BAB 8

Ensemble Method

Pengambilan keputusan yang dilakukan oleh manusia dalam kehidupan sehari-hari seringkali dipengaruhi oleh pendapat dari orang lain ataupun pendapat orang banyak. Seperti misalnya dalam pemilihan untuk menentukan pembelian Handphone berdasarkan pertimbangan review-review dari produk. Contoh lain pemilihan menentukan buku yang ingin dibeli, bisa saja suatu topik yang sama terdapat beberapa buku yang memiliki judul yang berbeda ataupun pengarang yang berbeda, sehingga dalam pemilihan buku mana yang akan dibeli manusia sering mempertimbangkan review-review dari buku-buku tersebut. Seringkali, pengambilan keputusan oleh sekelompok individu menghasilkan hasil yang lebih baik daripada keputusan yang dibuat oleh salah individu saja.

8.1 Pengertian Ensemble Method

Machine Learning sering kali melibatkan penyesuaian dan juga evaluasi dari suatu model pada dataset. Penyesuaian dan evaluasi ini bertujuan untuk mencari model mana yang paling tepat / memiliki performa yang baik pada dataset. Mengingat bahwa kita tidak dapat mengetahui model mana yang akan memiliki performa yang terbaik pada dataset, biasanya pencarian model terbaik akan dilakukan dengan cara melibatkan banyak trial and error sampai ditemukan model yang memiliki performa terbaik untuk dataset. Penggunaan trial dan error untuk menemukan model terbaik tentunya akan membutuhkan waktu relatif lama. Metode ensemble mengatasi masalah ini, dimana metode ensemble dengan cara menyiapkan beberapa model yang berbeda kemudian menggambungkan prediksi dari model-model tersebut.

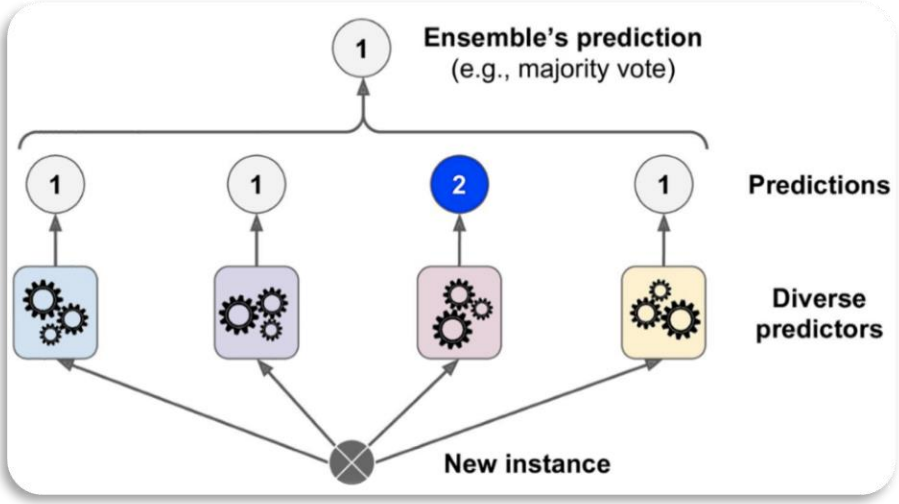
Beberapa pengertian ensemble method

a. Ensemble method merupakan kombinasi model yang memiliki performa lebih baik jika dibandingkan masing-masing model penyusunnya.
b. Ensemble method merupakan salah satu Teknik machine learning yang menggabungkan beberapa model untuk menghasilkan satu model prediksi yang optimal.
c. Ensemble method merupakan Teknik yang membuat beberapa model dan kemudian menggabungkannya untuk menghasilkan hasil yang lebih baik, biasanya menghasilkan solusi yang lebih akurat daripada satu model saja.
d. Ensemble method merupakan Suatu teknik untuk menggabungkan prediksi beberapa model dasar yang dibangun dengan algoritma pembelajaran yang

diberikan untuk meningkatkan kemampuan generalisasi / kehandalan dibandingkan satu model saja.

e. Ensemble method disebut sebagai Seni menggabungkan beragam kelompok algoritma (model individu) secara bersama-sama untuk berimprovisasi pada stabilitas dan kekuatan prediksi model.

Ilustrasi dari ensemble method dapat dilihat pada gambar 8.1. pada ilustrasi tersebut adalah contoh contoh teknik yang paling sederhana. Misalnya, jika ensemble method diterapkan pada masalah klasifikasi maka dapat dipilih beberapa algoritma klasifikasi untuk melakukan pelatihan dan prediksi pada suatu dataset. Algoritma yang dapat digunakan seperti Logistic Regression classifier, SVM classifiers, KNN, decision tree, dan lain-lain. Setiap algoritma akan mengasilkan prediksinya masing-masing dan prediksi akhir dari ensemble method untuk permasalahan klasifikasi adalah dengan cara memilih hasil prediksi yang paling sering muncul / mayoritas. Pada gambar 8.1 hasil prediksi dari tiga buah algoritma adalah kelas 1 sehingga hasil akhir prediksi adalah kelas 1 untuk data training yang baru.



Gambar 8. 1 ILustrasi Ensemble Method

8.2 Tipe dari Ensemble Method

Ensemble method dikategorikan kedalam dua buah kelompok yaitu metode ensemble homogen dan metode ensemble heterogen.

a. Metode ensemble homogen

Homogen dalam ensemble method berarti model klafisifkasi atau model regresi yang dibangun menggunakan algoritma yang sama hanya saja dataset yang digunakan adalah resampling dari dataset awal. Sehingga mungkin saja algoritma yang sama diterapkan terhadap sub-dataset yang berbeda-beda. Jenis metode ansambel ini biasanya digunakan untuk sejumlah besar kumpulan data. Metode yang termasuk kedalam metode ensemble homogen adalah bagging dan boosting.

b. Metode ensemble heterogen

Metode ansambel heterogen itu adalah kombinasi dari berbagai jenis algoritma klasifikasi dimana algoritma-algoritma yang berbeda diterapkan terhadap data yang sama. Metode yang termasuk kedalam metode ensemble heterogen adalah Stacking.

8.3 Bagging dan Random Forest

Sebelum membahas mengenai Bagging dan random forest, terlebih dahulu akan dibahas mengenai Teknik untuk melakukan resampling data yaitu Metode Bootstap.

A. Bootstrap

Bootstrap merupakan salah satu metode statistika yang dapat digunakan untuk melakukan resampling data. Proses pembuatan satu sampel data adalah sebagai berikut:

1. Pilih ukuran sampel
2. Ketika ukuran dari sampel saat ini lebih kecil dari ukuran yang dipilih
    a. Pilih observasi secara acak dari dataset
    b. Tambahkan kedalam sample
    c. Ulangi Langkah a sampai ukuran sample sesuai ukuran yang dipilih

Bootsrap dapat juga digunakan untuk estimasi performa dari machine learning yaitu dengan Teknik boostrap didapatkan data training dan juga data testing. Sampel ini tidak termasuk dalam sampel tertentu disebut sampel out-of-bag, atau disingkat OOB. OOB inilah yang nantinya akan menjadi data testing. Langkah-langkah yang dapat digunakan adalah sebagai berikut :

1. Pilih sejumlah sampel bootstrap yang akan dilakukan
2. Pilih ukuran sampel
    a. Pilih observasi secara acak dari dataset

   b. Tambahkan kedalam sample
   c. Ulangi Langkah a sampai ukuran sample sesuai ukuran yang dipilih
3. Untuk setiap sampel bootsrap lakukan :
   a. Gambarlah sampel dengan pengganti dengan ukuran yang dipilih
   b. Lakukan pelatihan pada sampel data
   c. Estimasi performa dari model menggunakan out-of-bag data
4. Hitung rata-rata prediksi setiap sample sebagai estimasi performa akhir

Contoh dengan satu iterasi:

Dataset : : [0,1, 0,2, 0,3, 0,4, 0,5, 0,6]

1. Tentukan ukuran sampel, Misal : 4
2. Pilih observasi pertama secara acak dari dataset. Pilih 0,2.
3. Ulangi langkah no 2 sebanyak 3 kali lagi. Sehingga sample data [0,2, 0,1, 0,2, 0,6]
   Catatan : nilai yang sama dapat muncul dalam sampel data sebanyak lebih dari satu kali

out-of-bag =[0.3, 0.4, 0.5]

train = [0.2, 0.1, 0.2, 0.6]

test = [0.3, 0.4, 0.5]

B. Bagging

Bagging merupakan singkatan dari bootstrap aggregating. Bagging dapat digunakan dalam permasalah klasifikasi dan regresi. Ketika digunakan untuk permasalahan regresi maka hasil prediksi dari bagging adalah rata-rata dari prediksi dari setiap model. Sedangkan jika digunakan untuk permasalahan klasifikasi hasil prediksi dari bagging adalah kelas mayoritas. Bagging merupakan prosedur umum yang dapat digunakan untuk mengurangi varians untuk algoritma yang memiliki varians tinggi. Contoh algoritma yang memiliki variance yang tinggi adalah Classification and Regression Tree (CART).

CART sangat sensitive terhadap data pelatihan yang spesifik. Jika data pelatihan diubah (misalnya, CART dilatih pada subset data training), CART yang dihasilkan bisa sangat berbeda sehingga dapat menghasilkan prediksi yang sangat berbeda. Asumsikan terdapat dataset yang memiliki 1000 baris data(x) dan akan digunakan

algoritma CART. Maka tahapan Bagging dengan menggunakan algoritma CART adalah sebagai berikut :

1. Buat resample data / sub-data dengan acak dari dataset. Missal dibuat masing-masing sub-data memiliki 100 baris data. Salah satu tenik resampling data yang dapat digunakan adalah Metode Boostrap.
2. Lakukan pelatihan menggunakan CART untuk setiap sub-set data
3. Dengan adanya sub-data baru, maka akan didapatkan hasil prediksi dari masing-masing data. Untuk persamalan klasifikasi hasil prediksi akhir didapat dari mayoritas kelas yang muncul, sedangkan untuk permasalan regersi hasil prediksi didapat dengan cara menghitung prediksi rata-rata dari setiap model.

Contoh : Jika 5 sub-dataset, dari setiap sub-data akan dilakukan pelatihan dengan menggunakan CART. Sehingga didpatkan 5 buah hasil prediksi missal hasil prediksi yang didapatkan adalah model_1 : biru, model_2: biru, model_3: merah, model_4: biru, dan model_5: merah. Maka kelas yang paling sering muncul / mayoritas yaitu kelas biru menjadi hsil prediksi dari metode bagging.

C. Random Forest

Random forest merupakan algoritma yang dikembangkan oleh Leo Breiman dan Adele Cutler. Random Forests merupakan perbaikan dari metode bagging yang menggunakan algoritma decision tree. Pada metode bagging, decision tree yang dibuat dapat memiliki banyak kesamaan struktur dan sehingg dapat menghasilkan korelasi yang tinggi. Sedangkan dengan menggambungkan hasil prediksi dari beberapa model dalam ensemble method akan lebih handal ketika melakukan prediksi terhadap sub-model yang tidak berkorelasi atau korelasinya rendah. Random forest mengubah algoritma pembelajaran pada sub-tree sehingga prediksi yang dihasilkan oleh semua sub-tree dapat memiliki korelasi yang kecil.

Dalam decision tree saat memilih split poin maka algoritma pembelajaran di izini=kan untuk mempertimbangkan seluruh fitur/ variable x untuk memilih split-poin paling optimal. Random forest merubah prosedur ini sehingga pada random forest penentuan split-poin tidak dilkukan terhadap keseluruhan fitur akan tetapi algoritma pembelajaran dibatasi pada beberapa fitur yang dipilih secara acak. Jumlah fitur yang dapat dicari untuk setiap split-point(m) harus ditentukan dan menjadi parameter dari

algoritma. Dapat dilakukan ujicoba dengan nilai yang berbeda-beda dan disesuaikan dengan menggunakan cross validation.

Untuk permasalahan klasifikasi secara default jumlah fitur untuk setiap split-point(m) yang baik adalah :

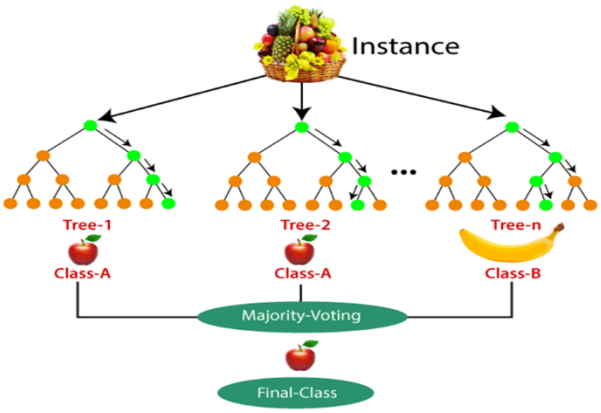
$$m = \sqrt{p}$$

Untuk permasalahan regresi secara default jumlah fitur untuk setiap split-point(m) yang baik adalah :

$$m = \frac{p}{3}$$

Di mana m adalah jumlah fitur yang dipilih secara acak yang dapat dicari pada split-point dan p adalah jumlah variabel input (jumlah fitur). Misalnya, jika terdapat sekumpulan data memiliki 16 variabel masukan, maka untuk permasalahan klasifikasi didapatkan:

$$m = \sqrt{16} = 4$$



Gambar 8. 2 Ilustrasi random forest

Ilustrasi dari random forest dapat dilihat pada gambar 8.2 pada gambar tersebut dibuat tiga buah sub-data. Penentuan sub data dapat menggunakan Teknik bootstrap. Selanjutnya Algoritma decision tree diterapkan ke masing-masing subdata. Pembuatan tree dapat menggunakan Gini index ataupun information gain untuk mencari rood node pada setiap tree. Kemudian lakukan prosedur sesuai dengan algoritma decision tree untuk membuat tree dari setiap sub-data. Lakukan pengecekan performa model dengan :

- Training Error : Lakukan Pengujian dengan Menggunakan data training yang termasuk dalam Sub-Dataset

- Testing Error :L akukan Pengujian dengan Menggunakan data Out-of-Bag (data yang tidak termasuk dalam Sub-Dataset)

Untuk setiap subdata akan menghasilkan prediksi kelas yang selanjutnya diambil kelas mayoritas sebagai keputusan akhir.

8.4 Boosting dan Adaboost

A. Boosting

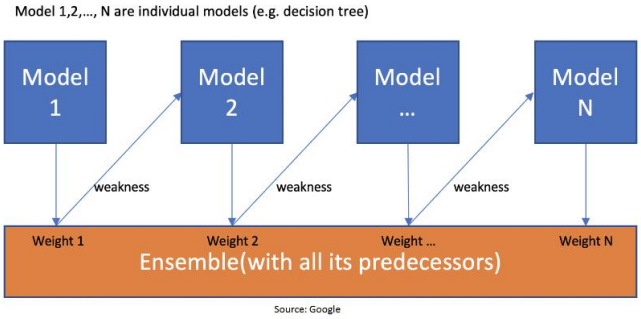
Teknik Boosting dirumuskan oleh Yoav Freund dan Robert Schapire pada tahun 1995. Boosting merupakan salah satu metode ensemble yang dapat digunakan untuk mengurangi bias pada algoritma pembelajaran. Boosting dapat digunakan untuk permasalahan klasifikasi maupun permasalahan regresi. Teknik boosting menggunakan pemrosesan secara sequential dimana keluaran dari proses sembelumnya akan menjadi masukan dari untuk proses selanjutnya.

Boosting menciptakan "strong Classifier" dari sejumlah "weak classifier". Hal ini dilakukan dengan cara membangun model dari data training, kemudian membuat model kedua yang mencoba untuk memperbaiki error dari model pertama. Kemudian Model akan terus ditambahkan hingga data training dapat diprediksi dengan sempurna atau dibatasi berdasarkan jumlah model maksimum yang ditambahkan.

B. Adaboost

Adaboost merupakan singkatan dari **Adaptive Boosting.** AdaBoost adalah algoritma boosting pertama yang sukses dikembangkan untuk binary classification. AdaBoost dapat digunakan untuk meningkatkan kinerja algoritma pembelajaran mesin apa pun. Ini paling baik digunakan dengan "weak-learner". Algoritma yang paling cocok dan paling umum digunakan dengan AdaBoost adalah decision tree dengan satu level. Yaitu tree yang hanya memiliki 1 rood node dan node yang lainnya adalah leafnode. Tree ini sering disebut dengan istilah *decision stumps.*

Ilustrasi mekanisme adaboost dapat dilihat pada gambar 8.3



Gambar 8. 3 Mekanisme dari adaboost

Tahapan algoritma Adaboost

1. Berikan bobot untuk setiap instance pada data training, formula untuk inisialisasi bobot yang dapat digunakan adalah:

$$Weight(xi) = \frac{1}{n}$$

dimana xi adalah instance pelatihan ke-i dan n adalah banyaknya data pada data training

2. Membuat **stump**

Yaitu dengan cara membuat *stump* untuk setiap fitur. Pada binary classification stump membuat satu keputusan pada satu variabel masukan dan mengeluarkan nilai +1.0 atau -1.0 untuk nilai kelas pertama atau kedua.

3. Hitung **Total Error (TE)**

$$error = \frac{(correct - n)}{n}$$

Dimana error adalah tingkat kesalahan klasifikasi, correct jumlah data pada data training yang diprediksi dengan benar oleh model dan n adalah banyaknya data pada data training

Contoh : jika model memprediksi 70 data secara benar dari keseluruhan data yang berjumlah 100 maka error would be (70-100)/100 or 0.3.

Total Error

$$Terror = \frac{\sum_{i=1}^{n}(w_i * Perror)}{\sum_{i=1}^{n} w}$$

dimana w adalah bobot untuk data pelatihanke- i dan perror adalah kesalahan prediksi untuk ata pelatihan i yaitu 1 jika salah diklasifikasikan dan 0 jika diklasifikasikan dengan benar.

Contoh : jika terdapat tiga buah instance data pelatihan dengan 0.01, 0.5 dan 0.2. hasil prediksi model adalah -1, -1 dan -1, sedangkan berdasarkan data actual adalah -1, 1 dan-1. Sehingga perror yang dihasilkan adalah 0, 1, and 0. misclassification rate Terror dapat dihitung sebagai berikut :

$$error = \frac{\sum_{i=1}^{n}(w_i * Perror)}{\sum_{i=1}^{n} w} = \frac{(0.01*0 + 0.5*1 + 0.2*0)}{(0.01 + 0.5 + 0.2)} = 0.704$$

4. Hitung stage value

   $Stage = \ln \frac{1-TError}{TError}$

   Dimana ln adalah natural logaritma, gunakan formula LN() pada excel.

5. Update bobot

   $new_{sample_{weight}} = w * e^{stage*perror}$

   - perror = 0 jika y(actual data) == p (prediksi), perror = 1 IF y != p
   - e adalah euler

6. Normalisasi bobot

   $Total\_newsampleweight = \sum_{i=1}^{n} new_{sample_{weight}}(i)$, n = banyaknya data

   $$Normalize\_weight(i) = \frac{new_{sample_{weight}}(i)}{Total\_newsampleweight}$$

7. Bobot yang baru akan digunakan sebagai stump pada iterasi selanjutnya
8. Ulangi Langkah 3-7

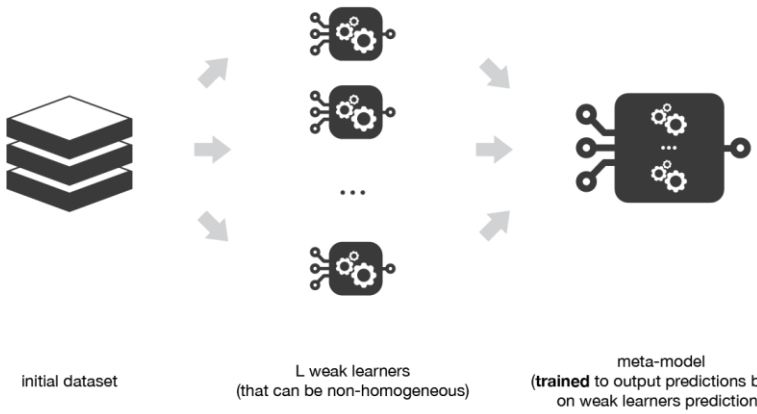
8.5 Stacking

Stacking berbeda dengan boosting dan bagging. Stacking dapat menggunakan algoritma yang berbeda-beda pada data yang sama. Dalam proses pelatihan stacking belajar untuk menggabungkan model dasar menggunakan meta-model. Dapat disimpulkan bahwa ide dasar dari stacking adalah mempelajari beberapa weak-learner yang berbeda dan menggabungkannya dengan melatih meta-mode untuk menghasilkan prediksi berdasarkan beberapa prediksi yang dikembalikan oleh meta-model. Jadi, dua hal yang perlu didefinisakan untuk membangun model adalah: L weak-learner yang akan dilatih dan meta-model yang menggabungkannya.

Sebagai contoh dalam permasalahan klasifikasi dapat dipilih wak-learner (L) adalah KNN classifier, logistic regression, dan SVM. Dan kemudian dipilih 1 model sebagai meta-model, misal ANN. Sehingga keluaran dari 3 buah weak-learner akan menjadi masukan dari ANN dan akan menghasilkan 1buah keputusan final.

Tahapan Stacking :

1. Pisahkan data pelatihan menjadi dua bagian sebagai bagian1 dan bagian2. Dapat juga menggunakan Teknik k-fold cross-validation.
2. Pilih L weak-learner (missal KNN classifier, logistic regression, dan SVM) dan lakukan pelatihan terhadap data bagian1 dengan masing-masing algoritma.
3. Untuk setiap L weak-learner lakukan prediksi mengunakan data bagian2.
4. Lakukan pelatihan pada data bagian2 dengan meta-model dengan menggunakan hasil prediksi yang dibuat oleh L weak-learner sebagai input.



Gambar 8. 4 Ilustrasi Stacking