelumnya. Selalu evaluasi.
4) Mendapatkan Data
   - Membersihkan data supaya bisa dimaksimalkan oleh ML. Tidak ada redundant dan hilangkan yang tidak ada korelasi. Variabel adalah kolom, observasinya adalah baris.
   - 50-80% digunakan untuk membersihkan data.
   - Mendapatkan data jangan dari Google, lebih ke data perusahaan atau pemerintahan karena lebih akurat.
5) Load, Clean, Inspect Data
   - Load data dengan library pandas dan function read_csv. Menampilkan dengan .head atau .tail.
   - Cek data yang kosong dengan isnull().value.any(). Pastikan data tidak ada duplicate.
   - Buatkan corelation plot sehingga tau ada korelasi dan tidaknya.
6) Merapihkan Data
   - Ubah data sesuai tipenya. Misalnya perlu numerik, ubah ke numerik.
7) Memilih Algoritma
   - Memilih algroitma berdasarkan faktor: learning type(unsupervised/supervised), result(perlu classification atau regression), complexity(simple vs complex), basic vs enchanced.
   - Dapat kandidat algoritma, disini tentukan keunggulan dan kelemahan algoritma. Tidak lupa juga alur algoritmanya gimana sehingga cocok dengan hasil yang dituju.
   - Contoh disini, pilih Naïve bayes karena perlu sedikit data dan probability based.
8) Melatih Model
   - Data melatih algoritma ML sehingga menghasilkan model. Model disini spesifik, datanya spesifik algoritmanya. Proses sebelumnya harus dilakukan dengan spesifik dan benar sehingga disini mendapatkan model yang baik.
   - Proses training dimulai dari split data untuk 70% training dan 30% testing. Setelah itu, train model. Kalau 100% kita pakai training ya ga bisa testingnya, prediksinya ga bener. Setelah itu,

evaluasi model, yang mana perlu di improve. Trus kalau ada fitur yang tidak digunakan bisa dibuang supaya training-nya lebih cepat.

- Dengan python, kita menggunakan scikit-kearn, mulai dari data splitting, preprocessing, feature selection, model training.
- Data splitting menggunakan function train_test_split, dengan atribut split_test_size.
- Row missing bisa didiamkan, atau bisa diganti. Kalau banyak jangan di hapus, coba ganti. Impute value dengan fill_0.fit_transofrm(…).
- Setelah itu bisa dilakukan pembuatan model dengan pemanggilan fungsi.

9) Menguji Akurasi

- Mendefinisikan datanya apakah sudah akurat (modelnya sudah bisa dipakai) atau mauditingkatkan performanya lagi. Gunakan metrics dari sklearn bisa diapatka akurasi dengan mengirim predictnya dengan hasil aslinya. Kita bisa menggunakan confussion matrics untuk melihat label yang benar terprediksi. Fungsi yang digunakan metrics.confusion_matrix(). Akan muncul true negative, true positive, dsb.
- Peningkatan performa dengan beberapa cara yaitu misalnya menggunakan model algoritma lain seperti RandomForestClassifier, LogisticRegression. Kalau ga berubah juga, kembali ke data processing atau split data atau balancing data untuk hasil prediksi yang tidak balance.
- Peningkata performa dengan cross validation, dimana data diulang pengelompokkanya dan ditest masing-masing kelompoknya. Untuk menyimpan model gunakan fungsi joblib. Setelah itu bisa load ulang dengan joblib.load()