Nama : Ghina Khoerunnisa
Program : Data Scientist Intern

Mempelajari Machine Learning dengan Python

Workflow Machine Learning : Permasalahan >> Persiapan data >> Memilih Algoritma >> Melatih Model >> Uji Model.

Mempersiapkan Data : Kita membutuhkan tidy data (lebih mudah dimanipulasi, variabel adalah kolomnya, observasi adalah barisnya). Data dapat didapatkan pada Google, Kaggle, Data Pemerintahan, Data Perusahaan. Setelah mendapatkan datanya, kita perlu meload, clean, dan inspect datanya. Load data dapat dilakukan dengan pd.read_csv('filename.csv'), untuk melihat ukuran datanya dapat menggunakan df.shape. Disini sebelah kiri merupakan jumlah baris dan kanannya merupakan jumlah kolomnya.Untuk melihat 5 data pertama dapat menggunakan df.head(), jika ingin melihat 5 data terakhir dapat menggunakan df.tail(). Kemudian cek apakah ada data yang kosong dengan cara df.isnull().values.any(). Lalu cek korelasi antar variabelnya karena kita harus memastikan data kita tidak repetitive agar mesinnya tidak bingung. Ketika ada data yang korelasinya tinggi maka salah satunya kita hapus. Setelah itu kita rapikan kembali datanya dengan cara memastikan seluruh tipe datanya tepat dan cek distribusinya.

Memilih Algoritma : Faktor-faktor dalam memilih algoritma adalah learning type, result, complexity, dan basic vs enhanced.Learning type ini ada dua supervised dan unsupervised tergantung dengan problem kita. Misalnya pada data diabetes nah itu berarti learning typenya supervised. Kemudian resultnya ada dua dari supervised yaitu regression (continuous value) atau classification (discrete value atau true/false). Karena initial algorithm (algoritma di awal untuk membuat model) maka pilih saja algoritma yang simpel. Sama seperti complexity, karena initial algorithm maka pilih saja yang basic. Kemudian kita bisa mempersempit lagi pilihan algoritmanya berdasarkan kasus yang ingin diselesaikan. Pada course ini setelah berdasarkan 4 faktor didapatkan algoritma naive bayes, logistic regression, dan decision tree. Naive bayes mempelajari seperti apa data orang yang diabetes dari data sebelumnya. Logistic Regression dengan pembobotan setiap fitur, lalu diproses sedemikian rupa sehingga mendapatkan nilai yang mana nilai tersebut lebih besar ke 1 atau ke 0. Decision tree, bentuknya seperti binary tree nah di decision tree ini kita harus punya banyak data untuk tahu nodenya dan split nodenya itu seperti apa. Oleh karena itu pada task diabetes dipilih algoritma naive bayes yang mengandalkan probability, bobotnya sama, dan membutuhkan sedikit data. Selain itu naive bayes juga simpel sehingga lebih mudah dipahami, performance nya cepat, dan juga stabil nantinya jika terjadi perubahan data dan dan sebagainya.

Melatih Model : Sebelum melatih model, model training itu sendiri merupakan kita membiarkan data yang spesifik untuk melatih algoritma learning agar dapat menghasilkan model yang spesifik seperti yang kita inginkan. Proses training yang pertama harus dilakukan adalah split datanya terlebih dahulu. Contohnya 70% data kita gunakan sebagai data training dan 30% data digunakan untuk testing. Kemudian setelah di split kita train model dengan data train lalu kita evaluasi modelnya dengan data test. Pada python kita bisa menggunakan library scikit-learn untuk melatih model.

Menguji Akurasi Model : Uji akurasi adalah proses dimana kita memasukkan data test ke dalam model yang telah dilatih kemudian nanti kita akan menentukan apakah kita perlu meningkatkan performansinya atau tidak. Untuk melihat performansi dari model yang telah

dilatih dapat menggunakan confusion matrix dan classification report. Pada confusion matrix terdapat TN (Predicted 0, Actual 0), FP (Predicted 1, Actual 0), FN (Predicted 0, Actual 1), TP (Predicted 1, Actual 1). Kemudian pada classification report kita dapat melihat score dari precision dan recall nya. Karena pada kasus diabetes kita tidak ingin adanya FP dan FN (lebih berbahaya) maka kita perlu menaikkan performansi precision dan recallnya. Selain itu ada cross validation, disini Cross Validation membagi data train menjadi train dan validation. Misal kita punya train 70% nah dengan cross validation akan membagi data menjadi 10 potongan (kalo cvnya 10) untuk menjadi data train dan data validasinya. Untuk menggunakan cross validation dapat menambahkan CV pada algoritma di scikit learnnya.