**DATA SCIENCE**

**[TERMS]**

\* Analysis: past events.

\* Qualitative Analysis: explain how? and why?

\* Quantitative Analysis: data + quantity in the past.

\* Analytics: potential future events.

\* Qualitative Analytics: intuition + analysis.

\* Quantitative Analytics: formulas + algorithms.

\* Business Intelligence: analyzing and reporting historical business data. make decisions. extract insights. extract ideas.

\* Machine Learning: make predictions. analyse patterns. give recommendations.

\* Artifical Intelligence: simulating human knowledge. decision making.

\* Data: information stored in digital format (symbols). traditional/big data. collect/measure/analyze/process.

\* Database: data collection organized in a way that it is easy to acces, manage and update.

\* Server: combination of hardware and software responsible for storing, managing and processing large amounts of data. web, database, ftp, etc.

\* Client-Server Model: client requests information. server responses.

\* Data Connectivity: connection of clients and servers, securing the swift and voluminous transfer of information between them.

\* Data Asset: data that is expected to have some value in the future. forms of data or software related to the processing of the information.

\* Application: program designed to perform a specific set of operations for the end user.

\* Programming: refers to the function the app performs. process of convertion inputs into outputs.

\* Interface: endpoints that relates device or software with specific functionalities to the outer world. plug of tv.

\* Application Programming Interface: API. contract allowing software to share data with each other. process the requests from the clients to the server. in response brings the requested data from the server.

\* API Application: colleciton of endpoints to which developlers can attach, and then extract specific information that can be used by those who are working with the apps.

\* Javascript Object Notation: json. allows communication betweeen software products through text files.

\* Integration: communication of different kinds of software with each other via APIs

**[BUSINESS ANALYTICS]**

\* Business Experience-Driven

- Business Case Studies (Past)

- Qualitative Analytics (Future)

\* Business Data-Driven

- Preliminary Data Report (Past)

\*Business Intelligence

- Reporting with Visuals (Past)

- Creating Dashboards (Past)

- Sales Forecasting (Future)

\*Artifical Intelligence

- Creating Real-Time Dashboards (Past)

- Client Retention

- Fraud Prevention

- Machine Learning

- Symbolic Reasoning

\* Data Analytics

- Data Science

**[PRELIMINARY DATA REPORT]**

\* first step of any data analysis.

**[DATA]**

\* raw data >> processing >> information >> business intelligence

\* processing: pre-processing, case-specific

\* pre-processing: class-labeling, data-cleansing, missing-values

\* class-labeling: numerical, categorical, text, digital media

\* case-specific: data balancing, shuffling, mining, masking,

\* database diagram: entity relationship diagram, relational schema

**[BUSINESS INTELLIGENCE]**

\* preliminary step of predictive analytics.

\* analyze past data and extract useful insights.

\* create appropriate models.

\* observation >> quantification >> measure >> metric

\* metric = measure + business meaning

\* key performance indicator = metric + business objective

\* visualizations: reports, dashboards

**[TRADITIONAL METHODS]**

\* predictive analytics

\* regression: causal relationships among different variables.

\* logistic regression, clustering, factor analysis, time-series

**[MACHINE LEARNING]**

\* data, model, objective function, optimization algorithm, trial and error

\* supervised learning: labelling data, minimizing loss, support vector machines, neural networks, deep learning, random forests, bayesian networks

\* unsupervised learning: unlabeled data, k-means, deep learning

\* reinforcement learning: maximizing reward

**[SOFTWARE]**

\* Languages: Python, Infographic (R), C, C++, Java, Scala

\* Database: SQL

\* Calculation: MatLab

\* Traditional Data: Excel, SPSS

\* Libraries: Apache Hadoop, Base, Mongo-DB

\* Business Intelligence Visualizations: Power BI, SAS, Tableau, Qlik

\* Econometric Time-Series Model: Eviews

\* Academic Statistical and Econometric Search. Stata

**[BAYESIAN NOTATION]**

\* set: collection of elements.

\* empty set (ø): a set that contains no elements.

\* x ϵ A: element x is a part of set A.

\* A ϶ x: set A contains element x.

\* x∉A: element x is not a part of set A.

\* Ɐx: for all x such that

\* A⊆B: A is a subset of B. A⊆A and ø⊆A

\* AՈB: intersection of two sets. elements those are on both sets.

\* AՍB: union of two sets. A + B - AՈB. all elements in both sets.

\* mutually exclusive: AՈB = ø

**[PROBABILITY]**

\* likelihood of an event occuring. value between 0 and 1. probability 1; absoulte certainty of event occuring. probability 0; absoulte certainty of event not occuring.

\* **theoretical probability:** P(A) = (preferred outcomes) / (sample space)

\* **experimental probability:** P(A) = (successful trials) / (all trials)

\* **sample space:** all possible events. P(A) + P(B) + P(C) + ... = 1

\* **dependent event:** an event that is affected by other event.

\* **independent event**: an event that is not affected by other event.

\* **conditional probability**

- probability of dependent events. P(A|B) = P(AՈB)/P(B) ≠ P(A)

- probability of independent events. P(A|B) = P(A)

- probability of simultaneous independent events. P(A, B, C, ...) = P(A) \* P(B) \* P(C) ...

\* **total probability law:** P(A) = P(A|B1)\*P(B1) + P(A|B2)\*P(B2) + ... (A is union of mutually exclusive sets B1, B2, ...)

\* **additive law:** P(AՈB) = P(A) + P(B) - P(AՈB)

\* **multiplication rule:** P(AՈB) = P(A|B) \* P(B) (probability of simultaneous events)

\* **bayes rule:** P(A|B) = [P(B|A) \* P(A)] / P(B) (how two events affect each other)

\* **expected value:** specific outcome we expect to occur when we run an experiment.

- categorical: E(Y) = n\*p (n: number of trials, p: probability)

- numeric: E(Y) =

- σ2 = E((Y- μ2)) = E(Y2) - μ2 (mean and variance relationship)

\* **frequency:** number of times a given value or outcome appears in the sample space.

\* **frequency distribution:** probabilities for each possible outcome of an event.

\* **frequency distribution table:** table matching each distinct outcome in the sample space to its associated frequency.

\* **probability frequency distribution:** divide every frequency by the size of the sample space.

\* **complement:** everything an event is not.

- (A’)’ = A

- P(A) + P(A’) = 1

- (A + A’) = (sample space)

**[COMBINATORICS]**

\* **combinatoric:** combinations of objects from a specific finite set. permutation, variation, combination.

\* **factorial:** product of all integers from 1 to n. n!, 0! = 1, n < 0; n! doesn’t exist.

\* **permutation:** number of different ways we can arrange a number of elements. P(n) = n!

\* **variation:** number of different ways we can pick and arrange a number of elements. order is relevant.

- repetitive: (n, p) = np (n: number of elements, p: number of positions)

- non-repetitive: V(n, p) = n! / (n-p)!

\* **combination:** number of different ways we can pick a number of elements. order is irrelevant. no multiples.

- repetitive: (n, p) = (n+p-1)! / [p! \* (n-1)!] (n: number of elements, p: number of selections)

- non-repetitive: C(n, p) = V(n, p) / P(p) = n! / [(n-p)! \* p!]

- symmetry:

- seperate sample spaces: C = n1 \* n2 \* ... \* np (ni: size of ith sample space)

**[PROBABILITY DISTRIBUTIONS]**

\* **distribution:** possible values a random variable can take and how frequently they occur.

- notation: variable ~ type (characteristics)

- Y: actual outcome

- y: one of the possible outcomes

- p(y) = P(Y=y): probability function. assigns probability to each distinct outcome in sample space.

- mean: average value of distribution.

- variance: measure on how spread out the data is. Var(Y)

- standard deviation: positive square root of variance.

- disreete distribution: finite outcomes. bar graph. P(Y ≤ y) = P(Y < y+1). discreete uniform, bernoulli, binomial, poisson

- continuous distribution: infinite outcomes. smooth curve. P(Y = y) = 0. P(Y ≤ y) = P(Y < y). normal, students-t, chi-squared, exponential, logistic

\* **population:** includes all possible outcomes of an event.

\* **sample:** includes only a few possible outcomes of an event.

\* **probability distribution of population:** mean (μ), variance (σ2), standard deviation (σ).

\* **probability distribution of sample:** mean (), variance (s2), standard deviation (s).

\* **trial:** observing an event and record the outcome.

\* **experiment:** collection of one or multiple trials.

\* **experimental probability:** probability of an event based on the experiment.

**[DISCREET - UNIFORM DISTRIBUTION]**

\* **Y ~ U(a, b)**

\* all outcomes have equal probability. dice, cards.

\* expected value and variance have no predictive power.

**[DISCREET - BERNOULLI DISTRIBUTION]**

\* **Y ~ Bern(p)**

- E(Y) = p

- p > (1-p)

- σ2 = p\*(1-p)

\* single trial and two possible outcomes. true/false. coin.

**[DISCREET - BINOMIAL DISTRIBUTION]**

\* **Y ~ B(n, p)** (n:number of trials, p:probability of success)

- Bern(p) = B(1, p)

- p(y) = C(y, n) \* py \* (1-p)n-y

- E(Y) = p \* n (number of heads expected after flipping coin 10 times is 5.)

- σ2 = n \* p \* (1-p)

\* two possible outcomes per iteration and many iterations. sequence of identical bernoulli events.

\* how likely an event is to occur over a series of trials.

**[DISCREET - POISSON DISTRIBUTION]**

\* **Y ~ Po(λ)**

- λ: anticipated value.

- p(y) = (λy \* e- λ) / y!

- E(Y) = σ2 = λ

\* likelihood of a certain event occuring (frequency) over a given interval of time.

\* how likely a specific outcome is, knowing how often the event usually occurs.

\* out of ordinary or not.

\* only non-negative values.

**[CONTINUOUS DISTRIBUTIONS]**

\* **probability density function (PDF):** f(y)

\* **cumulative distribution function (CDF)**: F(y) = P(Y≤y)

\* **probability of individual event:** P(Y=y) = 0

\* **probability of interval:**

\* **PDF to CDF:**

\* **CDF to PDF:**

\* **expected value:**

\* **variance:**

**[CONTINUOUS - NORMAL DISTRIBUTION]**

\* **Y ~ N(μ, σ2)**

- E(Y) = μ

- outliers: {μ-σ, μ+σ}

\* natural events. size of animals in nature.

\* bell-shaped. symmetric. thin tails.

\* %68 of values should fall within (μ-σ, μ+σ). (68%, 95.99%, 99.7%)

\* shift left: decrement μ.

\* shift right: increment μ.

\* higher peak-thinner tails: decrement σ

\* lower peak-fatter tails: increment σ

**[STANDARD NORMAL DISTRIBUTION - Z-TRANSFORM]**

\* transformation: altering every element of a distribution to get a new distribution.

- add: shift-right

- subtract: shift-left

- multiply: shrink

- divide: expand

\* Z-Transform: standard normal distribution to z-score table.

- E(Y) = 0, Var(Y) = 1

- Y ~ N(μ, σ2) → Z ~ N(0, 1)

- z = (y-μ)/σ

- Z: standardized variable

- z: critical value (value read from table of known values)

**[CONTINUOUS - STUDENTS’ T DISTRIBUTION]**

\* **Y ~ t(k)**

- k: degrees of freedom

- k > 2 → E(Y) = μ, Var(Y) = (s2\*k)/(k-2)

\* small sample size approximation of a normal distribution.

\* bell-shaped. symmetric. fat tails.

**[CONTINUOUS - CHI-SQUARED DISTRIBUTION]**

\* **Y ~ χ2(k)**

- E(Y) = k

- Var(Y) = 2k

- Y ~ χ2(k) → ~ t(k)

- Y ~ t(k) → Y2 ~ χ2(k)

\* hyphotesis testing for goodness of fit.

\* asymmetric. skewed right. non-negative values.

\* contains a table of known values.

**[CONTINUOUS - EXPONENTIAL DISTRIBUTION]**

\* **Y ~ Exp(λ)**

- λ: rate parameter. how fast reaches to plateau.

- E(Y) = 1/λ

- Var(Y) = 1/λ2

- Y ~ Exp(λ), X = ln(Y) → X ~ N(μ, σ2) (transformation of exponential distribution to normal distribution)

\* rapidly changing events. online website traffic, radioactive decay.

\* starts off high. initially decreases. eventually plateauing.

\* not contains a table of known values. so transformation is necessary.

**[CONTINUOUS - LOGISTIC DISTRIBUTION]**

\* **Y ~ Logistic(μ, S)**

- S: scale (smaller scale closer to 1)

- E(Y) = μ

- Var(Y) = (s2\*π2)/3

\* how continuous variable intputs can affect probability of of a binary outcome. forecast analysis. team sports.

\* CDF picks up near mean. cut-off point.

**[STATISTICS]**

\* **data set:** population (N parameters, whole), sample (n statistics, partial)

\* **types of data:** categorical, numerical (discreet, continuous)

\* **levels of measurement:** qualitative (nominal:no-order, ordinal:order), quantitative (interval, ratio:true-zero)

\* **visualizations for categorical data:** frequency distribution table, bar chart, pie chart, pareto diagram, cross table

\* **visualizations for numerical data:** frequency distribution table, histogram, scatter plot

\* **measures of central tendency:** mean, median, mode

- **mean:** simple average of data set. population mean: μ, sample mean: . /n

- **median:** midpoint of the ordered data set. position of median in data set: (n+1)/2

- **mode:** value that occurs most often. value with highest frequency.

\* **skewness:** measure of asymmetry. dataset is concentrated on one side or not.

-

- **positive (right-skewed):** mean>median>mode

- **negative (left-skewed):** mean<median<mode

- **zero (symmetric):** mean=median=mode

\* **measures of variability:** dispersion of data around its mean value. variance, standard deviation, coefficient of variation

- coefficient of variation (CV): standard-deviation/mean

- population variance: σ2 = → population standard deviance: σ → CV = σ/μ

- sample variance: s2 = → sample standard deviance: s → CV = s/

\* **measures of relationship:** covariance, linear correlation coefficient

- **covariance:** measure of correlation between two variables.

- positive: two variables move together.

- negative: two variables move in opposite directions.

- zero: two variables are independent.

- population covariance:

- sample covariance:

- **correlation:** standardized measure of correlation between two variables. values between [-1, 1].

- perfect positive correlation (1): one variable can be explained by other entirely.

- perfect negative correlation (-1): one variable can be explained by other entirely.

- absolutely independent variables (0): one variable cannot be explained by other.

- correlation coefficient: cov(x, y)/[stdev(x)\*stdev(y)]

- population correlation:

- sample correlation:

- causality: correlation symmetrical with respect to both variables. cov(x,y) = cov(y,x). correlation doesn’t imply causation.

**[INFERENTIAL STATISTICS]**

\* **distribution:** function that shows the possible values for a variable and how often they occur.

\* **discreet uniform distribution:** all outcomes have an equal chance of occuring.

\* **central limit theorem:** sampling distribution of the means approximates a normal distribution.

\* **sampling distribution:** distribution formed by samples.

- ~ N(μ, σ2/n)

- distribution of

- k: number of samples.

- n: size of samples.

- more samples closer to normal. k → ꝏ

- bigger samples closer to normal. n → ꝏ

\* **standard error:** standard deviation of the distribution of sampling distribution.

\* **estimator:** function that approximates a population parameter depending only on sample information.

- mean(μ) estimator:

- variance(σ2) estimator: s2

- correlation(ρ) estimator: r

- smaller variance more efficient estimator.

- bias: expected value of an estimator is (parameter+bias).

- efficiency: most efficient estimator is the one with the smallest variance.

\* **estimate:** output of estimator.

- point estimate: single value. located in the middle of confidence interval.

- confidence interval: interval. more precise than point estimates.

**[CONFIDENCE INTERVALS]**

\* **confidence interval:** interval within which we are confident the population parameter will fall.

- level of confidence: (1-α). 0<α<1. common α: 0.01, 0.05, 0.1

- reliability factor (population, z-statistic):

- reliability factor (sample, ): , (, degrees of freedom)

- margin of error (ME):

- margin of error (population variance, z-table, narrow):

- margin of error (sample variance, t-table, wide):

- confidence interval: [ - ME, + ME]

\* **confidence interval of two samples**

- variance of the difference:

- dependent:

- population known/independent:

- population assumed equal/independent:

-

-

- population assumed different/independent:

**[HYPOTHESIS TESTING]**

\* **scientific method:** systematic observation, measurement, experiment, formulation, testing, modification

\* **hypothesis:** idea that can be tested. supposition made on limited evidence as a starting point for further investigation.

\* formulate hypothesis, find right test, execute test, make decision

\* **null hypothesis:** H0. hypothesis to be tested. status quo. (yerleşik inanç). decisions: accept or reject.

\* **alternative hypothesis:** all hypothesis except null. H1, HA. challenging. (aykırı söylem)

\* **rejection region:** region at the tails of standard normal distribution where we reject the null hypothesis.

\* **acceptance region:** middle region of the standard normal distribution excluding rejection region.

\* **significance level (α):** probability of rejecting the null hypothesis that is true. common: 0.01, 0.05, 0.1. if observed statistic is too far away from 0 depending on the significance level we reject the null, otherwise we accept it.

- two-sided test: null contains equality or inequality. =, ≠

- one-sided test: null doesn’t contain equality or inequality. <, >, ≤, ≥

\* **statistical errors:**

- type 1 error: false positive. reject a true null hypothesis. probability is α.

- type 2 error: false negative. accept a false null hypothesis. probability is β.

- reject false null hypothesis. probability 1-β is power of the test.

\* **p-value:** smallest level of significance at which we can still reject the null hypothesis given the observed sample statistic.

- notable p-values: 0.000 (reject all), 0.05 (cut-off line)

- closer to zero more significant result.

\* **hypothesis testing methodology**

- calculate a statistic .

- scale it .

- check if z is in the rejection region. Z closer to z, more acceptable hypothesis. accept if |Z|< z, reject if |Z| > z.

- same applies for t-score.

- z-statistic: big samples, known/unknown variances.

- t-statistic: small samples. unknown variances.

- critical value: number-from-table.

- reject hypothesis if;

- |test statistic| > |critical value|

- p-value < significance-level

\* **hypothesis testing formulas (one population)**

- variance known:

- variance unknown:

\* **hypothesis testing formulas (two population)**

- dependent:

- independent-variance known:

- independent-variance assumed equal:

-

-

**[REGRESSION ANALYSIS]**

\* regression analysis: prediction method for variables which have cause-relationship between them. exam mark - acceptance

\* linear regression: linear approximation of a causal relationship between two or more variables. straight line.

\* simple linear regression model (population): (: intercept, : slope, : error)

\* simple linear regression equation (sample): (: estimated, x1: observation, b0: intercept, b1: slope)

\* sum of squares total (SST): (total variability)

\* sum of squares regression (SSR): (explained variability)

\* sum of squares error (SSE): (unexplained variability)

\* SST = SSR + SSE

\* Ordinary Least Squares Model (OLS) Regression parameters:

- dependent variable: variable to be predicted. y.

- coefficient of the intercept: b0

- coefficient of the independent variable: b1

- p-value of t-statistic: P>|t|. independent variable is significant or not. 0.000

- p-value of f-statistic: overall significance of the model.

- error method: least squares. minimum squares error. find minimum SSE by estimator.

- OLS estimator of β for simple linear regression:

- R-Squared: R2 = SSR/SST. variability of data. [0, 1]

- adjacent R-Squared: considering number of independent variables also. negative values are interpreted as 0.

\* Other Regression Methods: generalized least squares, maximum likelihood estimation, bayesian regression, kernel regression, gaussian progress regression

\* dummy variable: used to include categorical data into a regression model. yes to 1, no to 0, ...

\* standardization (feature-scaling): process of transforming data into a standard scale.

\* standardized variable: (x: original variable)

**[MULTIPLE LINEAR REGRESSION]**

\* multiple linear regression model (population):

\* multiple linear regression equation (sample):

\* adjusted R-Squared: (n: sample size, observations, p: estimators, predictors)

\* f-Statistic: testing overall significance of the model. lower f-statistic closer to non-significant model.

**[LINEAR REGRESSION ASSUMPTIONS]**

\* linearity: (line equation)

\* non endogeneity:

- ommited variable bias: difference between observed value and predicted value is correlated with independent variable. caused by missing relevant variable.

\* normality/homoscedasticity: , errors are normally distributed and having equal variances.

\* no autocorrelation:, errors are uncorrelated

\* no multicollinearity:

**[LOGISTIC REGRESSION]**

\* predicts the probability of an event occuring.

\* possible outcomes are categorical, not numerical.

\* assumptions: no endogeneity, normality/homoscedasticity, no autocorrelation, no multicollinearity

\* logistic regression model:

\* logistic regression model (odds):

\* logit regression model: log(odds) = = linear regression model (assumed ε = 0)

\* maximum likelihood estimation (MLE): machine learning process that tries to maximize likelihood function.

\* likelihood function: measures the goodness of fit of a statistical model. estimates how likely it is that the model at hand describes the real underlying relationship of variables. bigger likelihood function higher the probablity that model is correct.

\* log likelihood: log of likelihood. more popular metric than likelihood function. almost but not always negative. higher is better.

\* log likelihood null: log likelihood of a model which has no independent variables. check if model has explanatory power. it is used as the benchmark worst model.

\* log likelihood ratio test (llr): f-test for logistic regression. checks if model is statistically different from LL-null or worst model, in other words useless. lower is better.

\* pseudo R-Squared: comparing variations of same model. different models have non comparable pRS values. [0.2-0.4]

\* binary predictors: corresponding to dummy variables in logistic regression.

**[CLUSTER ANALYSIS]**

\* multivariate statistical technique that groups of observations on the basis some of their features or variables they are described by. dividing observations in the data set into groups (clusters).

\* goal of clustering is to maximize the similarity of observations within a cluster and maximize the dissimilarity between clusters. market segmentation, image segmentation.

\* classification: predicting an output category, given input data. linear regression, logistic regression. supervised learning.

\* clustering: grouping data points together based on similarities among them and difference from others. minimizing the distance between points in a cluster and maximizing distance between clusters. unsupervised learning.

\* within-cluster-sum-of-squares (wcss): divergence of points in a cluster. minimize as possible to obtain perfect clustering. clusters small, wcss low.

\* types of clustering:

- flat: number of clusters are chosen prior to clustering. no hierarchy. k-means

- hierarchical: explores all possible solutions at each step. taxonomy of the animal kingdom. dendrogram, heatmap.

- divisive (top-down): start with all observations are in same cluster. then split this cluster step by step.

- agglomerative (bottom-up): start from different clusters. then combine them until reaching whole cluster.

**[K-MEANS CLUSTERING]**

\* choose number of clusters. specify the number of seeds. assign each point to a centroid. adjust centroids.

\* centroid: mean position of a group of points. center of mass.

\* seed: starting centroid. selected randomly or by an algorithm.

\* elbow method: choosing number of clusters.

\* standardization: reducing the weight of higher numbers while increasing lower ones.

\*types of analysis:

- exploratory: data acquainting, pattern searching, planning. data visualization, descriptive statistics, clustering.

- confirmatory: confirm hypothesis. validate previous research.

- explanatory: explain phenomenon.

**[BASIC LINEAR ALGEBRA]**

\* mxn Matrice: (m: rows, n: columns, mxn: elements)

\* scalar: numbers having no dimension. 1 by 1 matrix. point. rank 0 tensor.

\* vector: numbers having one dimension. row vector (m:1). column vector (1:n). m or n is length. line. rank 1 tensor.

\* matrice: numbers having two dimensions. collection of numbers ordered in rows and columns. rank 2 tensor.

\* tensor: multi-dimensional array.

\* transposing matrice: mxn matrice to nxm matrice. rows to columns, columns to rows. scalar/vector transposing to same.

\* vector multiplication: dot (inner) product, tensor (outer) product.

\* dot product: sum of the products of the corresponding elements.

\* multiplication of matrices: . row vector to column vector.

**[NEURAL NETWORKS]**

\* training an algorithm: data, model, objective function, optimization algorithm

\*types of machine learning: supervised, unsupervised, reinforcement

- supervised: target outputs for given inputs are determined. classification (categorical output), regression (numerical output)

- unsupervised: no target outputs.

- reinforcement: reward after achieving goal.

\* target: desired value at which we are aiming.

\* linear model: (x: input, y: output, w: weight, b: bias, t: target)

\* objective function: measures how well the model’s output match the desired correct values. loss (supervised), reward (reinforcement).

\* loss function: lower the loss function higher the accuracy. loss:L(y, t), cost:C(y, t), error:E(y, t). L2-norm, cross entropy.

\* reward function: higher the reward function higher the accuracy.

\* L2-norm: loss function of regression. OLS. lower the error lower the loss. (y: outputs, t: targets)

\* cross entropy: loss function of classification.

\* gradient descent: optimization algorithm. . η should be high enough to minimize time, low enough not to oscillate. (oscillation: repetitive variation around a central value.) .

\* n-parameter gradient descent for linear model:

-

-

-

-

**[MACHINE LEARNING - TENSORFLOW]**

\* basic terms: independent variable = feature = input = attribute, coefficients = weights. higher weight higher impact.

\* label variable: nominal data. distinguishes the individuals from one another, does not carry any information. ID.

\* positive weight: feature increases in value. example: bigger engine higher price.

\* negative weight: feature decreases in value. example: higher mileage lower price.

\* dummy variable: assigning numerical value for categorical data.

\* benchmark: categorical data that is not taken as dummy as a rule. positive weight, more expensive than the benchmark. negative weight, less expensive than the benchmark.

\* layer: building block of neural networks. input → linear combination + non-linearity → output

\* node: building block of a layer. node = unit.

\* hidden node: building block of a hidden layer.

\* width of layer: number of nodes in the layer.

\* width of net: number of nodes in the biggest layer.

\* depth: number of hidden layers.

\* parameters: weights, biases. found by optimizing.

\* hyperparameters: width, depth, learning rate, batch size, momentum coefficient, decay coefficient. preset manually by user.

\* non-linearity: activation (transfer) function. models arbitrary functions. represents complicated relationships. it doesn’t change the shape of the expression just its linearity. two consecutive linear transformations are equivalent to single one. in order to stack layers non-linearity is added. activation functions transforms inputs into outputs of a different kind. non-linearities → stacking layers → depth → deep learning.

\* common activation functions: sigmoid (logistic function), tanh (hyperbolic tangent), rectified linear unit (ReLu), softmax. monotonic, continuous, differentiable.

\* deep neural network: has more than one layer. input layer → hidden layers → output layer (targets)

\* stacking layers: process of placing one layer after the other in a meaningful way.

\* softmax activation: transforms a bunch of arbitrarily large or small numbers into a valid probability distribution. considers information about the whole set of numbers. ranges from 0 to 1.

\* backpropagation: forward propagation is the process of pushing inputs through the net. at the end of each epoch , the obtained outputs are compared to targets to form the errors. backpropagation of errors is an algorithm for neural networks using gradient descent.it consists of calculating the contribution of each parameter to the errors. we backpropagate the errors through the net and update the parameters (weights and biases) accordingly

\* overfitting: machine learning training has focused on the particular training set so much, thus it has missed the point. captures all noise, high train accuracy, low test accuracy. prevention by splitting data into training (80%) and testing (20%) parts.

\* underfitting: model has not captured the underlying logic of the data. low train accuracy.

\* variance inflation factor (VIF): multicollinearity check. produces a measure which estimates how much larger the square root of the standard error of a estimate is when assumed that the variable is completely uncorrelated with other predictors.

\* residual: differences between the targets (actual values) and the predictions in machine learning. much lower than the mean: overestimation, much higher than the mean: underestimation. (y\_train - y\_hat)

\* bias-variance tradeoff: balance between underfitting and overfitting.

\* dataset: training (80%) + validation (10%) + test (10%). validation detects and prevents overfitting. test measures the final predictive power of the model. if training loss goes hand-in-hand with validation loss move along. if validation loss is increasing overfitting.

\* n-fold cross validation: training and validation sets are combined. then divide this data set into chunks. at each epoch one of the chunk is considered as validation set.

\* early stopping: technique to prevent overfitting. stop training early.

- preset number of epochs.

- stop when updates become too small.

- validation set strategy.

\* initialization: process of setting initial values of weights.

\* xavier (glorot) initialization: method is not so important, number of inputs and outputs is.

- uniform: in [-x, x] for

- normal: of 0, and a standard deviation

\* optimization: algorithms used to vary model’s parameter.

- stochastic gradient descent: updates the weights many times inside a single epoch, in batches.

- batching: process of splitting the dataset in n batches.

- momentum: rate of change in weight. (current update - previous update, α=0.9)

\* learning rate: small enough for gentle descend and prevent oscillation. big enough for reasonable time.

\* adaptive learning rate schedules: adaptive gradient algorithm, root mean square propagation, adaptive moment estimation.

\* adaptive gradient algorithm (adagrad): Gi(0) = 0

- (ϵ: small number that prevents divide by zero)

- (adaptive magic, monotonously increasing)

\* root mean square propagation (rmsprop): Gi(0) = 0

- (β=0.9, hyperparameter)

\* adaptive moment estimation: adds momentum. Mi(0) = 0

-

-

\* preprocessing: any manipulation of the dataset before running it through the model. compatibility, orders of magnitude, generalization. numerical: standardization, normalization, principal components analysis, whitening. categorical: one-hot encoding, binary encoding.

\* standardization: feature scaling. process of transforming data into a standard scale. figures of different scales appear similar.

\* normalization: converitng each sample into a unit length vector using L1 or L2 norm.

\* principal components analysis: dimension reduction technique used to combine several variables into a bigger variable.

\* whitening: removes most of the underlying correlations between data points.

\* binary encoding: convert numbers that represent categorical data into binary and take each binary digit as variable. better for many categories.

\* one-hot encoding: create many columns as there are possible values. variables are uncorrelated and unequivocal. better for few categories.

\* action plan: data + model + objective function + optimization algorithm

- preprocessing: prepare data and preprocess it. balance the dataset. divide dataset in training, validation, test datasets. save the data in tensor friendly format.

- machine learning algorithm: outline the model and choose the activation funcitons. set the appropriate advanced optimizers and the loss function. make it learn. test the accuracy.