* **Random Forest** = n x DecisionTree: train n cây = n bộ training data, sau đó lấy n bộ decision tree 🡺 Voting 🡺 Prediction.
* he so tuong quan lon (>=0.7) 🡺 tinh chat tuyen tinh lon 🡺 dung cac mo hinh tuyen tinh (logistic linear,…)
* diabetes: nen dung cac mo hinh phi tuyen tinh (support vector machine, random forest,…)

(Theo ly thuyet, **di lam nen thu tat ca**)

* balance data: khi chenh lech >= 10 lan.
* standardScaler: fit(): tinh toan 🡪 transform(): bien doi dua vao tinh toan o fit

= fit\_transform()

dung fit() rieng, vi co the sau buoc fit() nguoi ta muon them tinh toan.

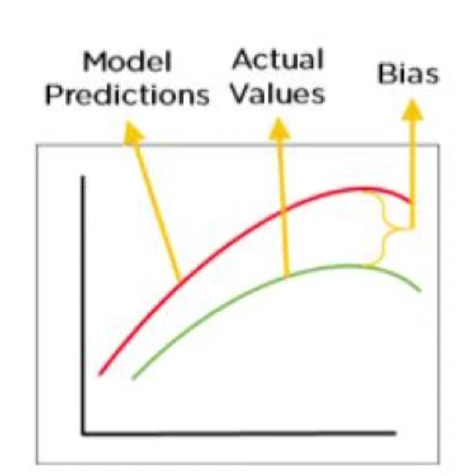
* tien xu ly phai lam sau khi phan chia (dam bao mo hinh ko biet duoc bo test truoc khi duoc train)
* chon mo hinh dua vao F1-score
* GridSearchCV: tim ra bo parameter tot nhat trong nhung bo dua ra
* nhieu gia tri (>100): dung one-hot ton bo nho 🡺 TFIDF

**Time series**

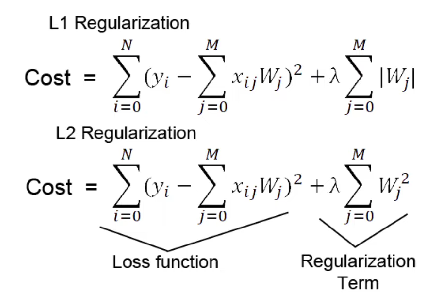
* Time series forecasting
* Time series data: bien doi ve dang data truyen thong để sử dụng ML: window sliding
* dự đoán nhiều giá trị trong tương lai (VD thời tiết, dự báo nhiều ngày sau)
* Recursive multi-step: đệ quy bằng window sliding. 🡪 **ưu điểm:** quá trình training nhanh, tiết kiệm bộ nhớ vì chỉ dùng 1 mô hình. **nhược điểm:** nếu bước đầu ko tốt 🡪 càng ngày càng sai lệch.
* Direct multi-step: dự đoán bao nhiêu giá trị trong tương lai thì cần bấy nhiêu mô hình, ko kết nạp prediction thành feature. 🡪 **nhược điểm:** cần nhiều mô hình, dự đoán càng xa càng giảm. 🡪 **ưu điểm:**

**Error in Machine Learning**

* Reducible Error: có thể giảm thiểu được
* bias: sai lệch giữa prediction và actual label. **under-fitting**



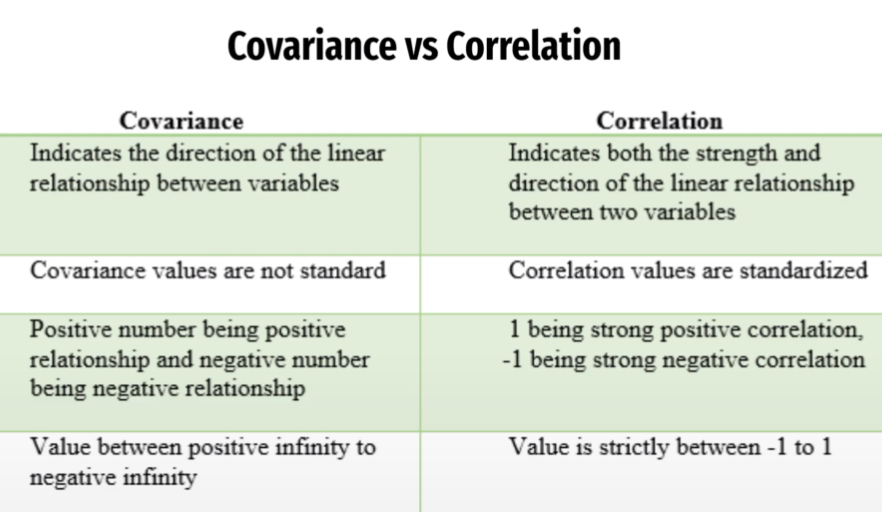
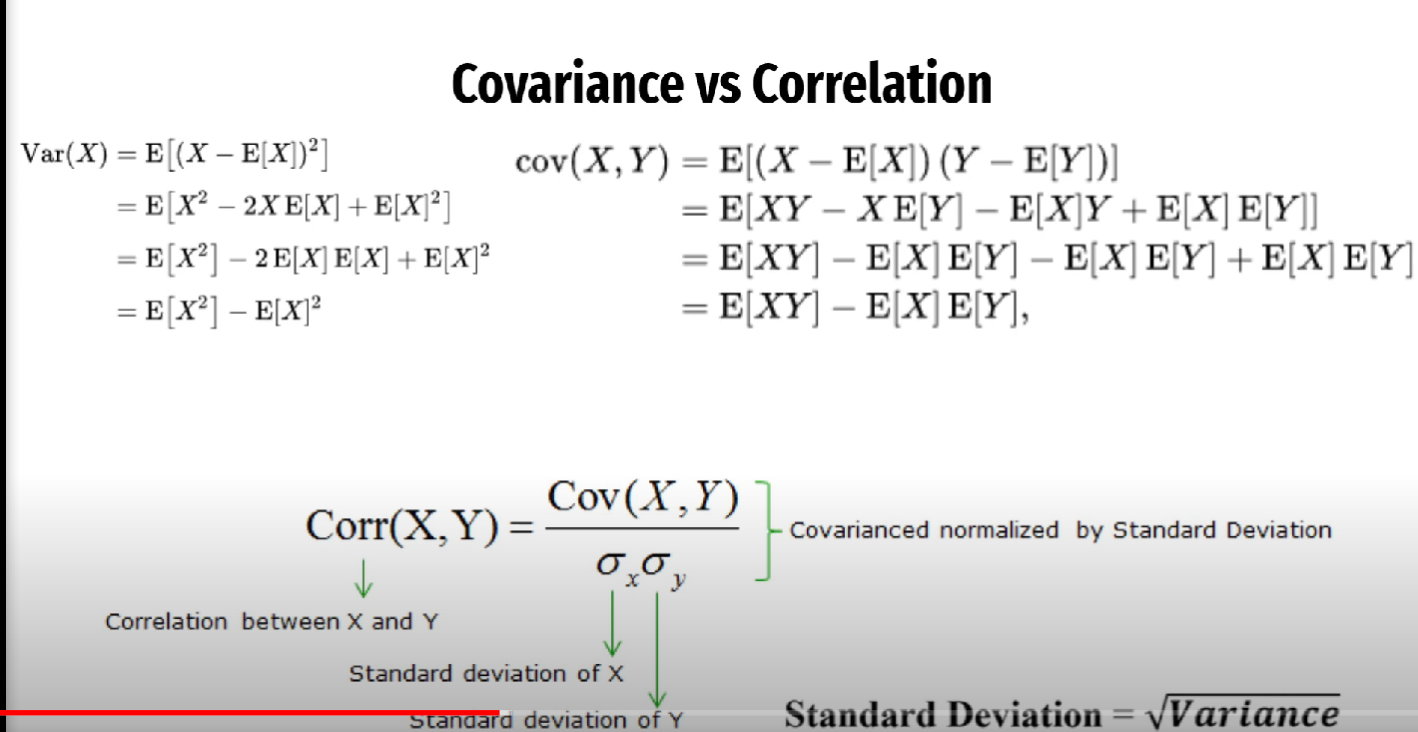
* variance: mức độ thay đổi của độ chính xác của prediction nếu input thay đổi. Cho biết khả năng tổng quát hóa của mô hình. **over-fitting**
* Irreducible Error: ko thể giảm thiểu được.
* **Choose ML model based on data:**
* Low bias + high variance: Decision tree, Random Forest, KNN, Kernel SVM 🡪 Phù hợp khi có nhiều data và ít features.
* High bias + low variance: Linear regression, Logistic regression, Linear SVM 🡪 Phù hợp khi có ít data và nhiều features.
* Extra: Nếu có nhiều data và nhiều features 🡪 giảm chiều dữ liệu, sử dụng Neural Network.
* **Choose ML model based on business domain:**
* Which metric is important? (accuracy, precision, recall, F1,…)
* What is the priority? (speed, memory, accuracy)
* Accuracy? (in medical field, accuracy is the most important factor)
* **Interpretability? giải thích kết quả**
* Giải thích được: Logistic linear, …
* Khó giải thích : SVM,… mô hình càng phức tạp, độ chính xác càng cao thì càng khó giải thích.
* **Tips for choosing ML model:**
* Start with a simple model: tiết kiệm resource, để hiểu, nếu mô hình đã tốt rồi thì ko cần chọn mô hình khác 🡪 tập trung cải thiện accuracy,…
* Try difference models and shortlist the best ones? : LazyPredict.
* Do Hyperparameter Tuning for each models?: GridSearchCV (small), RandomizedSearchCV (large).
* Compare amongst the best models with best hyperparameters to pick up the best one.
* **Regularization:** them vao ham mat mat mot dai luong nua de tac dong den ham loss. khong che do phuc tap cua mo hinh

****

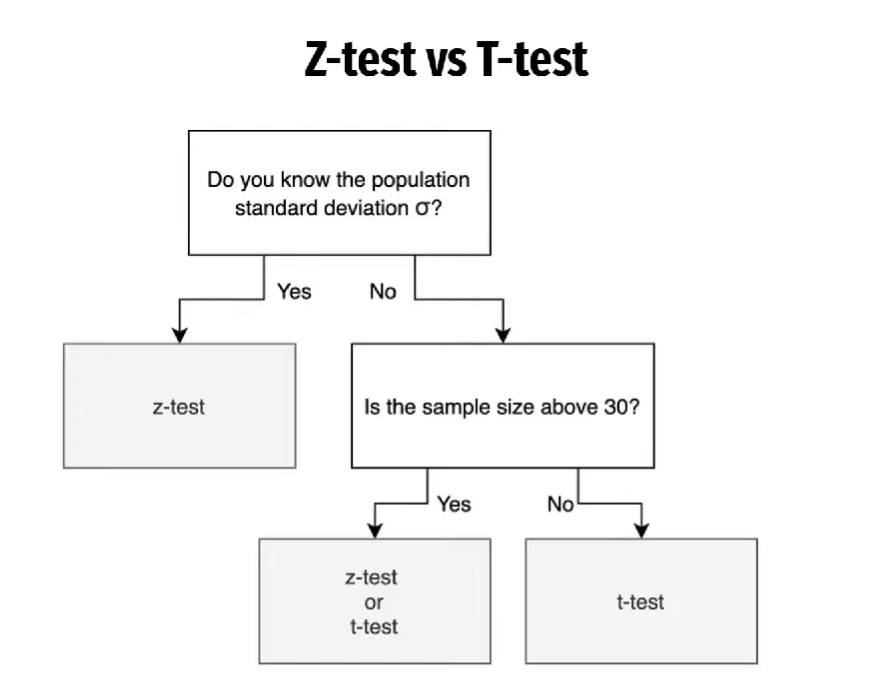
* **L1 Regularization (Lasso):** ép hệ số của các feature ít quan trọng về 0.
* **L2 Regularization (Ridge):**

**Dimensionality Reduction**

* **Feature Selection:** giu lai tap con feature quan trong nhat tu tap feature ban dau
  + dựa vào ma trận hệ số tương quan correlation coefficient. (hstq với target cao thì giữ lại).
  + nếu đa cộng tuyến 🡪 bỏ bớt.
  + variance threshold.
* **Feature Extraction:** convert dữ liệu sang 1 chiều ko gian mới (bằng phép nhân ma trận) 🡪 chọn feature trong chiều ko gian mới.
  + **Principle Component Analyst:** Unsupervised learning. Dùng nhiều trong visualization.
  + Khó quay lại giải thích.
  + Performance tốt hơn vì được transform sang chiều ko gian tối ưu.
* **So sánh variance 🡪 cần standard scaler để variance như nhau, nếu ko, feature có range thấp sẽ ít được quan tâm hơn.**
* **Khái niệm:**
  + **Curse of Dimensionality:** nhiều chiều 🡪 thuật toán khó xử lý được.
  + **Covariance vs Correlation:** hiệp phương sai vs hệ số tương quan



* **PCA:** 
  + **Step 1: Standardize the dataset**
  + **Step 2: Calculate the covariance matrix**
  + **Srep 3: Calculate engine**

****