DQLab

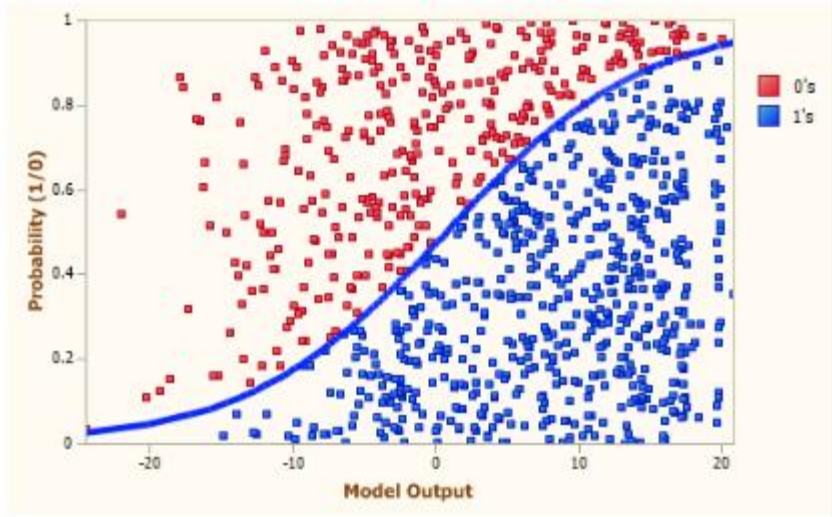
liveClass

Project Implementation

Outline

- Model Selection
- Model Evaluation
- Model Tuning

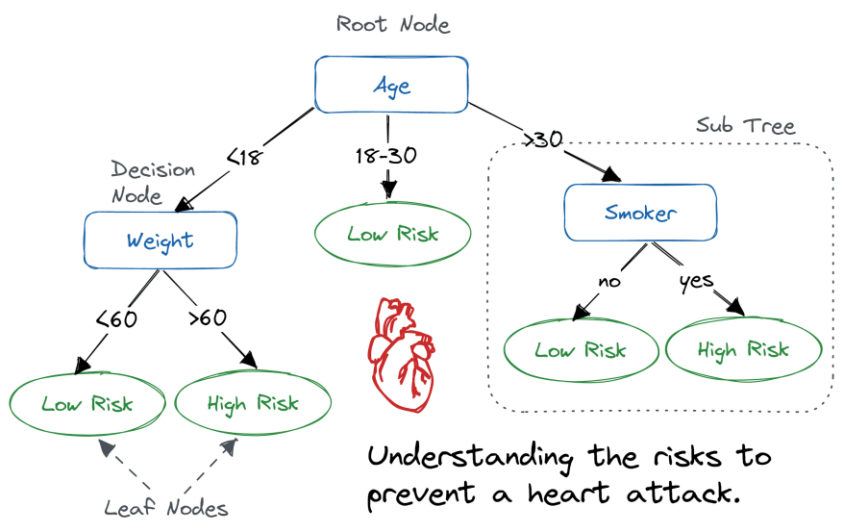
Logistic Regression



Logistic Regression dapat menentukan **decision boundary** dari variabel dependen (target).

1. `clf = LogisticRegression()`
2. `clf.fit(X_train, y_train)` dimana `X_train` – variabel features dan `y_train` – variabel target data training
3. `clf.predict(X_test)` untuk mendapatkan `y_pred` dan kita dapat menghitung probabilistik dengan `clf.predict_proba(X_test)`
4. Mengetahui akurasi (berapa persentase target diidentifikasi benar) dengan `accuracy_score(y_test, y_pred)`

Decision Tree

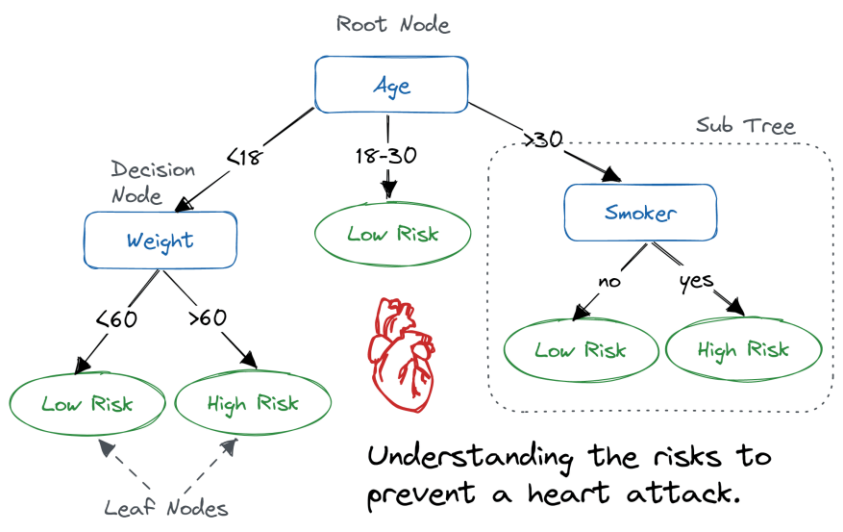


- Node sebagai fitur
- Cabang sebagai keputusan berdasarkan fitur

- Pembagian pertama berdasarkan usia
- Jika direntang 18-30 (Low Risk)
- Jika usia dibawah 18 tahun, terdapat pembagian kembali berdasarkan berat badan
- Terdapat pemahaman heuristic mengapa kejadian potensi jantung terjadi

	Is smoker	Risk
Old Age	Yes	High Risk
Old Age	No	Low Risk

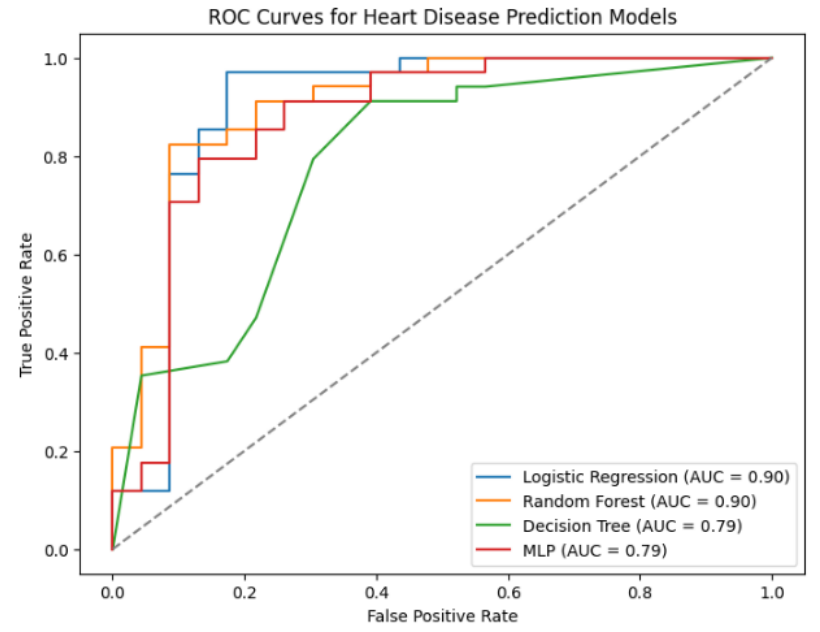
Decision Tree



1. clf = DecisionTreeClassifier()
2. clf.fit(X_train, y_train) dimana X_train – variabel features dan y_train – variabel target data training
3. clf.predict(X_test) untuk mendapatkan y_pred dan kita dapat menghitung probabilistik dengan clf.predict_proba(X_test)
4. Mengetahui akurasi (berapa persentase target diidentifikasi benar) dengan accuracy_score(y_test, y_pred)

- Node sebagai fitur
- Cabang sebagai keputusan berdasarkan fitur

Receiver Operating Characteristics



- TRUE POSITIVE (y-axis) pada sumbu Y #(Classifier predict positive, actually positive)/#(positive)
- FALSE POSITIVE (x-axis) pada sumbu X untuk #(Classifier predict positive, actually negative)/#(negative)
- AUC (Area Under Curve) mengukur keterampilan model (semakin besar, semakin baik model melakukan klasifikasi dua kelas)
- Baseline AUC adalah 0.5 (garis putus-putus)

AUC of ROC Curve

1. Y_score = clf.predict_proba(X_test)
2. fpr, tpr, thresh = roc_curve(Y_test, Y_score[:, 1])
3. roc_auc = auc(fpr, tpr) untuk mendapatkan luasan dimana auc (inputan FP dan TP arrays)
4. Jika model akurat namun hasil prediksinya rendah, ini bisa berarti variabel yang kita gunakan tidak tepat sasaran.

4 Kategori Metriks

	Actual Positives	Actual Negatives
Positive Predictions	True Positives (TP)	False Positives (FP)
Negative Predictions	False Negatives (FN)	True Negatives (TN)

Kategori pertama, TRUE dan FALSE, apakah model benar atau salah.

Kategori kedua, POSITIVE dan NEGATIVE, label target yang diterapkan pada model

Intrepretasi dalam kategori

TRUE POSITIVE: Penyakit jantung tepat ditangani lebih awal

FALSE POSITIVE: Salah penanganan bagi pasien

TRUE NEGATIVE: Pasien bisa pulang dirumah

FALSE NEGATIVE: Pasien merasa aman, padahal potensi penyakit jantung

Gunakan ini untuk mendapatkan
TN, FP, FN, TP (ordered)

```
print(confusion_matrix(y_test, y_pred).ravel())
```

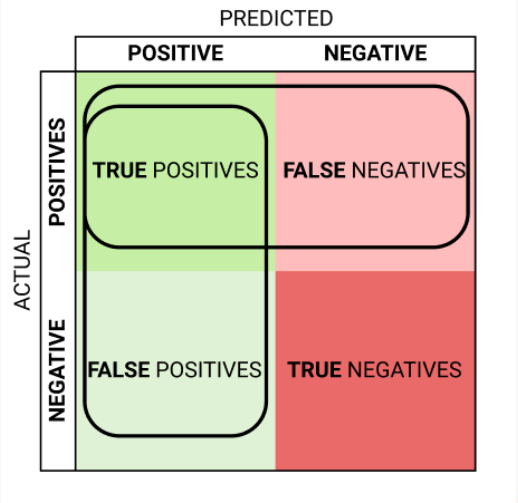
Model Evaluation

PRECISION : Proporsi data dengan total prediksi prediksi penyakit jantung (TP/(TP +FP))

1. Semakin besar, semakin baik orang melakukan perawatan dini

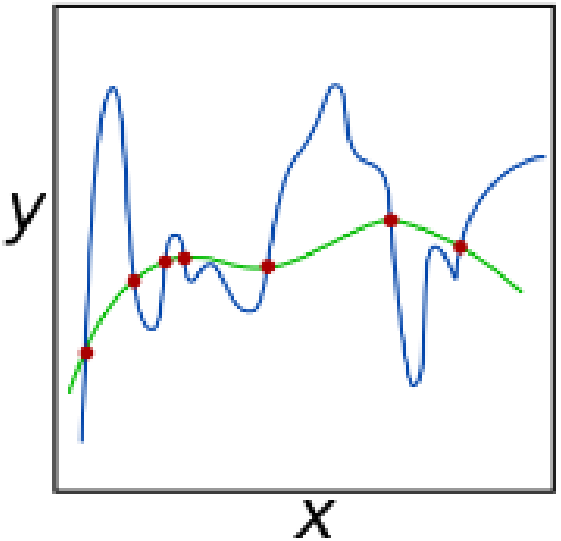
RECALL : Proporsi data dari total penderita penyakit jantung (TP/(TP +FN))

1. Semakin besar, model dapat menargetkan penyakit jantung dengan tepat



Regularization (Tuning Model)

Proses mengatasi overfitting, yaitu ketika sebuah model cenderung mengikuti data pelatihan



Disini kita akan melakukan perubahan parameter koefisien dalam model untuk mencegah model yang kompleks (out of samples)

Disini kita akan meningkatkan performansi metriks

Contoh Regularization (Tuning Model)

LogisticRegression: parameter c sebagai kebalikan kekuatan regularization. Artinya semakin kecil, maka semakin kuat regularisasinya (semakin besar penaltinya). Semakin besar penalti, semakin tidak rumit modelnya.

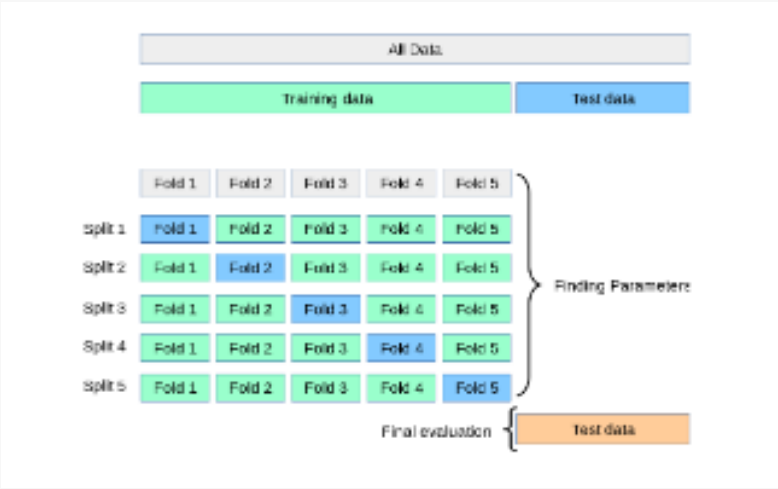
Dari tidak kompleks menjadi kompleks (**C=0.05 < C=0.5 < C=1)**

DecisionTree: parameter max_depth mempengaruhi banyaknya lapisan yang bisa dilalui model. Semakin dalam, semakin kompleks pohon yang dihasilkan.

Dari tidak kompleks menjadi kompleks **(max_depth=3 < max_depth=5 < max_depth=10)**

Cross Validation (Tuning Model)

Memperkirakan kinerja model dengan memisah-misahkan data.



Kita akan membuat hingga K “lipatan” untuk membagi training dan testing

Hyperparamater Tuning

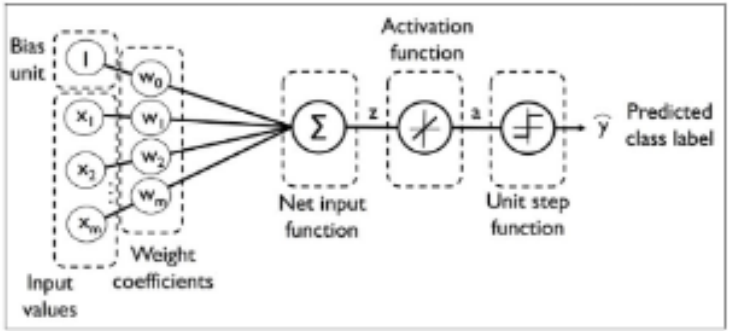
Parameter yang di konfigurasi sebelum training. Parameter yang mana dipelajari model, tidak dapat dikatakan hyperparameter: slope/kemiringan linear regression.

Sedangkan contoh hyperparameter: max_depth, n_estimators

MLP dalam Deep Learning

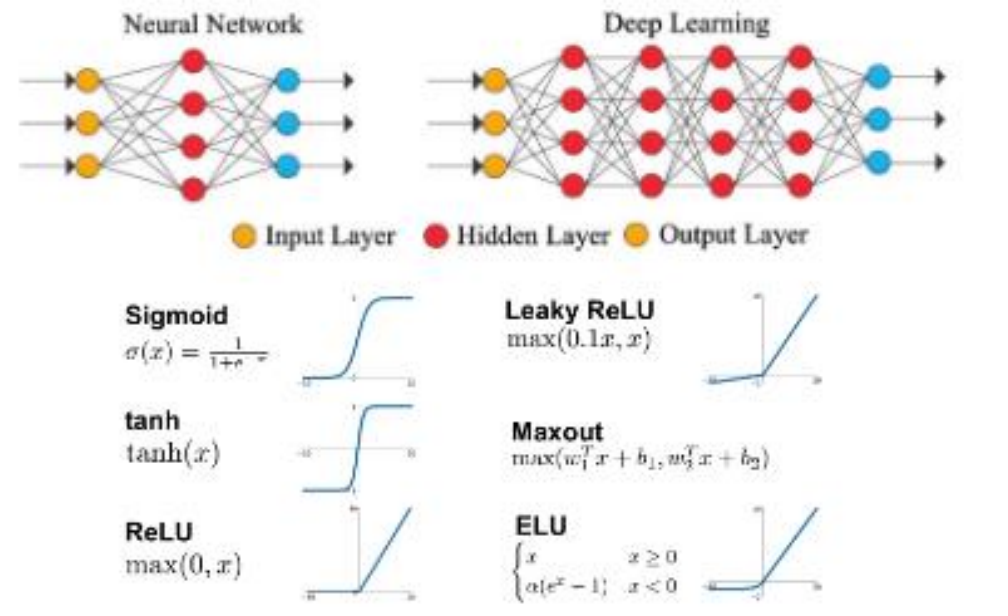
Kita tahu bahwa, deep learning (mencoba meniru otak manusia saat mengajar komputer untuk belajar).

- Blok bangunan dasar (perceptrons)



1. Input dilakukan standardisasi
2. Input dijumlahkan melalui bobot mendapatkan z
3. Nilai Z adalah kombinasi linear dari x
4. Output dilakukan transformasi melalui activation function (non linearitas)
5. Unit step func untuk menyesuaikan dengan output

Hidden Layers dan Activation Func



Catatan dalam menggunakan MLP

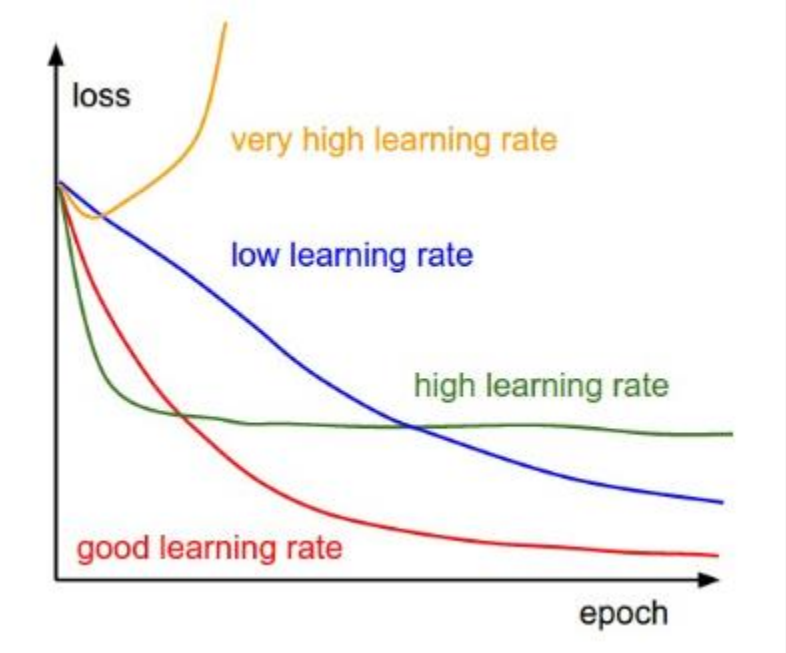
Parameter yang digunakan

1. Activation – Jenis aktivasi
2. Alpha – Konstanta regularisasi
3. Hidden Layer Size – Jumlah lapisan tersembunyi
4. Learning Rate – Seberapa cepat bobot jaringan dari feedback data training
5. Max Iter – Jumlah iterasi

Sebelum menggunakan MLP, perlu mempertimbangkan

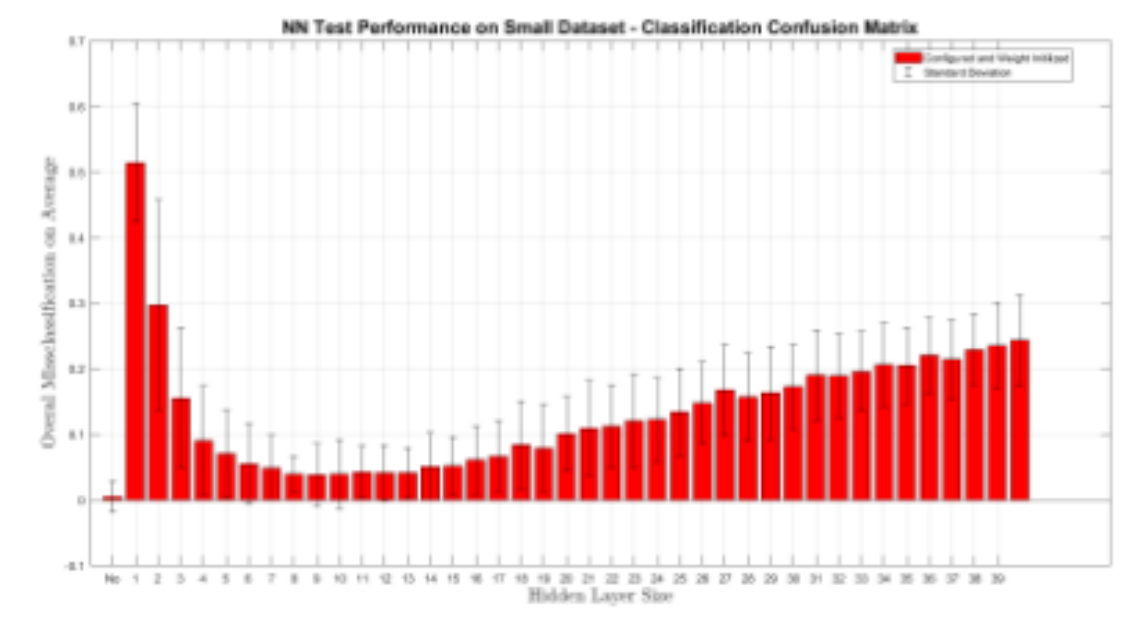
1. Strandarisasai data untuk membantu konvergensi yang lebih baik dari waktu ke waktu
2. Performansi lebih baik dengan banyak data (hanya akan memakan waktu yang panjang)

MLP Tuning (1/2)



1. Weight akan diupdate iteratif dengan back-propagation
2. Learning rate yang baik dia langsung turun dan stabil (warna merah)
3. Learning rate terlalu besar akan mendapatkan loss besar pula (warna kuning)

MLP Tuning (2/2)



Peningkatan performansi pada level komplesitas tertentu, lalu turun setelahnya

MODEL REVIEW

Regresi logistik: linear klasifikasi mengidentifikasi batas keputusan

Decision Tree: klasifikasi berdasarkan cabang

Random Forest: ansambel dari Decision Tree

Neural Networks (MLPs): lapisan (perceptron) menggunakan kombinasi fitur linier dengan fungsi aktivasi nonlinier

MODEL IMPLEMENTATION

Similiarity

1. Fitur transformasi dan regularisasi
2. Penyesuain dengan **classifier.fit(X_train, y_train**)
3. Prediksi dengan **predict_proba()** dan **predict()**

Perbedaan Parameter

1. Decision Trees: **max_depth, min_samples_split**
2. Random Forests: **n_estimators, oob_score**
3. Logistic Regression: **fit_intercept, class_weight**
4. Neural Networks: **hidden_layer_sizes, max_iter**


Thanks!