

Machine Learning? Apa Itu?

Machine Learning adalah teknik dimana komputer dapat mengekstraksi atau mempelajari pola dari suatu data, kemudian dengan pola yang telah dipelajari dari data historis, komputer mampu mengenali dan memprediksi trend, hasil atau kejadian di masa mendatang atau dari observasi baru tanpa perlu diprogram secara eksplisit

Sebagai contoh, pernah buka folder spam di email kamu? Di sana ada banyak email promosi dan sebagainya yang dikirim secara acak ke email kita padahal tidak kita inginkan. Lalu bagaimana, perusahaan seperti Google, Yahoo, Hotmail, Microsoft dan perusahaan penyedia lainnya secara otomatis memfilter pesan spam ini? Tentunya dengan machine learning.

Selain mengenali email sebagai spam atau bukan spam ada banyak contoh penggunaan machine learning lainnya, seperti memprediksi harga saham, pengenalan wajah (face recognition), mengenali tulisan tangan, mendeteksi fraud/scam kartu kredit, memprediksi cuaca, dan memprediksi permintaan barang, Machine Learning memungkinkan komputer mengenali pola tanpa diprogram secara eksplisit.

Terminologi Machine Learning

Dalam pembuatan model machine learning tentunya dibutuhkan data. Sekumpulan data yang digunakan dalam machine learning disebut DATASET, yang kemudian dibagi/di-split menjadi training dataset dan test dataset.

TRAINING DATASET digunakan untuk membuat/melatih model machine learning, sedangkan TEST DATASET digunakan untuk menguji performa/akurasi dari model yang telah dilatih/di-training.

Teknik atau pendekatan yang digunakan untuk membangun model disebut ALGORITHM seperti Decision Tree, K-NN, Linear Regression, Random Forest, dsb. dan output atau hasil dari proses melatih algorithm dengan suatu dataset disebut MODEL.

Umumnya dataset disajikan dalam bentuk tabel yang terdiri dari baris dan kolom. Bagian Kolom adalah FEATURE atau VARIABEL data yang dianalisa, sedangkan bagian baris adalah DATA POINT/OBSERVATION/EXAMPLE.

Hal yang menjadi target prediksi atau hal yang akan diprediksi dalam machine learning disebut LABEL/CLASS/TARGET. Dalam statistika/matematika, LABEL/CLASS/TARGET ini dinamakan dengan Dependent Variabel, dan FEATURE adalah Independent Variabel.

Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

- ☐ 1 : Dataset; 2 : Feature; 3 : Training Set; 4 : Test Set
☐ 1 : Data Point; 2 : Feature; 3 : Label; 4 : Dataset
☐ 1 : Feature; 2 : Observation; 3 : Label; 4 : Dataset
☒ 1 : Dataset; 2 : Feature; 3 : Label; 4 : Data Point

Submit Answer

Senja dan Aksara akan membuat suatu model machine learning yang dapat memprediksi apakah customer akan melakukan pembelian setelah mengunjungi beberapa halaman e-commerce. Target adalah 1 jika customer melakukan pembelian dan 0 jika tidak ada pembelian. Berikut, 10 baris pertama dari dataset yang digunakan oleh Senja dan Aksara. Kolom manakah yang dapat digunakan oleh Senja dan Aksara sebagai predictor variable atau feature?

ID	ProductRelated	BounceRates	ExitRates	Weekend	Revenue (Target)
0	1	0.2	0.2	FALSE	0
1	2	0	0.1	FALSE	0
2	1	0.2	0.2	FALSE	0
3	2	0.05	0.14	FALSE	0
4	10	0.02	0.05	TRUE	0
5	19	0.015789474	0.024561404	FALSE	0
6	1	0.2	0.2	FALSE	0
7	1	0.2	0.2	TRUE	0
8	2	0	0.1	FALSE	0
9	3	0	0.022222222	FALSE	0
10	3	0	0.066666667	FALSE	0

- ☐ ID, ProductRelated, BounceRates, ExitRates, Weekend, Revenue
☐ ID, ProductRelated, BounceRates, ExitRates, Weekend
☒ ProductRelated, BounceRates, ExitRates, Weekend
☐ ProductRelated, BounceRates, ExitRates, Weekend, Revenue

Supervised and Unsupervised Learning

Machine Learning itu terbagi menjadi 2 tipe yaitu supervised dan unsupervised Learning. Jika LABEL/CLASS dari dataset sudah diketahui maka dikategorikan sebagai supervised learning, dan jika Label belum diketahui maka dikategorikan sebagai unsupervised learning,

Mengenali email sebagai spam atau bukan spam tergolong sebagai supervised learning, karena kita mengolah dataset yang berisi data point yang telah diberi LABEL "spam" dan "not spam". Sedangkan jika kita ingin mengelompokkan customer ke dalam beberapa segmentasi berdasarkan variabel-variabel seperti pendapatan, umur, hobi, atau jenis pekerjaan, maka tergolong sebagai unsupervised learning,

“Jadi lebih banyak mencoba dan praktik ya untuk tahu yang tepat dan relevannya?”

“Iya, selain itu untuk supervised learning, jika LABEL dari dataset kalian berupa numerik atau kontinu variabel seperti harga, dan jumlah penjualan, kita memilih metode REGRESI dan jika bukan numerik atau diskrit maka digunakan metode KLASIFIKASI. Untuk unsupervised learning, seperti segmentasi customer, kita menggunakan metode CLUSTERING,

Quiz

Jika Senja ingin membuat model Machine Learning untuk mendeteksi transaksi kartu kredit sebagai fraud/scam di suatu e-commerce, Tipe machine learning manakah yang digunakan oleh Aksara untuk membuat model? Dan Jika Aksara ingin membuat segmentasi user dari suatu e-commerce, Tipe machine learning manakah yang tepat digunakan?

- ☐ Senja menggunakan Recommendation System dan Aksara menggunakan Unsupervised Learning.
- ☒ Senja menggunakan Supervised Learning dan Aksara menggunakan Unsupervised Learning.
- ☐ Senja menggunakan Unsupervised Learning dan Aksara menggunakan Supervised Learning.
- ☐ Senja menggunakan Supervised Learning dan Aksara menggunakan Recommendation System.

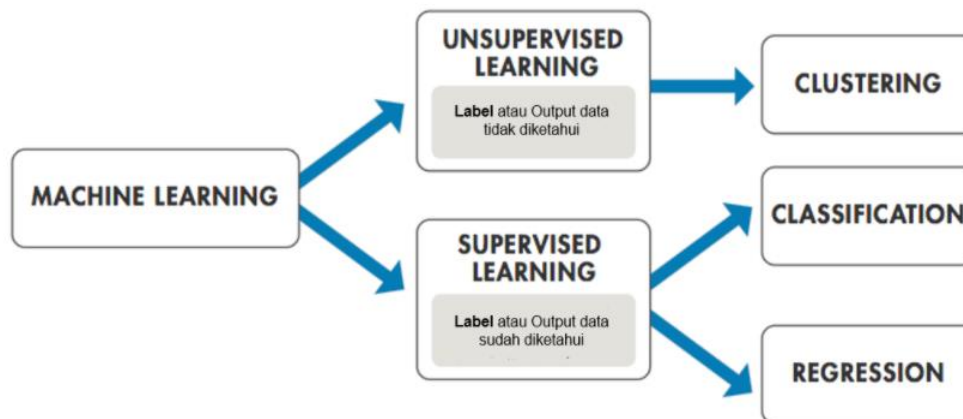
Submit Answer

Misalkan Aksara ingin membuat model Supervised Machine Learning untuk memprediksi apakah suatu email adalah "SPAM" atau "BUKAN SPAM" Manakah dari pernyataan berikut ini yang benar?

- ☐ Semua kolom/feature dari dataset dapat digunakan sebagai LABEL
- ☐ Aksara dapat menggunakan dataset yang tidak memiliki LABEL
- ☒ Aksara tidak dapat menggunakan dataset yang tidak memiliki LABEL "spam" dan "bukan spam"
- ☐ Aksara tidak dapat menggunakan dataset yang memiliki LABEL "spam" dan "bukan spam"

Submit Answer

seperti segmentasi customer, kita menggunakan metode CLUSTERING, salah Senja.



Senja memberikan tantangan kepada Aksara untuk membantu mereka mengembangkan intuisi dalam membedakan yang mana problem klasifikasi dan yang mana problem regresi. Aksara diminta untuk mengidentifikasi dari 4 contoh aplikasi machine learning berikut, manakah yang termasuk permasalahan klasifikasi. Bantulah Aksara untuk menyelesaikan tantangan dari Senja.

- ☒ Menggunakan financial data yang memiliki label untuk memprediksi apakah harga saham akan naik atau turun di minggu depan.
- ☐ Menggunakan data harga rumah yang memiliki label untuk memprediksi harga rumah baru berdasarkan beberapa feature atau variabel.
- ☐ Menggunakan data tidak berlabel untuk mengelompokkan siswa dari suatu e-learning platform ke dalam berbagai kelompok yang berbeda berdasarkan gaya belajar siswa.
- ☐ Menggunakan financial data yang memiliki label untuk memprediksi harga saham di minggu depan.

Submit Answer

ksplorasi Data: Memahami Data dengan Statistik - Part 1

“Selanjutnya saya akan menjelaskan secara singkat tentang tahapan-tahapan dalam pembuatan model machine learning. Membuat model machine learning tidak serta-merta langsung modelling, ada tahapan sebelumnya yang penting untuk dilakukan sehingga kita menghasilkan model yang baik. Untuk penjelasan ini, kalian akan mempraktekkan langsung ya. Kita akan memanfaatkan Pandas library. Pandas cukup powerful untuk digunakan dalam menganalisa, memanipulasi dan membersihkan data. Siap, Aksara?”

Aku mengantuk, siap dengan laptopku.

“Oke, Pertama- tama, kita check dimensi data kita terlebih dahulu. Aksara, silahkan load datanya dan gunakan .shape, .head(), .info(), dan .describe() untuk mengeksplorasi dataset secara berurut. Dataset ini adalah data pembeli online yang mengunjungi website dari suatu e-commerce selama setahun, yaitu 'https://dqlab-dataset.s3-ap-southeast-1.amazonaws.com/pythonTutorial/online_raw.csv',” perintah Senja.

Area teks penjelasan dan jawaban

Quiz

Berdasarkan praktek yang telah dilakukan pada bagian sebelumnya, pernyataan manakah yang tidak sesuai berdasarkan hasil eksplorasi?

- ☐ Jumlah baris dari dataset adalah 12,330
- ☐ Jumlah kolom dari dataset adalah 18
- ☐ Terdapat 10 variabel dengan tipe data float
- ☒ Nilai rata - rata (mean) dari feature BounceRates adalah 0.2

Submit Answer

Eksplorasi Data: Memahami Data dengan Statistik - Part 2

Data eksplorasi tidaklah cukup dengan mengetahui dimensi data dan statistical properties saja, tetapi kita juga perlu sedikit menggali tentang hubungan atau korelasi dari setiap feature, karena beberapa algorithm seperti linear regression dan logistic regression akan menghasilkan model dengan performansi yang buruk jika kita menggunakan feature/variabel saling dependensi atau berkorelasi kuat (multicollinearity). Jadi, jika kita sudah tahu bahwa data kita berkorelasi kuat, kita bisa menggunakan algorithm lain yang tidak sensitif terhadap hubungan korelasi dari feature/variabel seperti decision tree.”

“Pertanyaan menarik, mengetahui distribusi label sangat penting untuk permasalahan klasifikasi, karena jika distribusi label sangat tidak seimbang (imbalanced class), maka akan sulit bagi model untuk mempelajari pola dari LABEL yang sedikit dan hasilnya bisa misleading. Contohnya, kita memiliki 100 row data, 90 row adalah non fraud dan 10 row adalah fraud. Jika kita menggunakan data ini tanpa melakukan treatment khusus (handling imbalanced class), maka kemungkinan besar model kita akan cenderung mengenali observasi baru sebagai non-fraud, dan hal ini tentunya tidak diinginkan,” jelas Senja panjang lebar.

“Nja, kenapa mengetahui distribusi LABEL dari dataset itu penting?”

“Pertanyaan menarik, mengetahui distribusi label sangat penting untuk permasalahan klasifikasi, karena jika distribusi label sangat tidak seimbang (imbalanced class), maka akan sulit bagi model untuk mempelajari pola dari LABEL yang sedikit dan hasilnya bisa misleading. Contohnya, kita memiliki 100 row data, 90 row adalah non fraud dan 10 row adalah fraud. Jika kita menggunakan data ini tanpa melakukan treatment khusus (handling imbalanced class), maka kemungkinan besar

model kita akan cenderung mengenali observasi baru sebagai non-fraud, dan hal ini tentunya tidak diinginkan,” jelas Senja panjang lebar.

Sekarang coba inspeksi nilai korelasi dari fitur-fitur berikut pada dataset_corr yang telah diberikan sebelumnya

ExitRates dan BounceRates

Revenue dan PageValues

TrafficType dan Weekend

QUIZ

Berdasarkan hasil eksplorasi, pilihlah pernyataan berikut ini yang tidak sesuai dengan hasil eksplorasi adalah :

- ☐ ExitRates dan BounceRates memiliki korelasi yang sangat kuat dengan nilai korelasi 0.91.
- ☒ Distribusi label dari dataset tidak seimbang karena total data point dengan label 1 adalah 10422 dan total data point dengan label 0 adalah 1908.
- ☐ Feature yang memiliki korelasi yang paling kuat dengan Revenue adalah PageValues.
- ☐ Tidak ada nilai korelasi untuk Feature VisitorType dan Month.

Submit Answer

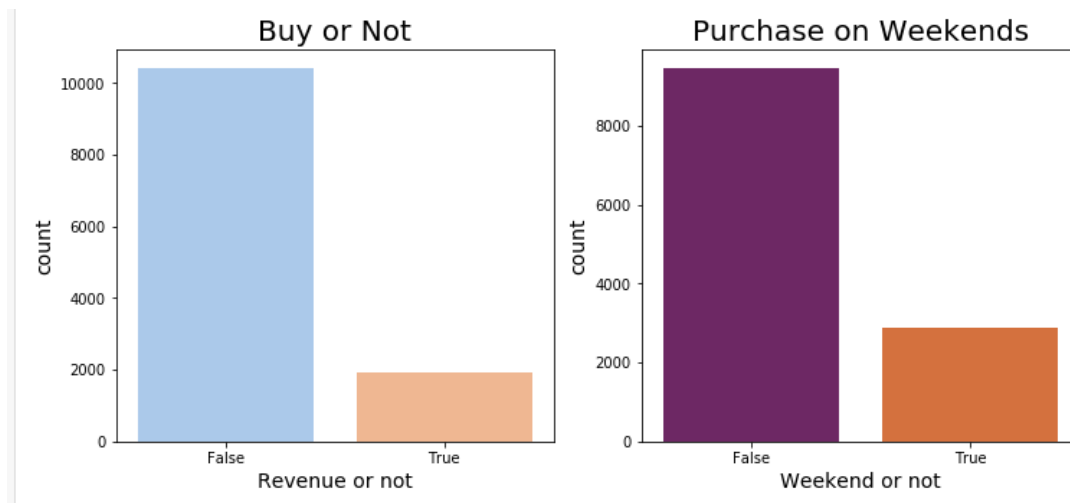
Eksplorasi Data: Memahami Data dengan Visual

“Aksara, satu lagi, dalam mengeksplorasi data, kita perlu untuk memahami data dengan visual.”

Aku tertarik, “Maksudnya?”

“Begini, selain dengan statistik, kita juga bisa melakukan eksplorasi data dalam bentuk visual. Dengan visualisasi kita dapat dengan mudah dan cepat dalam memahami data, bahkan dapat memberikan pemahaman yang lebih baik terkait hubungan setiap variabel/ features.

Misalnya kita ingin melihat distribusi label dalam bentuk visual, dan jumlah pembelian saat weekend. Kita dapat memanfaatkan matplotlib library untuk membuat chart yang menampilkan perbandingan jumlah yang membeli (1) dan tidak membeli (0), serta perbandingan jumlah pembelian saat weekend,

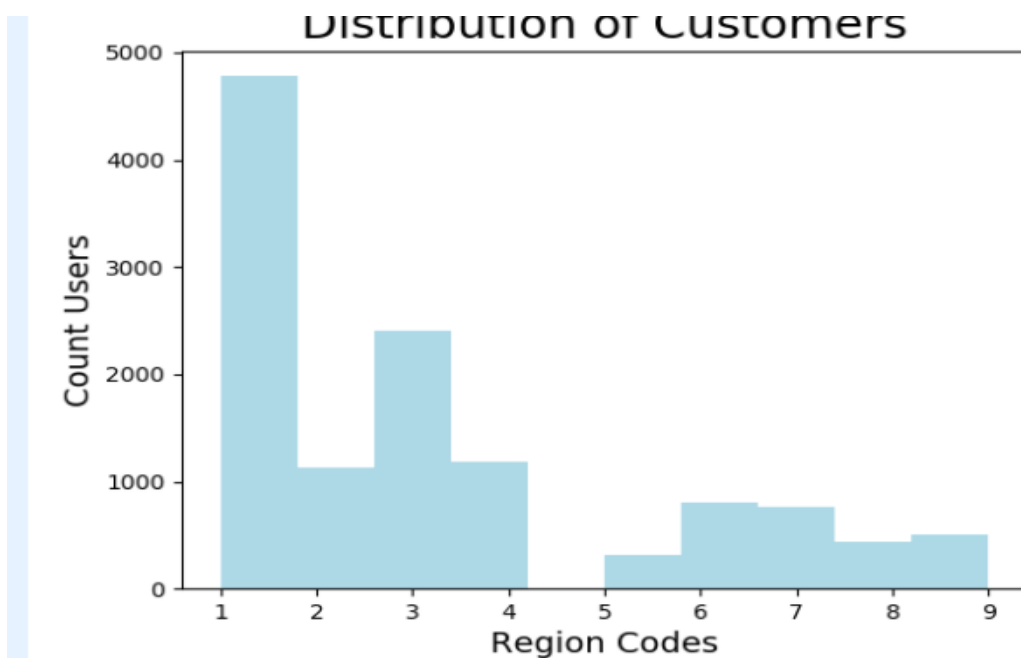


Tugas Praktek

Aku kemudian diminta Senja untuk membuat visualisasi berupa histogram yang menggambarkan jumlah customer untuk setiap Region.

Dalam membuat visualisasi ini aku akan menggunakan `dataset['region']` untuk membuat histogram, dan berikan judul 'Distribution of Customers' pada title, 'Region Codes' sebagai label axis-x dan 'Count Users' sebagai label axis-y.

Berikut tampilan dari histogram yang dimaksud setelah aku menjalankan kode dengan benar.



Data Pre-processing: Handling Missing Value - Part 1

“Kita sedang mengejar materi, Aksara. Semoga materi yang padat ini bisa kamu pahami dengan cepat ya. Kita akan move ke materi lanjutan,” ujar Senja padaku sembari membuka lembaran modul.

Aku ikut memperhatikan. Tampaknya akan ada proyek besar hingga aku harus cepat mempelajari soal machine learning ini.

“Setelah kita melakukan eksplorasi data, kita akan melanjutkan ke tahap data pre-processing. Seperti yang saya jelaskan sebelumnya, raw data kita belum tentu bisa langsung digunakan untuk pemodelan. Jika kita memiliki banyak missing value, maka akan mengurangi performansi model dan juga beberapa algorithm machine learning tidak dapat memproses data dengan missing value. Oleh karena itu, kita perlu mengecek apakah terdapat missing value dalam data atau tidak. Jika tidak, maka kita tidak perlu melakukan apa-apa dan bisa melanjutkan ke tahap berikutnya. Jika ada, maka kita perlu melakukan treatment khusus untuk missing value ini,” jelas Senja.

“Caranya bagaimana?”

“Pengecekan missing value dapat dilakukan dengan code berikut,” ujar Senja sambil mengarahkan layar laptopnya padaku.

Melalui layar dapat ku lihat Senja menggunakan metod `.isnull` pada dataset dan kemudian chaining-nya dengan method `sum`. Untuk jumlah keseluruhan missing value digunakan chaining method `sum` sekali lagi.

Data Pre-processing: Handling Missing Value - Part 2

“Wah, ternyata ada missing value di dataset kita. Apakah data point-nya bisa dihapus saja?” tanyaku.

“Ada beberapa metode yang dapat kita lakukan untuk menangani missing value. Pilihanmu tepat, Aksara, menghapus data adalah salah satunya. Tetapi, metode ini tidak dapat serta merta diimplementasikan. Kita juga perlu menganalisis penyebaran missing value, dan berapa persen

jumlah missing value dalam data kita,” jawab Senja lengkap. Tapi ada beberapa bagian yang masih membingungkan buatku.

“Aku masih agak bingung terutama penerapan metodenya,” sahutku jujur.

“Metode ini dapat diterapkan jika tidak banyak missing value dalam data, sehingga walaupun data point ini dihapus, kita masih memiliki sejumlah data yang cukup untuk melatih model Machine Learning. Tetapi jika kita memiliki banyak missing value dan tersebar di setiap variabel, maka metode menghapus missing value tidak dapat digunakan. Kita akan kehilangan sejumlah data yang tentunya mempengaruhi performansi model. Kita bisa menghapus data point yang memiliki missing value dengan fungsi **.dropna()** dari pandas library. Fungsi **dropna()** akan menghapus data point atau baris yang memiliki missing value.”

Data Pre-processing: Handling Missing Value - Part 3

“Kalau tidak dihapus, ada metode lain yang bisa dipakai?”

“Kita bisa menggunakan metode impute missing value, yaitu mengisi record yang hilang ini dengan suatu nilai. Ada berbagai teknik dalam metode imputing, mulai dari yang paling sederhana yaitu mengisi missing value dengan nilai mean, median, modus, atau nilai konstan, sampai teknik paling advance yaitu dengan menggunakan nilai yang diestimasi oleh suatu predictive model. Untuk kasus ini, kita akan menggunakan imputing sederhana yaitu menggunakan nilai rata-rata atau mean,” jelas Senja.

Imputing missing value sangat mudah dilakukan di Python, cukup memanfaatkan fungsi **.fillna()** dan **.mean()** dari Pandas

Data Preprocessing: Scaling

“Setelah berhasil menangani missing value, sekarang mari kita mempelajari tahapan preprocessing selanjutnya. Aksara, tolong tampilkan kembali 5 dataset teratas dan deskripsi statistik dari dataset. Coba perhatikan, rentang nilai dari setiap feature cukup bervariasi. Misalnya, **ProductRelated_Duration** vs **BounceRates**. **ProductRelated_Duration** memiliki rentang nilai mulai dari 0 - 5000; sedangkan **BounceRates** rentang nilainya 0 - 1. Bisa kamu lihat?”

Aku mengangguk. Senja pun melanjutkan,

“Beberapa machine learning seperti K-NN dan gradient descent mengharuskan semua variabel memiliki rentang nilai yang sama, karena jika tidak sama, feature dengan rentang nilai terbesar misalnya ProductRelated_Duration otomatis akan menjadi feature yang paling mendominasi dalam proses training/komputasi, sehingga model yang dihasilkan pun akan sangat bias. Oleh karena itu, sebelum memulai training model, kita terlebih dahulu perlu melakukan data rescaling ke dalam rentang 0 dan 1, sehingga semua feature berada dalam rentang nilai tersebut, yaitu nilai $\max = 1$ dan nilai $\min = 0$. Data rescaling ini dengan mudah dapat dilakukan di Python menggunakan **.MinMaxScaler()** dari Scikit-Learn library.”

“Kenapa ke range 0 - 1, tidak menggunakan range yang lain?” tanyaku penasaran.

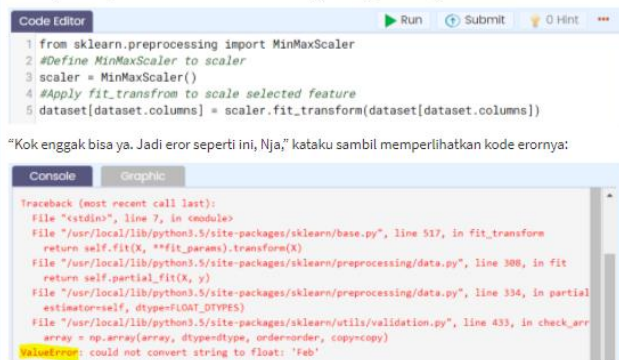
“Karena rumus dari rescaling adalah

$$z_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

dengan rumus ini, nilai max data akan menjadi 1 dan nilai min menjadi 0; dan nilai lainnya berada di rentang keduanya. Rumus ini tidak memungkinkan adanya rentang nilai selain 0 – 1,” ujar Senja sembari menggambarkan rumusnya pada buku catatanku.

Tugas Praktek

“Aksara, silakan praktikkan code untuk data scaling berikut,” pinta Senja.



```
Code Editor
1 from sklearn.preprocessing import MinMaxScaler
2 #Define MinMaxScaler to scaler
3 scaler = MinMaxScaler()
4 #Apply fit_transform to scale selected feature
5 dataset[dataset.columns] = scaler.fit_transform(dataset[dataset.columns])

Run Submit Hint

Kok enggak bisa ya. Jadi eror seperti ini, Nja,” kataku sambil memperlihatkan kode erornya:

Console
Traceback (most recent call last):
  File "<stdin>", line 7, in <module>
  File "/usr/local/lib/python3.5/site-packages/sklearn/base.py", line 517, in fit_transform
    return self.fit(X, **fit_params).transform(X)
  File "/usr/local/lib/python3.5/site-packages/sklearn/preprocessing/data.py", line 308, in fit
    return self.partial_fit(X, y)
  File "/usr/local/lib/python3.5/site-packages/sklearn/preprocessing/data.py", line 334, in partial_fit
    estimator=self, dtype=FLOAT_DTYPES)
  File "/usr/local/lib/python3.5/site-packages/sklearn/utils/validation.py", line 433, in check_array
    array = np.array(array, dtype=dtype, order=order, copy=copy)
ValueError: could not convert string to float: 'Feb'
```

“Ya, code diatas merupakan basic code untuk proses scaling dengan asumsi bahwa semua feature adalah numerik. Tetapi, ketika menjalankan code tersebut untuk dataset `online_raw`, pasti akan terjadi error. Proses scaling hanya bisa dilakukan untuk feature dengan tipe numerik, sedangkan dalam dataset `online_raw`, terdapat feature dengan tipe string atau karakter dan categorical, seperti `Month`, `VisitorType`, `Region`. Oleh karena itu, kita tidak dapat langsung menggunakan code di atas, tetapi kita perlu terlebih dahulu menyeleksi feature - feature dari dataset yang bertipe numerik.”

Senja pun membagikan catatan berisi langkah - langkah untuk proses scaling dengan dataset yang memiliki feature dengan tipe data yang berbeda:

1. Import **MinMaxScaler** dari **sklearn.preprocessing**
2. Deklarasikan fungsi **MinMaxScaler()** ke dalam variabel **scaler**
3. List semua feature yang akan di-scaling dan beri nama **scaling_column** yaitu :
`['Administrative', 'Administrative_Duration', 'Informational', 'Informational_Duration', 'ProductRelated', 'ProductRelated_Duration', 'BounceRates', 'ExitRates', 'PageValues']`
4. Berdasarkan contoh code yang dipraktekkan oleh Aksara, ganti **dataset.columns** dengan **scaling_column**.

Data Pre-processing: Konversi string ke numerik

"Aksara, kita memiliki dua kolom yang bertipe object yang dinyatakan dalam tipe data str, yaitu kolom `'Month'` dan `'VisitorType'`. Karena setiap algoritma machine learning bekerja dengan menggunakan nilai numeris, maka kita perlu mengubah kolom dengan tipe pandas object atau str ini ke bertipe numeris. Untuk itu, kita list terlebih dahulu apa saja label unik di kedua kolom ini," jelas Senja. Lalu Senja pun mulai mempraktikkan sambil menunjukkannya padaku.

Label unik kolom `'Month'`:

```
['Feb' 'Mar' 'May' 'Oct' 'June' 'Jul' 'Aug' 'Nov' 'Sep' 'Dec']
```

dan label unik kolom `'VisitorType'`:

```
['Returning_Visitor' 'New_Visitor' 'Other']
```

"Ok, kita dapat menggunakan LabelEncoder dari sklearn.preprocessing untuk merubah kedua kolom ini seperti ini," Senja pun menjawab sambil praktik dengan kode yang diketikkan sebagai berikut

Pendahuluan

"Akhirnya, setelah data eksplorasi dan preprocessing, datasetnya sudah siap untuk digunakan dalam proses modelling. Kalau dipikir-pikir, preprocessing ini panjang juga dan ribet, aku lebih senang langsung modelling aja," komentarku. "Sebenarnya tidak panjang, ini karena kamu masih tahap belajar sehingga kamu perlu mengerti konsepnya dan tidak asal membuat model, tetapi ketika kamu sudah mulai implementasi proses ini akan otomatis dilakukan sehingga tidak terasa panjang lagi. Semangat yah!"

"Sip, masih semangat kok. Jadi selanjutnya apa?"

"Oke, saya lanjutkan. Pertama-tama saya akan mengenalkan kamu pada library Scikit - Learn. Scikit-learn adalah library untuk machine learning bagi para pengguna python yang memungkinkan kita melakukan berbagai pekerjaan dalam Data Science, seperti regresi (regression), klasifikasi (classification), pengelompokkan/penggugusan (clustering), data preprocessing, dimensionality reduction, dan model selection (pembandingan, validasi, dan pemilihan parameter maupun model)."

Sementara Senja menjelaskan, aku membuka halaman demi halaman modul:

Ada beberapa library machine learning di Python seperti Keras, tetapi Scikit - Learn adalah yang paling basic sehingga jika kita menguasai scikit-learn, kita dapat dengan mudah mempelajari library machine learning yang lain.

Features & Label

Dalam dataset user online purchase, label target sudah diketahui, yaitu kolom Revenue yang bernilai 1 untuk user yang membeli dan 0 untuk yang tidak membeli, sehingga pemodelan yang dilakukan ini adalah klasifikasi. Nah, untuk melatih dataset menggunakan Scikit-Learn library,

dataset perlu dipisahkan ke dalam Features dan Label/Target. Variabel Feature akan terdiri dari variabel yang dideklarasikan sebagai X dan [Revenue] adalah variabel Target yang dideklarasikan sebagai y. Gunakan fungsi drop() untuk menghapus kolom [Revenue] dari dataset.

“Sudah baca petunjuknya, Aksara?”

“Sudah, ini bisa langsung aku coba jalankan?” tanyaku menunggu lampu hijau dari Senja.

“Silakan.”

Aku pun mulai bekerja untuk mengubah dataset ke dalam format yang diminta oleh Scikit - Learn sesuai arahan Senja dengan kode berikut:

Training dan Test Dataset

“Well done, Aksara! Nah, sebelum kita melatih model dengan suatu algorithm machine , seperti yang saya jelaskan sebelumnya, dataset perlu kita bagi ke dalam training dataset dan test dataset dengan perbandingan 80:20. 80% digunakan untuk training dan 20% untuk proses testing.”

Aku kembali menyimak.

“Perbandingan lain yang biasanya digunakan adalah 75:25. Hal penting yang perlu diketahui adalah scikit-learn tidak dapat memproses dataframe dan hanya mengakomodasi format data tipe Array. Tetapi kalian tidak perlu khawatir, fungsi train_test_split() dari Scikit-Learn, otomatis mengubah dataset dari dataframe ke dalam format array. Apakah kamu paham. Aksara? Atau ada pertanyaan?”

“Kenapa perlu ada Training dan Testing, Nja?”

“Fungsi Training adalah melatih model untuk mengenali pola dalam data, sedangkan testing berfungsi untuk memastikan bahwa model yang telah dilatih tersebut mampu dengan baik

memprediksi label dari new observation dan belum dipelajari oleh model sebelumnya. Lebih baik kita praktik saja ya, tampaknya kalau praktik kamu lebih paham.”

Senja menarik laptopnya, sepertinya sedang membuat dokumen untuk praktik. Dan, benar saja...

“Aksara silahkan bagi dataset ke dalam Training dan Testing dengan melanjutkan coding yang sudah kukerjakan ini. Gunakan `test_size = 0.2` dan tambahkan argumen `random_state = 0`, pada fungsi `train_test_split()`. Dicoba saja dulu yah, saya yakin kamu bisa.”

Training Model: Fit

“Good Job, Aksara! Sekarang saatnya kita melatih model atau training. Dengan Scikit-Learn, proses ini menjadi sangat sederhana. Kita cukup memanggil nama algorithm yang akan kita gunakan, biasanya disebut classifier untuk problem klasifikasi, dan regressor untuk problem regresi.”

Aku selalu suka ketika Senja mengapresiasi sesederhana apapun itu, karena selalu berhasil mendorong semangat belajarku. Aku jadi lebih berani untuk mencoba dan menyimak hal baru. Dan satu lagi, Senja selalu mau aku repotkan dengan meminta contoh.

“Boleh kasih contohnya, Nja?”

“Begini, sebagai contoh, kita akan menggunakan Decision Tree. Kita hanya perlu memanggil fungsi `DecisionTreeClassifier()` yang kita namakan “model”. Kemudian menggunakan fungsi `.fit()` dan `X_train, y_train` untuk melatih classifier tersebut dengan training dataset, seperti ini:”

Training Model: Predict

“Yang tadi sudah cukup saya rasa. Setelah model/classifier terbentuk, selanjutnya kita menggunakan model ini untuk memprediksi LABEL dari testing dataset (`X_test`), menggunakan fungsi `.predict()`. Fungsi ini akan mengembalikan hasil prediksi untuk setiap data point dari `X_test` dalam bentuk array. Proses ini kita kenal dengan TESTING,” sambung Senja.

Benar-benar membutuhkan konsentrasi penuh untuk materi modul ini. Aku pun melanjutkan proses testing menggunakan fungsi `.predict()` seperti ini:

valuasi Model Performance - Part 1

Aku menelusuri ulang susunan kodeku, aku merasa ini sudah lengkap dan siap. “Nja, ini semua sudah selesai menurutku, ada tahap akhir khusus kah?”

“Tentu saja. sekarang kita melanjutkan di tahap terakhir dari modelling yaitu evaluasi hasil model. Untuk evaluasi model performance, setiap algorithm mempunyai metrik yang berbeda-beda. Sekarang saya akan menjelaskan sedikit metrik apa saja yang umumnya digunakan. Metrik paling sederhana untuk mengecek performansi model adalah accuracy.”

“Gimana caranya?” tanyaku bingung karena awalnya kupikir ini sudah tuntas.

“Kita bisa munculkan dengan fungsi `.score()`. Tetapi, di banyak real problem, accuracy saja tidaklah cukup. Metode lain yang digunakan adalah dengan Confusion Matrix. Confusion Matrix merepresentasikan perbandingan prediksi dan real LABEL dari test dataset yang dihasilkan oleh algoritma ML,” tukas Senja sambil membuka template dari confusion Matrix untukku:

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

True Positive (TP): Jika user diprediksi (Positif) membeli (`[Revenue] = 1`), dan memang benar(True) membeli.

True Negative (TN): Jika user diprediksi tidak (Negatif) membeli dan aktualnya user tersebut memang (True) membeli.

False Positive (FP): Jika user diprediksi Positif membeli, tetapi ternyata tidak membeli (False).

False Negatif (FN): Jika user diprediksi tidak membeli (Negatif), tetapi ternyata sebenarnya membeli.

Pakai Metrik yang Mana?

Jika dataset memiliki jumlah data False Negatif dan False Positif yang seimbang (Symmetric), maka bisa gunakan Accuracy, tetapi jika tidak seimbang, maka sebaiknya menggunakan F1-Score.

Dalam suatu problem, jika lebih memilih False Positif lebih baik terjadi daripada False Negatif, misalnya: Dalam kasus Fraud/Scam, kecenderungan model mendeteksi transaksi sebagai fraud walaupun kenyataannya bukan, dianggap lebih baik, daripada transaksi tersebut tidak terdeteksi sebagai fraud tetapi ternyata fraud. Untuk problem ini sebaiknya menggunakan Recall.

Sebaliknya, jika lebih menginginkan terjadinya True Negatif dan sangat tidak menginginkan terjadinya False Positif, sebaiknya menggunakan Precision.

Contohnya adalah pada kasus klasifikasi email SPAM atau tidak. Banyak orang lebih memilih jika email yang sebenarnya SPAM namun diprediksi tidak SPAM (sehingga tetap ada pada kotak masuk email kita), daripada email yang sebenarnya bukan SPAM tapi diprediksi SPAM (sehingga tidak ada pada kotak masuk email).

Akalia, ada tantangan buatmu. Ini ada data classification report dari SKIT - Learn. Kamu punya ide bagaimana untuk menginterpretasikan report ini?

```

The default value of n_estimators will change from 10 in version 0.20 to 100 in 0.22.

Training Accuracy : 0.9924690070675473
Testing Accuracy : 0.8834820221681535

```

	precision	recall	f1-score	support
0	0.90	0.97	0.93	3077
1	0.75	0.46	0.57	622
accuracy			0.88	3699
macro avg	0.82	0.72	0.75	3699
weighted avg	0.87	0.88	0.87	3699

Aku termenung sesaat, cukup bingung. Senja pun tersenyum dan memberiku tiga pilihan pernyataan untuk menyelesaikannya. Aku sedang berpikir mana diantaranya yang paling tepat:

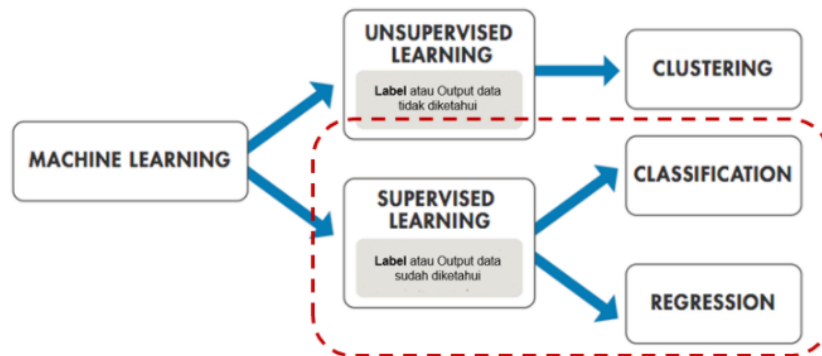
- ☐ Model dapat memprediksi customer yang akan berbelanja dengan baik terlihat dari accuracy yang tinggi yaitu 0.88
- ☐ Nilai precision yang cukup baik menunjukkan bahwa model cenderung untuk memprediksi bahwa customer akan membeli suatu produk, tetapi ternyata tidak membeli.
- ☒ Nilai recall dan F1-score yang kecil untuk class 1 (membeli) dan cukup tinggi untuk class 0 (tidak membeli) menunjukkan bahwa model cenderung salah mengklasifikasikan customer sebagai tidak membeli.

[Submit Answer](#)

Pendahuluan

Setelah pemahaman dengan prosedur machine learning modelling. Selanjutnya materi akan membahas mengenai machine learning algorithm.

Sebagai dasar, akan dipelajari beberapa algorithm machine learning yaitu Logistic Regression, dan Decision Tree untuk classification problem, dan Linear regression untuk regression problem.



Logistic Regression merupakan salah satu algoritma klasifikasi dasar yang cukup populer. Secara sederhana, Logistic regression hampir serupa dengan linear regression tetapi linear regression digunakan untuk Label atau Target Variable yang berupa numerik atau continuous value, sedangkan Logistic regression digunakan untuk Label atau Target yang berupa **categorical/discrete value**.

Contoh continuous value adalah harga rumah, harga saham, suhu, dsb; dan contoh dari categorical value adalah prediksi SPAM or NOT SPAM (1 dan 0) atau prediksi customer SUBSCRIBE atau UNSUBSCRIBED (1 dan 0).

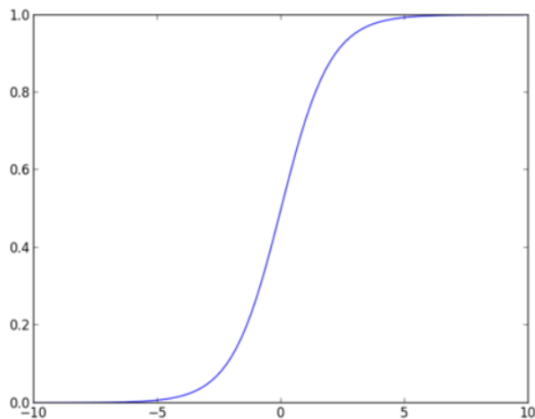
Umumnya Logistic Regression dipakai untuk binary classification (1/0; Yes/No; True/False) problem, tetapi beberapa data scientist juga menggunakannya untuk multiclass classification problem. Logistic regression adalah salah satu linear classifier, oleh karena itu, Logistik regression juga menggunakan rumus atau fungsi yang sama seperti linear regression yaitu:

$$f(\mathbf{x}) = b_0 + b_1x_1 + \dots + b_rx_r$$

yang disebut Logit, dimana Variabel b_0, b_1, \dots, b_r adalah koefisien regresi, dan x_1, \dots, x_r adalah explanatory variable/variabel input atau feature.

Output dari Logistic Regression adalah 1 atau 0; sehingga real value dari fungsi logit ini perlu ditransfer ke nilai di antara 1 dan 0 dengan menggunakan fungsi sigmoid.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$



Jadi, jika output dari fungsi sigmoid bernilai lebih dari 0.5, maka data point diklasifikasi ke dalam label/class: 1 atau YES; dan kurang dari 0.5, akan diklasifikasikan ke dalam label/class: 0 atau NO.

Logistic Regression hanya dapat mengolah data dengan tipe numerik.

Pada saat preparasi data, pastikan untuk mengecek tipe variabel yang ada dalam dataset dan pastikan semuanya adalah numerik, lakukan data transformasi jika diperlukan.

Pemodelan Permasalahan Klasifikasi dengan Logistic Regression

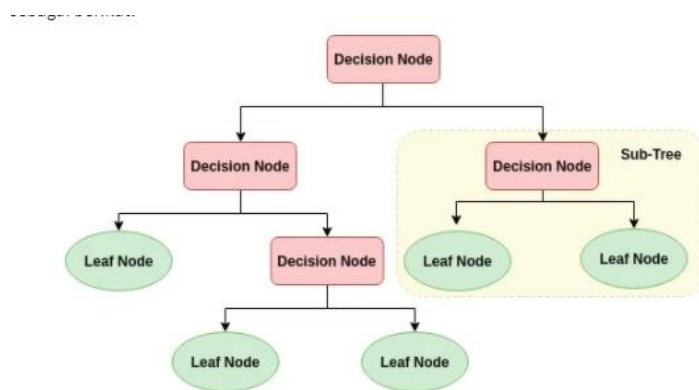
Pemodelan Logistic Regression dengan memanfaatkan Scikit-Learn sangatlah mudah. Dengan menggunakan dataset yang sama yaitu **online_raw**, dan setelah dataset dibagi ke dalam Training

Set dan Test Set, cukup menggunakan modul **linear_model** dari Scikit-learn, dan memanggil fungsi **LogisticRegression()** yang diberi nama **logreg**.

Kemudian, model yang sudah ditraining ini bisa digunakan untuk memprediksi output/label dari test dataset sekaligus mengevaluasi model performance dengan fungsi **score()**, **confusion_matrix()** dan **classification_report()**.

Classification - Decision Tree

Decision Tree merupakan salah satu metode klasifikasi yang populer dan banyak diimplementasikan serta mudah diinterpretasi. *Decision tree* adalah model prediksi dengan struktur pohon atau struktur berhierarki. Decision Tree dapat digunakan untuk classification problem dan regression problem. Secara sederhana, struktur dari decision tree adalah sebagai berikut:

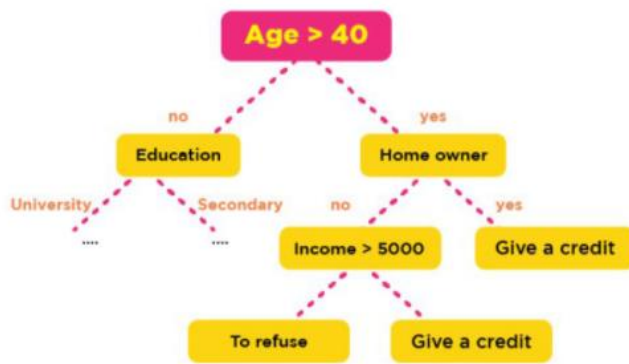


Decision tree terdiri dari :

1. Decision Node yang merupakan feature/input variabel;
2. Branch yang ditunjukkan oleh garis hitam berpanah, yang adalah rule/aturan keputusan, dan
3. Leaf yang merupakan output/hasil.

Decision Node paling atas dalam decision tree dikenal sebagai akar keputusan, atau feature utama yang menjadi asal mula percabangan. Jadi, decision tree membagi data ke dalam kelompok atau kelas berdasarkan feature/variable input, yang dimulai dari node paling atas (akar), dan terus bercabang ke bawah sampai dicapai cabang akhir atau leaf.

Misalnya ingin memprediksi apakah seseorang yang mengajukan aplikasi kredit/pinjaman, layak untuk mendapat pinjaman tersebut atau tidak. Dengan menggunakan decision tree, dapat *break-down* kriteria-kriteria pengajuan pinjaman ke dalam hierarki seperti gambar berikut :

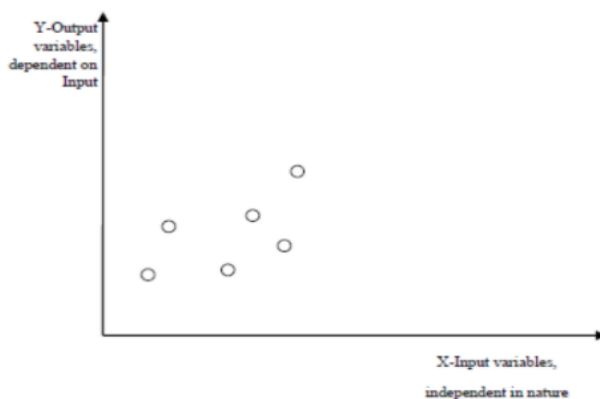


Seumpama, orang yang mengajukan berumur lebih dari 40 tahun, dan memiliki rumah, maka aplikasi kreditnya dapat diluluskan, sedangkan jika tidak, maka perlu dicek penghasilan orang tersebut. Jika kurang dari 5000, maka permohonan kreditnya akan ditolak. Dan jika usia kurang dari 40 tahun, maka selanjutnya dicek jenjang pendidikannya, apakah universitas atau secondary. Nah, percabangan ini masih bisa berlanjut hingga dicapai percabangan akhir/leaf node.

Seperti yang sudah dilakukan dalam prosedur pemodelan machine learning, selanjutnya dapat dengan mudah melakukan pemodelan decision tree dengan menggunakan scikit-learn module, yaitu **DecisionTreeClassifier**.

Regression: Linear Regression - Part 1

Regression merupakan metode statistik dan machine learning yang paling banyak digunakan. Seperti yang dijelaskan sebelumnya, regresi digunakan untuk memprediksi output label yang berbentuk numerik atau continuous value. Dalam proses training, model regresi akan menggunakan variabel input (features) dan variabel output (label) untuk mempelajari bagaimana hubungan/pola dari variabel input dan output.



Model regresi terdiri atas 2 tipe yaitu :

1. Simple regression model → model regresi paling sederhana, hanya terdiri dari satu feature (univariate) dan 1 target.
2. Multiple regression model → sesuai namanya, terdiri dari lebih dari satu feature (multivariate).

Adapun model regresi yang paling umum digunakan adalah Linear Regression.

Regression: Linear Regression - Part 2

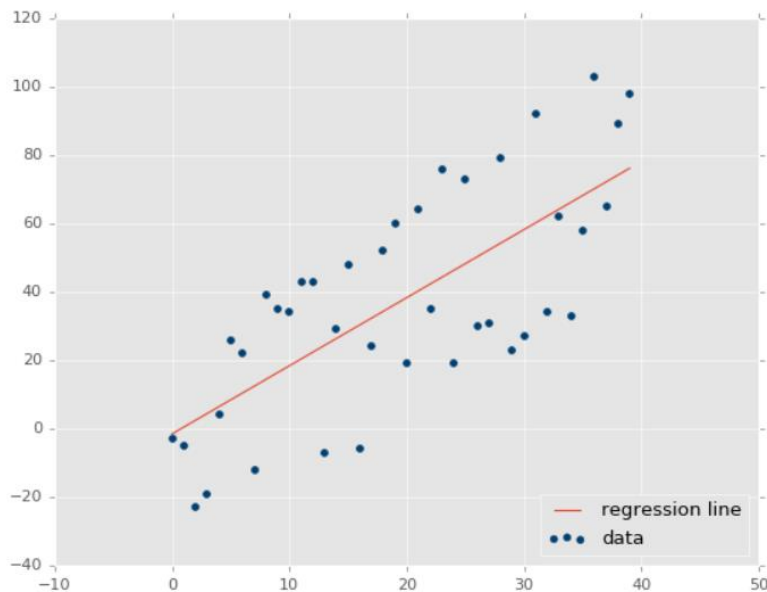
Linear regression digunakan untuk menganalisis hubungan linear antara dependent variabel (feature) dan independent variabel (label). Hubungan linear disini berarti bahwa jika nilai dari independen variabel mengalami perubahan baik itu naik atau turun, maka nilai dari dependen variabel juga mengalami perubahan (naik atau turun). Rumus matematis dari Linear Regression adalah:

$$y = a + bX$$

untuk simple linear regression, atau

$$y = a + b_1X_1 + b_2X_2 + \dots + b_iX_i$$

untuk multiple linear regression dengan, y adalah target/label, X adalah feature, dan a, b adalah model parameter (intercept dan slope).



Perlu diketahui bahwa tidak semua problem dapat diselesaikan dengan linear regression. Untuk pemodelan dengan linear regression, terdapat beberapa asumsi yang harus dipenuhi, yaitu :

1. Terdapat hubungan linear antara variabel input (feature) dan variabel output(label). Untuk melihat hubungan linear feature dan label, dapat menggunakan chart seperti scatter chart. Untuk mengetahui hubungan dari variabel umumnya dilakukan pada tahap eksplorasi data.
2. Tidak ada multicollinearity antara features. Multicollinearity artinya terdapat dependency antara feature, misalnya saja hanya bisa mengetahui nilai feature B jika nilai feature A sudah diketahui.
3. Tidak ada autocorrelation dalam data, contohnya pada time-series data.

Pemodelan Linear regression menggunakan scikit-learn tidaklah sulit. Secara prosedur serupa dengan pemodelan logistic regression. Cukup memanggil **LinearRegression** dengan terlebih dahulu meng-*import* fungsi tersebut :

```
from sklearn.linear_model import LinearRegression
```

“Setelah memahami konsep dasar dari regression, kita akan berlatih membuat model machine learning dengan Linear regression. Untuk pemodelan ini kita akan menggunakan data ‘Boston Housing Dataset’. Setelah pembelajaran kamu sampai di sini, tahu tidak mengapa kita tidak bisa menggunakan data “online purchase”, Aksara?”

Pertanyaan Senja padaku terdengar seperti ujian. Aku berpikir sejenak sebelum menjawab, “Hmm, karena untuk linear regression target/label harus berupa numerik, sedangkan target dari online purchase data adalah categorical. Apakah benar?” jawabku ragu-ragu. Senyum Senja cukup melegakanku.

“Tepat sekali, Senja. Kalau begitu kita bisa lanjut ke pemodelan. Tujuan dari pemodelan ini adalah memprediksi harga rumah di Boston berdasarkan feature - feature yang ada. Asumsikan saja bahwa kita sudah melakukan data eksplorasi dan data pre-processing. Jadi, data yang akan digunakan adalah data yang siap untuk diproses ke tahap pemodelan.”

Regression Performance Evaluation

Aku sudah sampai tahap evaluasi. Sudah sejauh ini tapi ada bagian membingungkan yang kutemukan. Berhubung Senja masih duduk di sebelahku untuk memantau proses kerjaku, aku pun bertanya,

“Kalau mengevaluasi perfoma dari model klasifikasi, aku pakai akurasi dan confusion matrix. Nah, kalau modenya regression, metode evaluasinya bagaimana yah. Nja?”

Senja yang sedang fokus di depan layar laptop akhirnya menoleh oleh sahatanku.

“Untuk model regression, kita menghitung selisih antara nilai aktual (y_{test}) dan nilai prediksi (y_{pred}) yang disebut **error**, adapun beberapa metric yang umum digunakan. Coba kamu ke mari, aku jelaskan langkah-langkahnya.”

Keuntungan bertanya pada Senja adalah ia selalu menyempatkan waktunya untuk menjelaskan secara maksimal! Aku fokus memperhatikan:

Mean Squared Error (MSE) adalah rata-rata dari squared error:

$$\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Root Mean Squared Error (RMSE) adalah akar kuadrat dari MSE:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Mean Absolute Error (MAE) adalah rata-rata dari nilai absolut error:

$$\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Semakin kecil nilai MSE, RMSE, dan MAE, semakin baik pula performansi model regresi. Untuk menghitung nilai MSE, RMSE dan MAE dapat dilakukan dengan menggunakan fungsi `mean_squared_error()`, `mean_absolute_error()` dari `scikit-learn.metrics` dan untuk RMSE sendiri tidak terdapat fungsi khusus di scikit-learn tapi dapat dengan mudah kita hitung

dengan terlebih dahulu menghitung MSE kemudian menggunakan **numpy** module yaitu, **sqrt()** untuk memperoleh nilai akar kuadrat dari MSE.

Pendahuluan

Tak seperti kemarin, hari ini Senja memintaku datang ke ruangannya.

“Silakan duduk. Aksara. Kita bakal full belajar materi baru hari ini, jadi bakal lebih intens. Makanya aku minta kamu ke ruanganku.”

Aku diam menyimak.

“Seperti yang sudah dijelaskan sebelumnya, Machine Learning terdiri atas 2 tipe yaitu supervised dan unsupervised learning. Kita telah banyak membahas tentang supervised learning yaitu Klasifikasi model dan Regression Model. Sekarang kita akan mempelajari dasar- dasar terkait unsupervised learning,” jelas Senja.

Kulihat dengan cepat ia membuka laptop dan menampilkan layar presentasi di depan yang menampilkan rangkuman materi sekaligus slide contoh gambar:

Unsupervised Learning adalah teknik machine learning dimana tidak terdapat label atau output yang digunakan untuk melatih model. Jadi, model dengan sendirinya akan bekerja untuk menemukan pola atau informasi dari dataset yang ada. Metode unsupervised learning yang dikenal dengan clustering. Sesuai dengan namanya, Clustering memproses data dan mengelompokkannya atau mengcluster objek/sample berdasarkan kesamaan antar objek/sampel dalam satu kluster, dan objek/sample ini cukup berbeda dengan objek/sample di kluster yang lain. Contohnya pada gambar berikut:



"Nja, mau tanya. Kita tahu dari mana bentuk polanya?"

"Pada awalnya kita tidak mengetahui bagaimana pola dari objek/sample, termasuk juga tidak mengetahui bagaimana kesamaan maupun perbedaan antara objek yang satu dengan objek yang lain. Setelah dilakukan clustering, baru dapat terlihat bawah objek/sample tersebut dapat dikelompokkan ke dalam 3 kluster. Untuk menjelaskan tentang metode Clustering, kita akan menggunakan metode clustering yang sangat populer, yaitu K-Means Algorithm yang akan kita praktikkan nanti."

K-Means Clustering

"Jadi, Algorithm K-Means itu apa dan bagaimana cara kerjanya?" tanyaku antusias.

"K-Means merupakan tipe clustering dengan centroid based (titik pusat). Artinya kesamaan dari objek/sampel dihitung dari seberapa dekat objek itu dengan centroid atau titik pusat."

Aku masih penasaran. "Jadi, bagaimana kita mengukur kedekatan objek dan centroid?"

"Untuk menghitung kedekatan, digunakan perhitungan jarak antar 2 buah data atau jarak Minkowski. Saya share yah rumusnya," ujar Senja.

Aku menyimak isi rumus yang dibagikan Senja di slide presentasinya:

$$d(x_i, x_j) = (|x_{i1} - x_{j1}|^g + |x_{i2} - x_{j2}|^g + \dots + |x_{ip} - x_{jp}|^g)^{1/g}$$

x_i , x_j adalah dua buah data yang akan dihitung jaraknya, dan p = dimensi/jumlah dari data

Terdapat beberapa tipe perhitungan jarak yang dapat digunakan, yaitu :

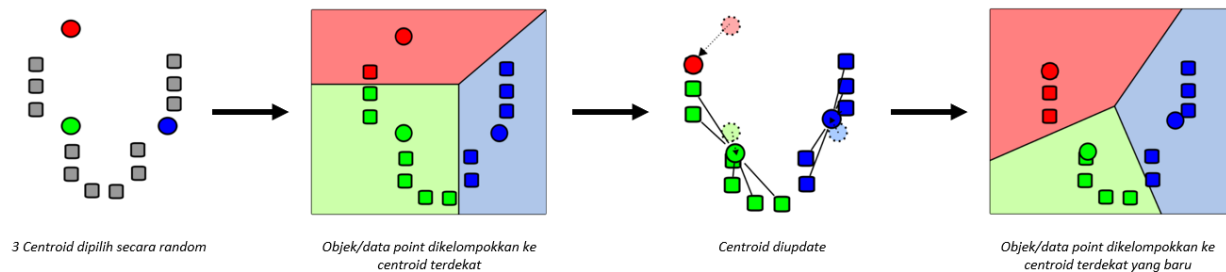
1. Jarak Manhattan di mana $g = 1$
2. Jarak Euclidean di mana $g = 2$
3. Jarak Chebychev di mana $g = \infty$

"Nja, aku masih bingung, cara menentukan centroid bagaimana caranya?"

"Untuk menentukan centroid, pada awalnya kita perlu mendefinisikan jumlah centroid (K) yang diinginkan, semisalnya kita menetapkan jumlah $K = 3$; maka pada awal iterasi, algorithm akan secara random menentukan 3 centroid. Setelah itu, objek/sample/data point yang lain akan dikelompokkan sebagai anggota dari salah satu centroid yang terdekat, sehingga terbentuk 3 cluster data. Sampai sini cukup dipahami?"

“Yup, boleh lanjut, Nja,” sahutku mempersilakan Senja kembali menjelaskan.

“Iterasi selanjutnya, titik-titik centroid diupdate atau berpindah ke titik yang lain, dan jarak dari data point yang lain ke centroid yang baru dihitung kembali, kemudian dikelompokkan kembali berdasarkan jarak terdekat ke centroid yang baru. Iterasi akan terus berlanjut hingga diperoleh cluster dengan error terkecil, dan posisi centroid tidak lagi berubah.”



“Kamu sudah bisa lihat di layar ya, Aksara. Menurutmu, apakah ada perbedaan prosedur antara unsupervised learning dan supervised learning?”

Aku tahu ini pertanyaan untuk menguji pemahamanku.

“Secara prosedur, tahap eksplorasi data untuk memahami karakteristik data, dan tahap preprocessing tetap dilakukan. Tetapi dalam unsupervised learning, kita tidak membagi dataset ke feature dan label; dan juga ke dalam training dan test dataset, karena pada dasarnya kita tidak memiliki informasi mengenai label/target data,” jawabku mantap.

“Tampaknya kamu sudah paham. Saatnya kita mulai praktik membuat programnya.”

Measuring Cluster Criteria

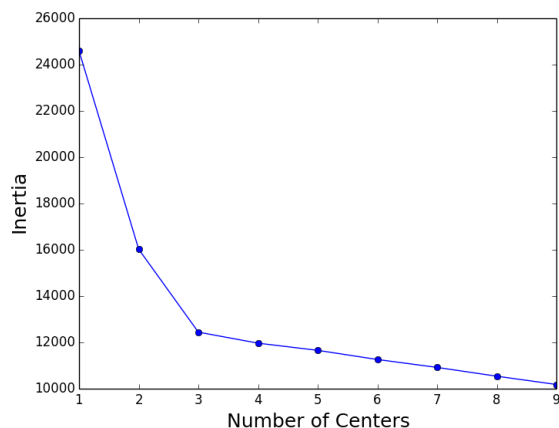
“Segmentasinya udah jadi nih, Nja. Tapi, bagaimana kita tahu bahwa membagi segmentasi ke dalam 5 cluster adalah segmentasi yang paling optimal? Karena jika dilihat pada gambar beberapa data point masih cukup jauh jaraknya dengan centroidnya.”

“Clustering yang baik adalah cluster yang data point-nya saling rapat/sangat berdekatan satu sama lain dan cukup berjauhan dengan objek/data point di cluster yang lain. Jadi, objek dalam satu cluster tidak tersebut berjauhan. Nah, untuk mengukur kualitas dari clustering, kita bisa menggunakan inertia,” jawab Senja langsung.

Aku kembali bertanya karena rasanya masih ada yang janggal. “Memang apa fungsi inertia, Nja?” “Inertia sendiri mengukur seberapa besar penyebaran object/data point data dalam satu cluster, semakin kecil nilai inertia maka semakin baik. Kita tidak perlu bersusah payah menghitung nilai inertia karena secara otomatis, telah dihitung oleh `KMeans()` ketika algorithm di fit ke dataset. Untuk mengecek nilai inertia cukup dengan **print** fungsi `.inertia_` dari model yang sudah di fit ke dataset.”

“Kalau begitu, bagaimana caranya mengetahui nilai K yang paling baik dengan inertia yang paling kecil? Apakah harus trial Error dengan mencoba berbagai jumlah cluster?”

“Benar, kita perlu mencoba beberapa nilai, dan memplot nilai inertia-nya. Semakin banyak cluster maka inertia semakin kecil. Sini deh, saya tunjukkan gambarnya.”



Meskipun suatu clustering dikatakan baik jika memiliki inertia yang kecil tetapi secara praktikal in real life, terlalu banyak cluster juga tidak diinginkan. Adapun rule untuk memilih jumlah cluster yang optimal adalah dengan memilih jumlah cluster yang terletak pada “elbow” dalam inertia plot, yaitu ketika nilai inertia mulai menurun secara perlahan. Jika dilihat pada gambar maka jumlah cluster yang optimal adalah $K = 3$.

Quiz

Berdasarkan inertia plot yang diperoleh, berapakah nilai k yang optimal berdasarkan rule ‘elbow’ method?

- ☐ 4
- ☒ 5
- ☐ 6
- ☐ 7

Submit Answer

enutup/Kesimpulan

Congratulations! Akhirnya selesai satu lagi modul **Machine Learning With Python for Beginner**. Berdasarkan materi-materi yang telah kupelajari dan praktekkan dalam modul ini, aku telah mendapatkan pengetahuan (*knowledge*) dan praktek (*skill*) yang diantaranya

- Memahami apa itu machine learning dengan jenisnya untuk pemodelan
- Memahami dan mampu melakukan Eksplorasi Data & Data Pre-processing
- Memahami dan mampu melakukan proses-proses Pemodelan dengan Scikit-Learn
- Memahami dan mampu melakukan proses-proses pemodelan dengan menggunakan algoritma pada Supervised Learning
- Memahami dan mampu melakukan proses-proses pemodelan dengan menggunakan algoritma pada Unsupervised Learning
- Mengerjakan mini project yang merupakan integrasi keseluruhan materi dan tentunya materi-materi pada modul-modul sebelumnya untuk menyelesaikan persolan bisnis.