# Machine learning

ML involves algo that learn rules or patterns from data to achieve a goal such as minimizing a prediction error.

**PCA**

the proportions of overall variance explained by the principal components.

PCA reduces dimensionality of the data. **Eigenvectors** describe how the variables of the system fluctuate together.

**Dynamic programming**

DP solves subproblems in a bottom-up approach.

**Monte Carlo**

It relies on **repeated random sampling** to obtain numerical results.

**Variance reduction**

To improve the **accuracy of Monte Carlo approximation**. By **reducing the variance** of the integrand.

**Bootstrap**

Approximate **distribution of statistic by Monte Carlo** simulation with **sampling** from the **empirical distribution** or the **fitted distribution of the observed data.**

**Confusion matrix**

A **2x2** matrix contains a **summary of prediction** result on a **classification problem**

The confusion matrix shows that when we predicts Y is classified as 1, it has accuracy rate of 51.7 percent for the linear model.

**What is P value**

The p value is the **minimum level of significance** where the **null hypotheses** can be **rejected** and in favor of the alternative hypothesis.

**Log transformed data**

Makes the distribution looks more symmetrical and normal. Moreover, the variance can be more equal.

### Mean squared error (MSE)

MSE is defined as Mean **or Average of the square of the difference between actual and estimated values.**

### Bias

The bias error is an **error from erroneous assumptions** in the learning algorithm. **High bias** can cause an algorithm to miss the relevant relations between features and target outputs **(underfitting).**

### Variance

The variance is an error from **sensitivity to small fluctuations** in the training set. **High variance** may result from an algorithm modeling the random noise in the training data **(overfitting).**

### Cross validation

CV is a popular strategy for model selection. The main idea is to split the data one or several times. This is done so that each split is used once as a **validation set** and the remainder as a **training set**: part of the data is used to train the algorithm, and the remaining part is used to estimate the algorithm's predictive performance. Then CV selects the algo with the smallest estimated error or risk.

### Overfitting and regularization

**There are multiple ways to address the risk of overfitting:**

*Dimensionality reduction*

It improves the feature to sample ratio by representing the existing features with **fewer, more informative, and less noisy features.**

*Ensemble models*

Such as random forests, combine multiple trees while **randomizing the tree construction.**

### Boosting

Boosting is an algorithm that **convert** a series of **weak** classifier into one **strong** classifier. **Boosting decreases the model's bias.**

**AdaBoost**
It changed the **weights** on the training data to reflect the cumulative errors of the current ensemble on the training set, before fitting a new, weak learner.

The algo starts with an **equally weighted training set** and then successively alters the sample distribution. After each iteration, AdaBoost **increases the weights of incorrectly classified observations and reduces the weights of correctly predicted samples** so that the subsequent **weak learners focus more on particularly difficult cases**. Once trained, the new decision tree is incorporated into the ensemble with a weight that reflects its contribution to reducing the training error.

**Random forests**
Decision trees are not only useful for their transparency and interpretability. They are also fundamental building blocks for more powerful ensemble models that combine many individual trees, while randomly varying their design to address the overfitting problems.

Ensemble learning involves combining several machine learning models into a single new model that aims to make better predictions than any individual model.

For ensemble learning to achieve this goal, the individual models must be accurate and independent.

**Two groups of ensemble methods:**
**Averaging methods**
This resembles the construction of a portfolio from assets with **uncorrelated returns** to **reduce the volatility** without sacrificing the return.

**Boosting methods**
Train base estimators sequentially with the specific goal of reducing the bias of the combined estimator. The motivation is to **combine several weak models into a powerful** ensemble.

**Bootstrap aggregation (Bagging)**

**Bagging**

Is a bootstrap select **classifier from the training** data with replacement with **equal weight** and put them into a **single predictive model**. The result is decided by the majority vote. **Bagging decreases the model's variance.**

**How bagging lowers model variance:**
**Randomizing** how each tree is grown
**Averaging** their prediction

This technique works best with **complex models** that have **low bias and high variance** such as deep decision trees, because its goal is to limit overfitting. Boosting methods, in contrast, work best with weak models such as shallow decision trees.

How to build a random forest
The RF algo builds on th**e randomization** introduced by bagging to further reduce variance and improve predictive performance.

**Pros and cons of random forests**
Advantages:
It provides a reliable feature importance estimate.
RF can perform on par with the best supervised learning algo.
They offer **efficient estimates** of the test error without incurring the cost of repeated model training associated with cross validation.

Cons:
Training a large number of deep trees can have **high computational costs** and use a lot of memory.

Predictions are slower, which may create challenges for applications that require low latency.
An ensemble model is inherently less interpretable than an individual decision tree.

# Shrinkage models (Combat Overfitting)

**How to hedge against overfitting: Shrinkage models**

*Ridge regression*
It shrinks the regression coefficients by adding a penalty to the objective function that

equals the sum of the squared coefficients which in turn corresponds to the L2 norm of the coefficient vector.

*Lasso regression*

A basis in signal processing, also s**hrinks the coefficients** by adding a penalty to the sum of squares of the residuals but the lasso penalty has a slightly different effect. The lasso penalty is the sum of the absolute values of the coefficient vector, which corresponds to its L1 norm. The lasso solution is a **quadratic programming problem.**

**Shrinkage method**: Ridge and Lasso: We can fit a model containing **all p predictors** using a technique that constraints or regularizes the coefficient estimates that **shrinks the coefficient estimates towards zero**. It turns out that shrinking the coefficient estimates can significantly **reduce their variance.**

**Ridge regression**

As lambda increases, the flexibility of the ridge regression fit decreases, leading to decreased variance but increased bias. Similar to least squares, ridge regression minimizes where lambda is a tuning parameter. Ridge regression seeks coefficient estimates that fit the data well by making the RSS small.

**Lasso regression**

It minimizes the quantity.    As with ridge regression the lasso shrinks the coefficient estimates towards zero. However, the L1 penalty has the effect of forcing some of the coefficient estimates exactly equal to zero when the tuning parameter lambda is sufficiently large.

Lasso has a major advantage over ridge regression., in that it produces simpler and more interpretable models that involve only a subset of the predictors.

Unlike ridge regression, the **lasso performs variable selection**, and hence results in models that are easier to interpret.

# Dimensionality reduction

**Curse of dimensionality**

As the **number of independent features grows** while the number of observations remain constant, the a**verage distance between data points also grows,** and the density of the **feature space drops exponentially,** with dramatic implications for machine learning. Prediction becomes much **harder when observations are more distance,** that is, different from each other.

Dimensionality reduction seeks to represent the data more efficiently **by using fewer features.**

DR **compresses the data** by finding a different, **smaller set of variables** that **capture** what matters **most in the original features** to minimize the loss of information.

**Principal Component Analysis (PCA)**
PCA aims to capture most of the variance in the data to make it easy to recover the original features and ensures that each component adds information. **It reduces dimensionality by projecting the original data into the principal component space.**

*Assumptions of PCA*
**High variance implies a high signal-to-noise ratio**
The data is standardized so that the variance is comparable across features.
**Linear transformations** capture the relevant aspects of the data.
Higher order statistics beyond the first and second moments do not matter, which implies that the data has a normal distribution.

**Singular value decomposition (SVD)**
The **PCA algorithm solves the problem** either **by computing the eigenvectors** of the covariance matrix **or by using the SVD.**

This algo is slower when the number of observations is greater than the number of features but yields better numerical stability, especially when some of the features are strongly correlated. SVD generalizes the eigendecomposition that we just applied to the **square and symmetric covariance matrix** to the more general case of mx n rectangular matrices.

In linear algebra, the singular value decomposition (SVD) is a factorization of a real or complex matrix. It generalizes the eigendecomposition of a square normal

matrix with an orthonormal eigenbasis to any {\displaystyle m\times n} matrix. It is related to the polar decomposition.

**Uniform Manifold Approximation and Projection (UMAP)**

UMAP is a more recent algorithm for visualization and general dimensionality reduction. It assumes the data is **uniformly distributed** on a locally connected manifold and looks for the closest low-dimensional equivalent using fuzzy topology.

UMAP is comparative with t-SNE for visualization quality and preserves more of the global structure with superior run time performance. UMAP has **no computational restrictions** o**n embedding dimension,** making it viable as a **general purpose dimension reduction technique** for machine learning.

Foundation for UMAP are based on **manifold theory** and **topological data analysis**. UMAP sues local manifold approximations and patches together their local fuzzy simplicial set representations to construct a topological representation of the high dimensional data.

First step: find a metric such that the data is uniformly distributed with regard to that metric.

As with other k-neighbour graph based algo, UMAP can be described in two phases. In the first phase a particular **weighted k-neighbour** is constructed. In the second phase a **low dimensional layout** of this graph is computed.

Axioms:
There exists a manifold on which the data would be **uniformly distributed.**
The underlying manifold of interest is locally connected
Preserving the topological structure of this manifold is the primary goal.

**Uniform distribution**

The distribution describes an experiment where there is an arbitrary outcome that lies between certain bounds.[1] The bounds are defined by the parameters, a and b, which are the minimum and maximum values. The interval can either be closed (e.g. [a, b]) or open (e.g. (a, b)).[2] Therefore, the distribution is often abbreviated U (a, b), where U stands for uniform distribution.[1]

**Weakness of UMAP**

It lacks the strong interpretability of PCA.
The dimension of the UMAP embedding space have no specific meaning.
Since UMAP is based on the distance between observations rather than the source features, it

does not have an equivalent of factor loadings that liner techniques such as PCA or Factor Analysis can provide.

**Define manifold**

In mathematics, a manifold is a **topological space** that locally resembles **Euclidean space near each point**. More precisely, an n-dimensional manifold, or n-manifold for short, is a topological space with the property that each point has a neighborhood that is homeomorphic to an open subset of n-dimensional Euclidean space.

**Define homeomorphism**

In the mathematical field of topology, a homeomorphism, topological **isomorphism,** or bicontinuous function is a continuous function between topological spaces that has a continuous inverse function.

**Topological data analysis**

In applied mathematics, topological based data analysis (TDA) is an approach to the analysis of datasets using techniques from topology. Extraction of information from datasets that are **high-dimensional, incomplete and noisy** is generally challenging. TDA provides a general framework to analyze such data in a manner that is insensitive to the particular metric chosen and provides dimensionality reduction and robustness to noise.

**t-distributed stochastic neighbor embedding (t-SNE)**
**Page 420**

t-SNE detects patterns in high-dimensional data. It takes a **probabilistic, nonlinear** approach to locate data on several different but related low dimensional manifolds.

t-distributed stochastic neighbor embedding (t-SNE) is a statistical method for visualizing **high-dimensional data** by giving each datapoint a location in a two or three-dimensional map.

The algo proceeds by converting **high dimensional distances** into conditional probabilities, where **high probabilities imply low distance** and **reflect the likelihood** of sampling two points based on **similarity.**

It accomplishes this by first positioning a **normal distribution** over each point and computing the density for a point and each neighbor, where the perplexity parameter controls the effective number of neighbors. In the second step, it arranges points in low dimensions and uses similarly c**omputed low dimensional probabilities to match the high dimensional distribution**. It measures the **difference between the distributions** using the **Kullback-Leibler divergence**, which puts a **high penalty on misplacing similar points in low dimensions.**

The **low dimensional probabilities** use **Student's t-distribution** with one degree freedom because it has **fatter tails that reduce the penalty** of misplacing points that are more distant in high dimensions to manage the crowding problem.

**t-SNE vs PCA**

t-SNE differs from PCA by preserving only small pairwise distances or local similarities whereas PCA is concerned with preserving large pairwise distances to maximize variance.

**Application of t-SNE**

t-SNE could be used on high-dimensional data and then the o**utput of those dimensions then become inputs to some other classification model.**

Also, t-SNE could be used to investigate, learn, or **evaluate segmentation.** Often times we select the number of segments prior to modeling or iterate after results. t-SNE can often times show clear separation in the data.

This can be used prior to using your segmentation model to select a cluster number or after to evaluate if your segments actually hold up. t-SNE however is not a clustering approach since it does not preserve the inputs like PCA and the values may often change between runs so it's purely for exploration.

*Weakness*: **computational complexity.** Does not facilitate the projection of new data points into the low dimensional space.

**Kullback-Leibler**

The Kullback Leibler distance (KL-distance) is a natural distance function from a "true" probability distribution, p, to a "target" probability distribution, q.

**Entropy**

Entropy is a way of measuring the **uncertainty/randomness o**f a random variable X. In other words, entropy measures the **amount of information** in a **random variable**. It is normally measured in bits. **KL divergence** is the measure of the **relative difference between two**

**probability distributions** for a given random variable or set of events. KL divergence is also known as Relative Entropy.

**Relevance to my previous project**

**2.1.3 SVM model with heavy-tailed error distribution (HTSVM)**

The HTSVM is an extension of SVM. Chan and Hsiao (2013) stressed that since Gaussian distribution used by the conventional volatility model has lack of mass for more extreme values, there is a need for adding a new weapon which can fully capture heavy-tailed distribution. Watanabe and Asai (2001) also showed that HTSVM can fit the financial data much better than its counterparts such as the normal and the generalized error distributions.

---

**The logistic regression model**

It models the probabilities of the output class given a function is linear in x, just like the linear regression model. The logit is called the log-odds since it is the logarithm of the odds. Logistic regression represents a logit that is linear in x.

**Bayesian**

Uncertainty is summarized by the prior. **Information** in the data comes through the **Likelihood.** After observing the data, we update our uncertainty. **Posterior** is the **new** summary of our **uncertainty.**

**Bayesian methods**

$\theta$ is a random quantity

P($\theta$) is a distribution that is called prior distribution

Bayesian comes from an observed sample y = ($y_1$,… $y_T$) via P(y| $\theta$) and the prior distribution, P($\theta$).

**JCC:**

It is assumed that the data vector, x, has been drawn from a conditional pdf f(x|$\theta$), where θ is a random vector of parameters. The PDF of $\theta$ conveys the *a priori* (existing beforehand, before any experience) information about θ. Observing the data x will affect our knowledge about $\theta$, and the way to update this information is to use Bayes' formula.

**Definition 8.1. (Prior, Likelihood, and Posterior).**

*The pdf of $\theta$ is called the prior pdf.

*The conditional pdf f(x|$\theta$) is called the Bayesian likelihood function.

*The central object of interest is the posterior pdf f($\theta|x$) which, by Bayes' theorem is proportional to the product of the prior and likelihood:

f($\theta|x$) ∝ f((x|$\theta$)f($\theta$)

**Bayes's theorem** states that the probability that event A occurs, given that event B has occurred, is equal to the probability that both A and B occur, divided by the probability of the occurrence of B: P(A|B) = $\frac{P(A\cap B)}{P(B)}$

**Let A = $\theta$ the set of parameters and B = y the observed data. Then P($\theta$|y) =** $\frac{P(\theta\cap y)}{P(y)}$ = $\frac{P(y|\theta)P(\theta)}{P(y)}$

Here P(y| $\theta$) is the likelihood and p($\theta$) is the prior probability distribution of the parameters.

**Constructing the posterior:**

As we wish to learn about the parameters $\theta$, we need not know p(y):

P($\theta$|y) = $\frac{P(y|\theta)P(\theta)}{P(y)}$ ∝ P(y| $\theta)P(\theta)$

Where:

P(y) = $\int P(y|\theta)P(\theta)d\theta$

We have the likelihood, P(y| $y|\theta$), and need only specify the prior P($\theta$).

**Concepts:**

P(y|$\theta$): the likelihood function

P($\theta$): the prior probability distribution of the parameter, $\theta$.

P($\theta$|y): the posterior probability distribution of the parameters.

P(y) = $\int P(y|\theta)P(\theta)d\theta$: the marginal likelihood or marginal distribution of the data.

# Labor economics

**The simple RBC model:**
Capacity utilization
Labor search and matching
Home production

By business cycle accounting, we compute four wedges: efficiency wedge, government wedge, consumption wedge, labor wedge, and capital wedge. The latter two can be considered labor income tax and capital income tax and therefore, they may affect the optimal decision of the households.

Higher labor tax will reduce your incentive to work.

Tax and government debt are used to finance the government consumption.

**Labor productivity**
To calculate a country's labor productivity, you would divide the total output by the total number of labor hours.

**Policies to Improve Labor Productivity**

There are a number of ways that governments and companies can improve labor productivity.

- **Investment in physical capital**: Increasing the investment in capital goods including infrastructure from governments and the private sector can help productivity while lowering the cost of doing business.
- **Quality of education and training**: Offering opportunities for workers to upgrade their skills, and offering education and training at an affordable cost, help raise a corporation's and an economy's productivity.
- **Technological progress**: Developing new technologies, including hard technology like computerization or robotics and soft technologies like new modes of organizing a business or pro-free market reforms in government policy can enhance worker productivity.

**G cubed:**

G-Cubed contains a strong foundation for analysis of both short run macroeconomic policy analysis as well as long run growth consideration of alternative macroeconomic policies. Intertemporal budget constraints on households, governments and nations (the latter through accumulations of foreign debt) are imposed. To accommodate these constraints, forward looking behavior is incorporated in consumption and investment decisions. G-Cubed also contains substantial sectoral detail. This permits analysis of environmental policies which tend to have their largest effects on small segments of the economy. By integrating sectoral detail with the macroeconomic features of the MSG2 model, G-Cubed can be used to consider the long run costs of alternative environmental regulations yet at the same time consider the macroeconomic implications of these policies over time. The response of monetary and fiscal authorities in different countries can have important effects in the short to medium run which, given the long lags in physical capital and other asset accumulation, can be a substantial period of time.

Overall, the model is designed to provide a bridge between computable general equilibrium models and macroeconomic models by integrating the more desirable features of both approaches.

**CGE (G-Cubed)**

- Dynamic
- Intertemporal
- General Equilibrium
- Multi-Country
- Multi-sectoral

- Econometric

Macroeconomic

**Household**

Number of households and their Demographic:

Region of residence, source of income: occupation, skills, assets, preference in consumption

**Decision made by households**

Consumption of goods and services

**Supply of labor**

Saving: supply of financial capital

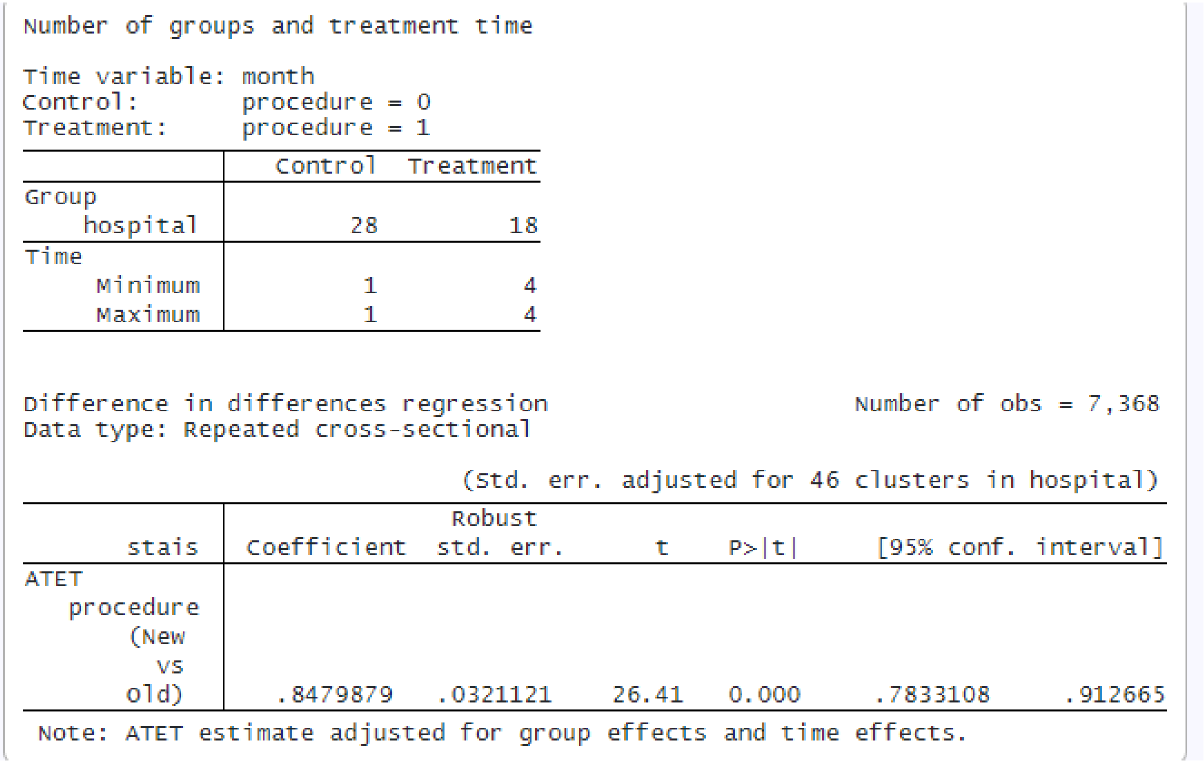
**Labor market**

- Labor mobility
    - ⇨ Complete mobility between sectors
    - ⇨ Immobile between regions
- Wages adjust slowly
    - ⇨ Respond to current and expected inflation
    - ⇨ Excess demand or supply of labor
    - ⇨ Unemployment can occur in the short to medium run
- Full employment in the long run

**Experimental design (Difference In Difference)**

Difference in differences (DID) offers a nonexperimental technique to estimate the average treatment effect on the treated (ATET) by **comparing the difference across time** in the differences **between outcome means in the control and treatment groups**, hence the name difference in differences. This technique controls for unobservable time and group characteristics that confound the effect of the treatment on the outcome.

Difference in difference in differences (DDD) adds a control group to the DID framework to account for unobservable group- and time-characteristic interactions that might not be captured by DID. It augments DID with another difference for the new control group, hence the name difference in difference in differences.



```
Number of groups and treatment time

Time variable: month
Control:       procedure = 0
Treatment:     procedure = 1
-----------------------------------
             |   Control  Treatment
-------------+---------------------
Group        |
    hospital |        28         18
-------------+---------------------
Time         |
     Minimum |         1          4
     Maximum |         1          4
-----------------------------------


Difference in differences regression                     Number of obs = 7,368
Data type: Repeated cross-sectional

                                 (Std. err. adjusted for 46 clusters in hospital)
---------------------------------------------------------------------------------
             |               Robust
       stais | Coefficient  std. err.      t    P>|t|     [95% conf. interval]
-------------+-------------------------------------------------------------------
ATET         |
   procedure |
        (New |
          vs |
        Old) |   .8479879   .0321121    26.41   0.000     .7833108     .912665
---------------------------------------------------------------------------------
 Note: ATET estimate adjusted for group effects and time effects.
```

The first table gives information about the control and treatment groups and about treatment timing. The first section tells us that 28 hospitals continued to use the old procedure and 18 hospitals switched to the new one. The second section tells us that all hospitals that implemented the new procedure did so in the fourth time period. If some hospitals had adopted the policy later, the minimum and maximum time of the first treatment would differ. The second table gives the estimated ATET, 0.85 (95% CI [0.78,0.91]). Treatment hospitals had a 0.85-point increase in patient satisfaction relative to if they hadn't implemented the new procedure.

**Model selection**

I would choose the method that gives me the lowest test MSE versus the lowest training MSE. Usually, the MSE will be small if the predicted responses get close to the true responses and large otherwise. Although, initially, we obtain the estimate from training observation, what we really want to know is whether that estimate is approximately equal to previously unseen test observations never been trained. In general, James et al (2013) stressed that the training MSE should always be smaller than test MSE. The reason is that most statistical models have tendency to minimize the training models. More specifically, training data is inclined to be over-fitted that makes the test MSE large. In sum, the patterns from method implemented in training data are prevented in the test data.

**KNN**

I would choose the method that gives me the lowest test MSE versus the lowest training MSE. Usually, the MSE will be small if the predicted responses get close to the true responses and large otherwise. Although, initially, we obtain the estimate from training observation, what we really want to know is whether that estimate is approximately equal to previously unseen test observations never been trained. In general, James et al (2013) stressed that the training MSE should always be smaller than test MSE. The reason is that most statistical models have tendency to minimize the training models. More specifically, training data is inclined to be over-fitted that makes the test MSE large. In sum, the patterns from method implemented in training data are prevented in the test data.