Supervised Learning

Selasa, 14 April 2020 18.55

Supervised learning adalah salah satu cabang dari machine learning dimana mesin diberikan fakta acuan atau contoh nilai value output yang diharapkan dalam proses pembelajaran mesin.Sehingga tujuan dari supervised learning adalah untuk membuat fungsi yang jika diberikan data dan contoh output yang diinginkan dapat memperkirakan hubungan antara input dan output data yang diobservasi untuk dapat direplikasi hasilnya di data serupa.

KNN = K nearest neighbor
Untuk klasifikasi berdasarkan jarak euclidian antar titik
Penggunaan :
Library(class)
Var1 <-Knn(training_data,testing_data,training_labels, k = nilai k) #membuat model knn
Mean(var_1 == test_labels) #mengetahui akurasi model

Pada knn nilai k berarti jumlah titik data atau neighbor yang diperhitungkan untuk klasifikasi. Default nilai k pada fungsi knn() adalah 1, artinya hanya neighbor terdekat dan termirip yang diperhitungkan untuk mengklasifikasi data unlabel. Klasifikasi ditentukan berdasarkan jumlah neighbor terbanyak. Jika seri maka akan dirandom. Nilai k lebih besar tidak berarti lebih baik, karena classifier lebih peka terhadap perbedaan label dalam neighborhood yang kecil. Namun nilai k besar membuat model lebih resisten terhadap data- data yng kemungkinan bersifat noises. Rule of thumbnya adalah nilai k adalah akar dari jumlah observasi. Tapi opsi terbaik menenetukan nilai k adalah dengan mengetes beberapa nilai lalu dibandingkan hasilnya

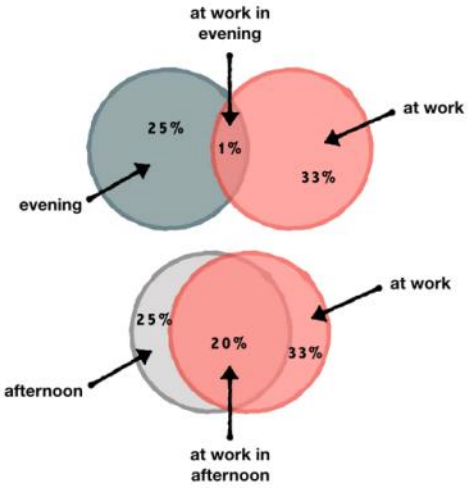
Sebelum menggunakan knn data yang dimiliki perlu dipersiapkan terlebih dahulu. Ada atribut yang bisa dikuantifikasi dan ada yang tidak(contoh : bentuk). Karena knn menggunakan jarak euclidian maka semua atribut perlu dikuantifikasi. Atribut yang gabisa dikuantifikasi dapat menggunakan dummy coding. Dummy coding menempatkan nilai biner 1 jika atribut tsb. Termasuk dengan suatu label dan 0 jika tidak.

Dari semua atribut yang sudah dikuantifikasi tadi masih perlu dinormalisisasi karena tidak semua atribut memiliki range nilai yang sama.

```
normalize <- function(df){
        Return((df - min(df) /max(df) - min(df)))}
```

Naïve Bayes
Bayesian method pada dasarnya menggunakan informasi kemungkinan kejadian2 tunggal untuk menghitung kemungkinan sebuah kejadian yang beririsan. Kejadian yang tidak memengrauhi atau memberikan info ttg kejadian lainnya disebuat independent event. Sebaliknya disebut dependent event.Probability sebuah kejadian dapat diperkirakan dari probability event-event yang beririsan dan mempengaruhi kejadian tsb. Naïve bayes cenderung bekerja dgn baik pada permasalahan yang informasinya dari berbagai atribut perlu dipertimbangkan secara simultan dan dievaluasi menyeluruh.Naive bayes lebih cocok digunakan untuk data kategorical

Conditional probability and dependent events



The **conditional probability** of events A and B is denoted P(A | B)

- P(A | B) = P(A and B) / P(B)
- P(work | evening) = 1 / 25 = 4%
- P(work | afternoon) = 20 / 25 = 80%

Menggunakan naïve bayes

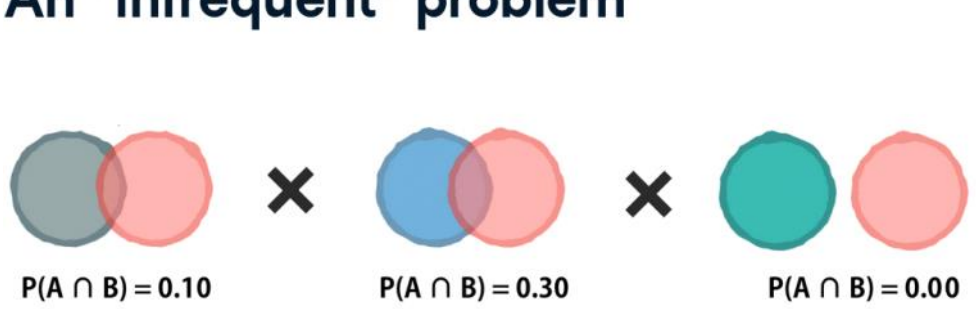
```
Library(naivebayes)
Var1 <- naïve_bayes(event_A ~ event_B, data = df) #membuat model naïve bayes
Predict(var1, data_test) # mendapatkan nilai prediksi
```

Bayesian method
p(A|B) = p(A & B)/p(B)

Keterbatasan dari menggunakan naïve bayes adalah hanya tidak bisa memprediksi berdasarkan banyak event. Sehingga model naïve bayes menggunakan asumsi untuk mensimplifikasi sebagai berikut :

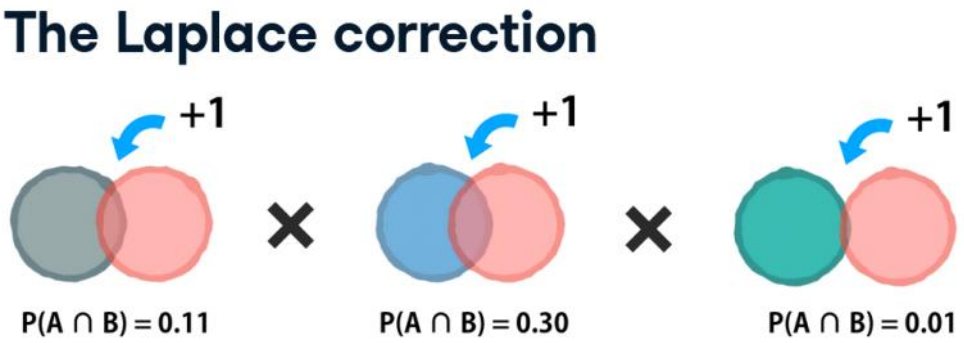
Screen clipping taken: 10/31/2020 11:03 PM

Naïve simplification mengasumsikan bahwa kejadian hitam dan biru adalah indpendent, untuk mendapatkan irisan kejadian merah,biru,dan hitam maka irisan merah dan hitam dikalikan dengan irisan merah dan biru.



Screen clipping taken: 10/31/2020 11:05 PM

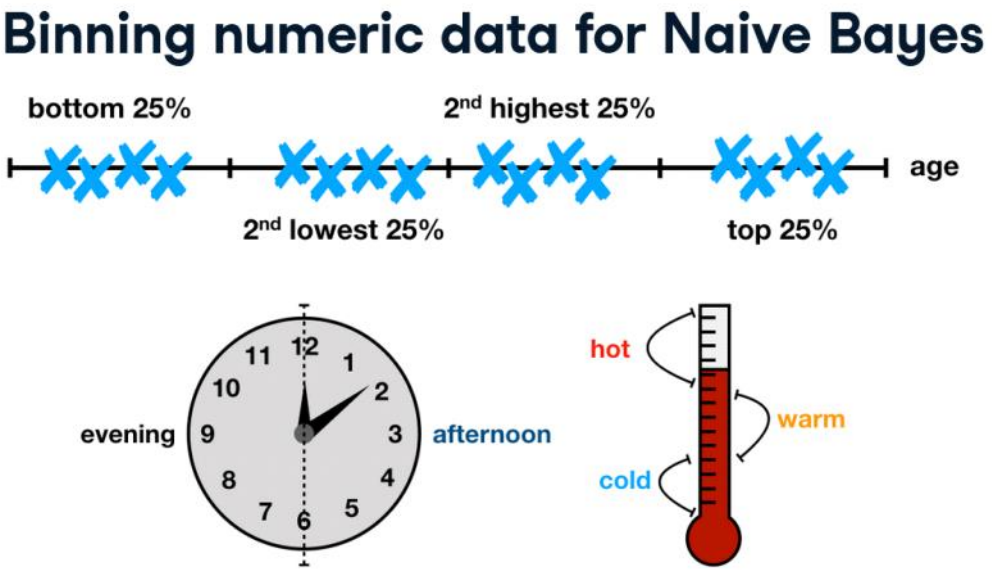
Permasalahan yang timbul dari menggunakan naïve simplification adalah jika ada dua kejadian yang sebelumnya terobservasi belum pernah terjadi namun mungkin akan terjadi di masa depan. Akan tidak ada irisan antar dua kejadian itu dan menimbulkan hasil 0 untuk irisan semua kejadian.



Screen clipping taken: 10/31/2020 11:06 PM

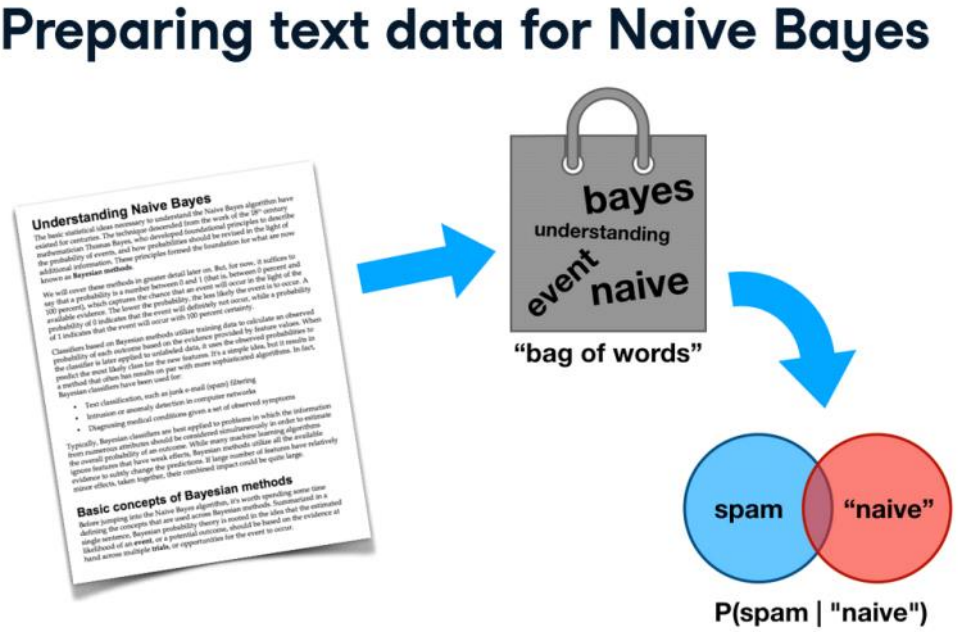
Oleh untuk mengatasi infrequent problem, ditambahkan laplace correction atau laplace estimator. Biasanya angka yang digunakan kecil seperti 1.

Karena lebih sulit bagi naïve bayes untuk memproses data numeris, maka dapat dilakukan binning untuk mengubah data numeris menjadi kategoris. Binning adalah metode membagi sekelompok data numeris ke bin atau kategori yang lebih berarti.



Screen clipping taken: 10/31/2020 11:37 PM

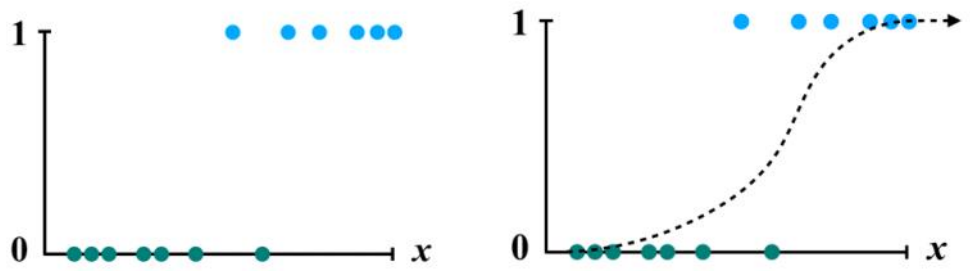
Untuk unstructured data seperti text maka dapat menggunakan model bag of words untuk mendeteksi berapa kemungkinan sebuah kata muncul di suatu dokumen. Bag of words menghasilkan tabel dimana rownya adalah dokumen dan kolomnya adalah kata dari bag of words. Tiap cellnya berisi munculnya sebuah kata di dokumen tsb.



Screen clipping taken: 10/31/2020 11:40 PM

Logistic Regression

Logistic regression adalah salah satu metode regresi yang digunakan untuk klasifikasi. Hal ini dilakukan dengan kurva logistic yang diberi input nilai x berapapun, outputnya akan selalu diantara 0 dan 1 seperti probability.Semakin besar probabilitynya, semakin mungkin output tsb. Dilabeli 1.

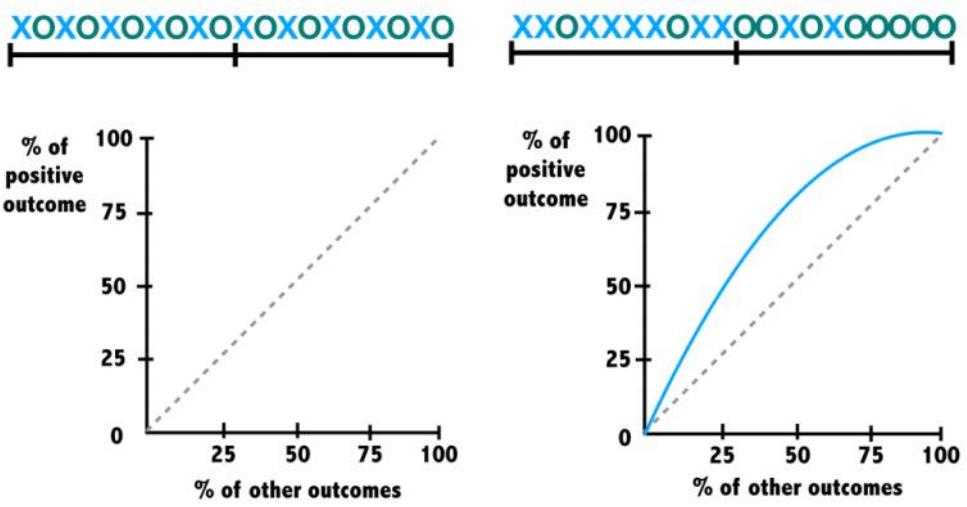


Screen clipping taken: 10/31/2020 11:49 PM

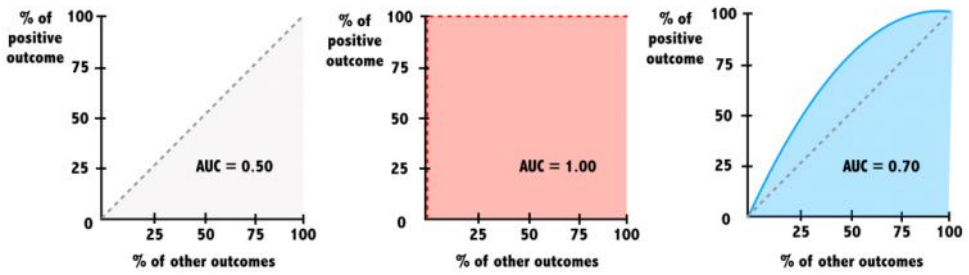
Menggunakan logistic regression

Model <- Glm(y~ x1+x2+x3, data = training_dataset, familiy = "binomial") #membuat model logistic regression
Prob <- predict(model,test_dataset,type = "response") #mendapatkan probability
Pred <- ifelse(prob>0.5, 1,0) #mendapatkan output klasifikasi, threshold bisa disesuaikan

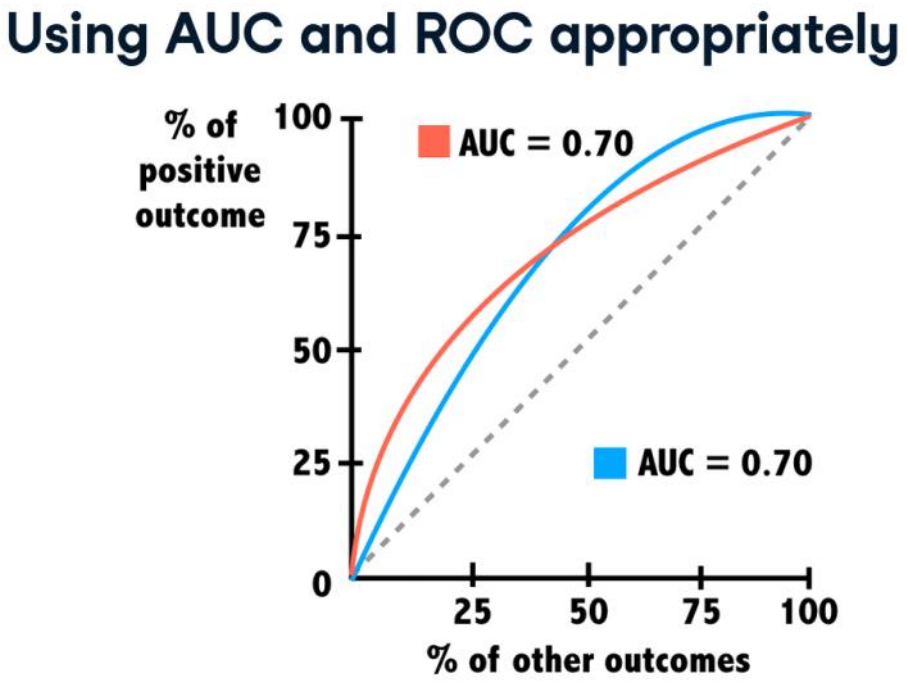
ROC curves
ROC (Receiver Operating Characteristic) curve digunakan untuk memvisualisasikan performa model klasifier.ROC mengukur brp persen outcome yg diinginkan dibandingkan secara kesuluruhan. Garis diagonal adalah baseline preforma model. Semakin jauh model dari garis tsb. Semakin bagus modelnya. Untuk mengkuantifikasi performa model digunakan AUC (Area Under Curve).Semakin mendekati 1 semakin bagus performa modelnya.Selain nilai AUC, bentuk kurvanya juga harus dipertimbangkan untuk memilih model mana yang lebih cocok.



Screen clipping taken: 11/1/2020 12:17 AM



Screen clipping taken: 11/1/2020 12:19 AM



Screen clipping taken: 11/1/2020 12:20 AM

Cara menggambar ROC

```
Library(pROC)
ROC <- roc(actual_data,predicted_data) #membuat kurva ROC
Plot(ROC) #plot kurva ROC
AUC(ROC) #menghitung AUC
```

Tambahan Bacaan:
https://developers.google.com/machine-learning/crash-course