**DATA SCIENCE FUNDAMENTS**

**Topics:**

* **Linear Regression**
* **Polynomial Regression**
* **Logistic Regression**
* **Decision Tree**
* **KNN ( K Nearest Neighbours )**
* **Clustering- Kmeans**
* **Clustering- Hierarchical Clustering**
* **Dimonsionaliiy reduction: Feature Analysis + PCA**
* **Regularization + Lasso Regression**
* **Linear Regression**

Understand the behavior of certain variable (dependent )by studing the influence another variable (independent variable) has on it.

We want to predict Y based on X

Ex: Does “age” affect “ income”?

The simplest from of regression

Linear model is -> a straight line throught the data.

The higher X, the Higher or lower Y

“line of the best fit”

[A diagram of a graph

Description automatically generated](https://www.geeksforgeeks.org/ml-linear-regression/?ref=header_outind)

Linear relation = linear function

. in math I will be like this: *f(x)=ax +b , yi= B0 + Bix*

**What is B0 and Bi?**

.=> Beta 0 is the intercept

. where the function crosses the X-axis

. value of Y when x=0

* Beta 1 is the shope

Positive Beta 1 -> the functions grows

Negative Beta 1 -> the function lowers

The increase amount Y with each increase of of X

**Multiple linear regression:**

A new “ Slope” is created for each parameter

f(x)=ax+bx+cx+dx+….+e

yi=B0 +B1x+B2x+ B3x+b4x+…

B0=intercept

Validating our model:

**Multiple metrics**

. Model itself

e.g R2

. Compare your model to other models:

.Mean Absolute Error (MAE)

. Mean Squared Error (MSE)

R^2

Only used in linear regression.

Represents the amount of variation of the dependent variable (y) explained by the independent variable variable (x)

Always between 0 and 1

The higher, the better.

Does not require testing data.

A good indicator of how good a model is on it’s own.

**Mean Absolute Error (MAE) :**

. Represents how accurate a regression model is.

. Calculated by the average of all the errors made in testing

. Requires testing data.

. A good indicator of how good a model when comparing models

**Mean Squared Error (MSE):**

. Similar to Mean Absolute Error (MAE)

. Squares each error

. Metric “punishes” outlines -> indicate overfitting

Often used over MAE  
Linear regression: **Supervised** learning algorithm

**Y= mX+c**

Dependent and independent variables:

1. Dependent variable (y): the value we are trying to predict.
2. Independent variable (x): The features used to make prediction

Line of best fit: the line that minimizes the distance between it self and all the data points.

**Types of linear regression:**

1. Simple linear regression: 1 independent variable (x) and 1 dependent variable (y)

Example: predications a person salary based on years of experience

1. Multiple linear regression: multiple independent variables and one dependent variable

Example: predicating house prices y based on featues like size, number of bedrooms and location.

**Advantage and disadvantages:**

**Advantage:**

. Simple and easy to interpret.

. Fast and efficient for small to medium datasets.

. Works well when the relationship between x and y is **linear.**

**Disadvantages:**

. Assusmes a linear relationship, which is often not ture.

.Sensitive to outliners.

.Cannot model more complex relationships between X and y.

**Some Short Question with Answer:**

**Q1: What is linear regression and how does it works?**

A1:Linear regression is a statical method used to model the relationship between a dependent variable and one or more independent variables. It works by fitting line to the data points that minimizes the difference between the predicted and actual values.

**Q2: What is the role of the cost function in linear regression?**

A2: The cost function measures the error in the model by calculating the difference between predicted and actual values, often using mean squared error. The objective is to minimize this cost to find the best-fitting line.

**Q3:What does R-squared singnify in linear regression?**

A3:R-squared indicates the proportion of the variance in the dependent variable that is explined by the independent variabls. A higher R-squared means a better fit of the model to the data.

**Q4:Whys us it important to check for multicolliinearity in linear regression?**

A4: Multicolliinearity occurs when idependent variables are highly correlated with each other. It can make it difficult to determine the individual effect of each variable and reduce the reliability of the coefficient estimates.

**Q5:How can linear regression handle categorical variables?**

**A5:** Categorical variables can be converted into numerical format using techniques like one-hot encoding or dummy variables, establing them to be used in linear regression models.

**Q6: What are residuals, and why are they important?**  
**A6:** Residuals are the differences between actual and predicted values. Analyzing residuals helps assess the model’s accuracy and identify patterns that may indicate violations of assumptions.

**Q7: What are the key assumptions of linear regression?  
A7:** The assumptions are:

1. Linearity: The relationship between variables is linear.
2. Independence: Observations are independent.
3. Homoscedasticity: Residuals have constant variance.
4. Normality: Residuals are normally distributed.

**Q8: What is overfitting in linear regression, and how can it be avoided?  
A8:** Overfitting happens when the model captures noise in the training data, reducing its ability to generalize to new data. It can be avoided by simplifying the model or using regularization techniques like Ridge or Lasso regression.

**Q9: How does multiple linear regression differ from simple linear regression?  
A9:** Simple linear regression involves one independent variable predicting one dependent variable, while multiple linear regression involves two or more independent variables predicting a single dependent variable.

**Q10:** **Why is feature scaling important in linear regression?**  
**A10:** Feature scaling ensures that all independent variables contribute equally to the model by standardizing their ranges. This is particularly important when the features have widely different units or scales.

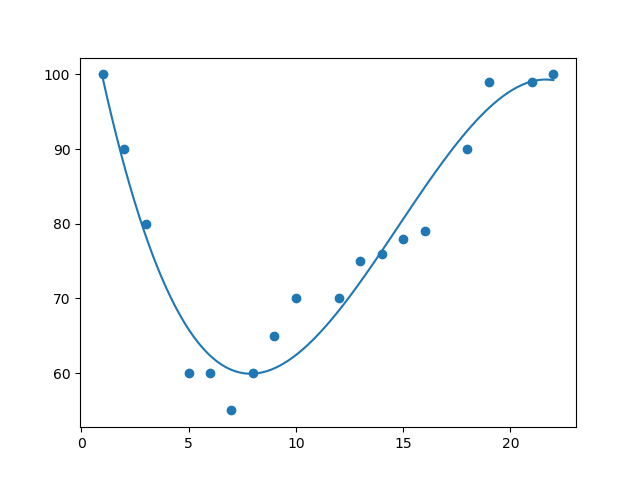
* **Polynomial Regression**

From PDF:

A dummy variable is a numeric variable that repesents categorical data, such as gender, race, political affiliation etc. it only takes the value 0 and 1 to indicate the absence or presence of said effect that may be expected to shift the outcome.

**Interaction effect:**

1. Occurs when the effect of one variable depends on the value of another
2. Captures combined influence of variables, beyond their individual effects
3. Often represented by multiplying two variables
4. Useful for modeling more complex relationships in data.
5. Helps reveal if variables together have a stronger or different impact than separately.



**Underfitting :**

1. Model is too simple to capture underlying patterns.
2. Model fails to learn from the training data.
3. Poor performance on the both training and test data.

**Overfitting:**

1. Model is too complex, capturing noise and random fluctiuations.
2. Model Learned too well from training data.
3. Performs well on training data but poorly on new unseen data.

**Extra:**

**What is polynominal regresssion?**

* Polynomial regression is a technique used when the relationship between the dependent variable y and the independent variable x is non-linear. Instead of fitting a straight line to data, we fit a curve that better captures the pattern.
* Y= dependent variable ( target)
* X =independent variable ( input features)
* n= degree of the polynomial

if n =1 it becames a linear regression mode . For n=2 , it becames quadratic regression model and so on.

**Why use polynominal regression ?**

Linear regression works well when the relationship between x and y is linear. However if there is non-linear relationship , linear regression fails to capture the pattern. In such cases, polynominal regression fits a curve that better explains the trend in the data.

**Example:**

1. if we try to model the growth of a tree using a straight line, it won’t fit well.
2. But by fitting a curve we can capture the upward growth trend accurately.

**Key concepts :**

**# Degree of the Polynomial (n)**

1. The degree of the ploynomial determines the complexity of the curve
2. N=1 (linear), n=2 (quadratic) n=3 (cubic) etc;

**Overfitting and underfitting:**

1. Underfitting: if the degree is too low (n=1) the model might be too simple to capture the compltexity of the data.
2. Overfitting: if the degree is too high n=10, the model might capture nosie instead of the true pattern resulting in poor generalization.
3. They key is to select the right degree of the polynomial to balance bias and variance**.**

**SHORT QUESTION:**

**Q1: What is polynomial Regression?**

A1: Polynomial regression is a form of regression alaysis where the realtionship between the independent variable x and the dependent variable y is modeled as an n degree polynomial. It is used to capture non-linear relationship in the data that cannot be modeled effectively with linear regression.

**Q2: How does polynomial regression differ from linear regression?**

A2: Linear regression models a straights-line relationship between the variables, assuming that the target variable changes linearity with the predictor variable. Ploynomial regression, on the other hand,extands this by adding higher-degree terms, allowing the model to capture curves and non-linear patterns in the data.

**Q3: When should you use polynomial regression?**

A3: Polynomial regression is appropriate when the data exhibits a curved relationship between the independent and dependent variables. If plotting the data shows a non-linear pattern, a polynomial model might provide a better fit than a linear one.

**Q4: What is the general equation for a 2nd-degree polynomial regression?**

A4: The general equation is y=β0+β1x+β2x2+ϵy = \beta\_0 + \beta\_1x + \beta\_2x^2 + \epsilony=β0​+β1​x+β2​x2+ϵ, where β0,β1,β2\beta\_0, \beta\_1, \beta\_2β0​,β1​,β2​ are coefficients representing the intercept and weights of the polynomial terms, and ϵ\epsilonϵ represents the error term capturing the noise in the data.

**Q: Can polynomial regression be performed using linear regression algorithms?  
A:** Yes, polynomial regression can be implemented using linear regression by transforming the input features. For example, by adding x2,x3x^2, x^3x2,x3, etc., as additional features, the regression algorithm treats it as a standard linear regression problem with multiple features.

1. **Q: What is the risk of using a very high-degree polynomial in regression?  
   A:** High-degree polynomials can lead to overfitting, where the model fits the training data too perfectly but fails to generalize to new data. This results in a complex curve that captures noise rather than the true underlying pattern in the data.
2. **Q: How do you preprocess data for polynomial regression in Python?**  
   **A:** To preprocess data, you can use the PolynomialFeatures class from the sklearn.preprocessing module. This generates additional polynomial and interaction terms, enabling the regression model to fit higher-degree polynomials to the data effectively.
3. **Q: What metric is typically used to evaluate the performance of a polynomial regression model?**  
   **A:** Common metrics include Mean Squared Error (MSE), which measures the average squared difference between predicted and actual values, and R-squared (R2R^2R2), which indicates the proportion of variance explained by the model. These metrics help assess how well the model fits the data.
4. **Q: How can you determine the optimal degree of the polynomial?**  
   **A:** The optimal degree can be determined using cross-validation, where the dataset is divided into training and validation subsets. By testing different degrees and evaluating their performance on validation data, you can identify the degree that balances bias and variance most effectively.
5. **Q: Is it necessary to scale the data before performing polynomial regression?  
   A:** Scaling is important, especially for high-degree polynomials, as the magnitudes of polynomial terms can grow exponentially. Without scaling, this can lead to numerical instability during optimization and affect the performance of the regression model.

* **Logistic Regression**

**From PDF:**

# Regression, but not really

1. In it’s core a from of regression
2. S-shape **curve**
3. Based on X, we predict Y
4. Y always lies between 0 and 1

**Regression as Classification :**

[A diagram of a curve

Description automatically generated](https://www.geeksforgeeks.org/understanding-logistic-regression/?ref=header_outind)

**Binary classification:**

1. Always 0 or 1
2. Predicts if a certain factor is true or false

e.g=spam, cancer.

3. Smple yet limited

**Multiclass classification:**

1. Can be anything
2. Predicts the type of a certain factor

Eg= wine quality rating…

1. More complicated yet unlimited.

Is classification supervised or unsupervised? **Supervised.**

**Evalution Matrics:**

1. **True Positives (TP)** : The amount of label outcome is 1 and the predicted label is 1. In simple words we correctly predicted the positvie outcome.
2. **True negative (TN**): The amout of label outcome is 0 and the predicted label is 0. Simple words we correctly predicted the negative outcome.
3. **False Positive (FP):** The amout of label outcome is 0 and the predicted label is 1. Simple words we incorrectly predicted the negative outcome.
4. **False Negative (FN):** The amout of label outcome is **1** and the predicted label is **0.** Simple words we correctly predicted the negative outcome.

**Confusion Matrix:**

1. The same goes for multiclass classification
2. Calculate TP/TN/FP/FN per class
3. Create an overall idea of the model by calculating **an average**.

**Accuracy :**

1. Between **0** and **1**
2. Higher is the better.

**Precision:**

1. Masures how often the same prediction will be made, given more or less the same independent variables
2. Higher is the better.

**Recall:**

1. Also called the True Positive rate (TPR)
2. Proportion of actual positives correctly indentified
3. Facus on minimizing false negatives

Important in mdeical diagnoses.

**Specificity:**

1. Also called the True negative rate(TNR)
2. Proportion of actual negatives correctly indentified
3. Focuses on minimizing false postives

Important fraud detection,…..

**F1 Score:**

1. Harmonic mean of precision and recall
2. **Gives a more balanced overview than accuarcy**

**SHORT QUESTION:**

Q1: What is logistic regression?

A1: Logistic regression is a statical method used to model the relationship between a dependent binary variable and one or more independent variables.it predicts probabilites that the dependent variable belongs to a particular category, typically 0 and 1.

Q2: How does logistic regression differ from linear regression?

A2: While linear regression predicts a continuous output,logistic regression predicts probability, which are then converted into binary outcomes using a threshold. Logistic regression uses the sigmoid function to constrain predictions between 0 and 1.

Q3: When should logistic regression be used?

A3: Logistic regression is ideal when the dependent variable is categorical (e,g binary classification like “yes” or “No”. It’s is commonly used in tasks such as spam detection, medical diagnosis, and customer chum predication.

Q4: What is the sigmoid function in logistic regeneration?

A4: The sigmoid function, s(x)=1/1+e^-x,maps any real-valued number to the range between 0 and 1. In logistic regression, it is used to convert the linear combination of inputs into a propability for classification.

Q5:How are model cofficients interpreted in logistic regression?

A5:Model coefficients represent the log-odds of the dependent variable. A positive coefficents increases the odd of the outcome being 1, while a negative coefficents decreases it. The magnitude indicates the strangth of the relationship between the variable and the outcome.

Q6: What are some assumptions of logistic regression?

A6: Logistic regression asssums (1) the dependent variable is binary ,(2) there is little or no multicollinearity among predictions, (3) the predictions are linearly related to the log-odds of the outcome, and (4) large sample sizes for reliable results.

Q7:What are common evalution metrics for logistic regression?

A7: Common metrics include accuracy, precision, reacll, F1-score, and Area under the curve . these metrics assess how well the model distingusihes bwtween the classes and handels imbalanced datasets.

Q8: How can logistic regression handle multi-class classification problems?

A8: Logistic regression can handle multi-class classification using extensions like one-vs-Rest, where separate binary models are built for each alss, or softmax regression , which generalizes the sigmoid function to handle mutiple classes.

Q9: What are the risks of using logistic regression on non-linear data?

Logistic regression assumes a linear relationship, between predictors and the log-odds of the outcome. On non-linear data, it may underperform due to it’s inability to capture complex patterns, leading to underfitting and poor predictions.

Q10: How do regularizations techniques affect logistic regression?

A10: Regularization techniques like L1 (Lasso) and L2 (Ridge) prevent overfitting by penalizing large coefficients. L1 can also peform feture selection by shrinking some coefficients to zero, while L2 reduces the magnitude of coefficients to stabilize the model.

* **Decision Tree**

A **Decision Tree** is a **supervised machine learning** algorithm used for both **classification and regression** tasks. It works by splitting the data into subsets based on the feature values, which helps make decisions or predictions. The tree structure represents decisions in a flowchart-like format, where each internal node is a decision based on a feature, each branch represents an outcome of the decision, and each leaf node represents a final prediction or outcome.

**Decision Trees:**

1. Basically a flowchart
2. Mimics human decision making
3. Server splits where you ask a question
4. Based on the answer we can ask further questions.  
   [A diagram of a tree

   Description automatically generated](https://www.geeksforgeeks.org/decision-tree-introduction-example/?ref=header_outind)

**Inner Workings:**

1. Each split is based on one or two variables
2. Split so as to maximize information gain

Each split must deliver information gain

1. We keep asking questions, but two important things in our mind

**Overfitting :**

**We don’t have** to worry about this since it’s classification

**Recognise overfitting :**

1. Overfitting occurs in all data science models, even in classification
2. Also in Logistic regression, even though we glossed over it.
3. A model can ask too many questions / wrong questions
4. Compare **testing accuracy** vs **training accuracy**.

**Higher accuracy on training data + low accuracy on testing data= overfitting**

**Combat overfiting:**

1. Same fix as polynominal regression:

Keep polynomial degree low

Keep amount of nodes low

1. The fewer questions a model asks the more it needs to learn from those questions
2. Learning trends > recognizing values

**Metrics :**

**The same classification metrics apply:**

1. Accuracy
2. Precision
3. Recal
4. Specifity
5. F1 Score

**Random forest :**

1. Extanded from of a decision tree
2. Chain of decision trees
3. Each tree:

Gets a subset of data

Makes a prediction

1. Majority vote
2. Can also used for regression analysis.

**Decision tree for Classification vs Regression?**

**# Classification trees:** Used when the output variable is categorical. Each leaf node contains a class label.

Example : Predict whether an email is spam or not spam based on featutes like the subject line and content.

**# Regression Trees**: Used when the output variable is continuous. Each leaf node contains a predicted numeric value.

Example: predict the price of a house based on features like size, number of rooms, and location.

**Advantages and Disadvantages of Decision trees:**

**Advantages:**

1. Decision trees are easy to visualize and interpert, even by non-experts.
2. It can work with a variety of data types.
3. Unlike algorithms like SVM or KNN, decision trees don’t require normalization or scaling of features.
4. Decision trees can capture non-linear relationships between features and the target.

**Disadvantages:**

1. Decision trees can easily **overfit the** data, easpacially when the tree is deep.
2. Small changes in the data can result in a completely different tree.
3. If some features are more dominant or have more levels, the tree may favor them for splitting.
4. If not pruned or controlled, decision trees may not generalize well to unseen data.

**Pruning Decision Trees:**

Pruning is the process of cutting back the tree to avoid overfitting. There are two types of pruning:

**Pre-pruning (Easy Stoping) :** Stops the tree from growing when it reaches a certain depth or other stoping criteria.

**Post-pruning ( Cost Complexity pruning) :** prunes branches after the tree has been built by evaluating the performance on a validation set.

**SHORT QUESTION:**

Q1: What is decision tree?

A1:A decision tree is a supervised learning algorithm used for classification and regression task. It models decisions and their possible consequences as a tree-like structure, with nodes representing features, branches representing decision rules, and leaves representing outcomes.

Q2: How does a decision tree work?

A2: decision tree splits the dataset into subsets based on feature values. Starting from the root node, it applies a series of decisions (e.g., thresholds or conditions) to partition the data recursively, creating branches until it reaches a leaf node that represents the final output.

**Q:** What are the advantages of decision trees?  
**A:** Decision trees are easy to interpret and visualize, can handle both categorical and numerical data, and do not require feature scaling. They are also capable of capturing non-linear relationships and interactions between features.

**Q:** What are the limitations of decision trees?  
**A:** Decision trees are prone to overfitting, especially when the tree is deep or the dataset is noisy. They are sensitive to small changes in the data, which can lead to different splits and outcomes. Additionally, they can struggle with imbalanced datasets.

**Q:** What is the criterion used for splitting in decision trees?  
**A:** Common criteria for splitting include **Gini Impurity**, **Entropy (used in Information Gain)**, and **Mean Squared Error (MSE)** for regression tasks. These criteria measure how well a split separates the data into distinct classes or reduces variability.

**Q:** How do decision trees handle categorical and numerical features?  
**A:** For numerical features, decision trees find optimal thresholds to split the data (e.g., x>10x > 10x>10). For categorical features, they create splits based on unique categories or subsets of categories. Modern implementations like scikit-learn handle both seamlessly.

* **KNN ( K Nearest Neighbours )**

**KNN is** one of the simplest and most intuitive **supervised** machine learning algorithms used for classification and regression. It predicts the output for a new data point based on ‘k’ closest points in the dataset.

**What is KNN?**

* **KNN** is a non-parametric and instance -based learning algorithm. Instead of building an explict model, it memorizes the training data and makes predictions on-the fly.

When a new data points is given, **KNN** calcualtes the distance between this point and all points in the training set. It then selects the K closest neighbours and assigns the new point the majority class or the average of the neighbours values.

[A diagram of a group of data

Description automatically generated with medium confidence](https://www.geeksforgeeks.org/k-nearest-neighbours/?ref=header_outind)

**Metrics:**

Once more, the same metrics apply:

1. Accuracy
2. Precision
3. Recall
4. Specificity
5. F1 Score

**Overfitting:**

**# Knn tendsd to overfit on low values of k**

* Doesn’t learn patterns, just remembers some data

On high values it tends to underfit

* Doesn’t learn patterns, just does the dame for every datapoint

The best value is somewhere in between

**Cross validation:**

1. Used to compare different models
2. Create folds in the data
3. For each fold, train a model with that fold as test set.
4. Take the average outcome : Classification -> majority vote

**K-fold:**

1. ‘K’ represnts the amount of folds
2. Most common model : 10 fold
3. The value of ‘K’ cannot be higher than the amount of datapoints!

**How Does KNN Work?**

1. Choose K: Select the number of neighbours (k) to consider for predicition.
2. Calculate Distance: For a new data point, calculate the distance to all points in the training dataset using a distance metric ( like manhattan distance)
3. Select neighbours: pick the K closest neighbours ( smallest distances).
4. Make a Prediction:
5. Classification: Use a majority vote among the k neighbours to determine the class of the new point.
6. Regression: Use the mean of the k neighbours as the predicted value.

**Example :**

Imagine you have a dataset of fruits with features like color (red, green) and size (small, big). You want to classify a new fruit as either an **apple** or an **orange**.

If **K = 3**, the algorithm finds the 3 nearest points. If 2 out of 3 are apples and 1 is an orange, the new fruit is classified as an **apple** (majority vote).

**Choosing the Value of K**

The value of **K** plays a critical role in the performance of the KNN algorithm.

* **Small K (like K=1 or K=3)**: High variance, sensitive to noise, and may lead to overfitting.
* **Large K (like K=20 or K=50)**: Can smoothen the decision boundary, leading to better generalization, but may underfit the data.
* **Optimal K**: The best K is often chosen using **cross-validation**.

**How to Select the Best Value of K?**

To select the best value of K, we can:

1. Use **Cross-validation**: Try different values of K and evaluate the model on a validation set.
2. Use an **Odd K**: For binary classification (two classes), use an odd K to avoid ties.
3. Use **Elbow Method**: Plot the error rate for each K and choose the point where the error stops decreasing significantly

**Advantages and Disadvantages of KNN:**

**Advantages :**

1. Simple to understand and implement.
2. No training time since it memorizes the dataset.
3. Works well for smaller datasets where the relationship between points are simple.
4. No assumptions about data distribution (non- parametic)

**Disadvantages:**

1. Computationally expensive for large datasets as it requires distance computation for all points.
2. Sensetive to irrelevent features and requires features scalling.
3. Imbalanced data can affect predictions since the majority class might dominate.
4. As the number of features increases, the distance between points becames less meaningful.

* **Clustering- Kmeans**

K-Means is an **unsupervised** machine learning algorithm used for clustering data into a specified number of groups (clusters). It partitions the data points in a such way that points within the dame cluster are more similar to each other than to those in other clusters.

**What is K-Means Clustering?**

K-Means clustering divides a dataset into **K clusters** based on the similarity of data points. It works by minimizing the distance between data points and the centroid (center) of their respective clusters.

The algorithm attempts to find groups in the data where:

1. Data points in the same cluster are **as close as possible** to each other.
2. Data points in different clusters are **as far apart as possible**.

**K means:**

1. Basic clustering algorithm.
2. Start with a parameter ‘K’ =amount of clusters
3. Divide dataset amoungest each cluster.
4. Data is categorized without a specific label.

[A diagram of a diagram of a circle and a circle with a circle and a circle with a circle and a circle with a circle with a circle and a circle with a circle with a circle and

Description automatically generated](https://www.geeksforgeeks.org/ml-k-means-algorithm/?ref=header_outind)

**Inner Workings:**

1. Assign random starting point to each cluster (centriod)
2. Assign each datapoint to nearest centriod.
3. Move cluster to middle of each assigned datapoint
4. Repeat untill coveraged.

**Elbow method:**

1. Similar to finding the best values of k for knn.
2. Train a bunch of models.
3. Calculate distortion per model
4. Average squared distance between each datapoint and its nearest centriod .
5. Find the right balance between low distortion and low value of k.
6. Pick the value in the middle of the “Elbow”.

**Overfitting:**

We +don’t need to worry about overfitting because **it’s clustering**. Clustering can be **overfit.**

1. Clustering models can’t overfit like regression or classification.
2. There is no label to memorize
3. Overfitting occurs when define too many clusters.
4. Underfitting occurs when too few clusters.

**Evaluation metrics:**

1. No labels so there is no evaluation in the model
2. We can only confirm by ploting the data.
3. 2 columns data, create graph.

Accuracy

Precision

Correlation Matrix

As it is clustering no need to train and split the dataset.

**Advantages and Disadvantages of K-Means**

**✅ Advantages**

* **Simple and easy to implement.**
* **Fast and efficient** for small to medium datasets.
* Scales well to large datasets (with optimizations).
* Useful for a wide range of applications (e.g., market segmentation, document clustering).

**❌ Disadvantages**

* Requires specifying the number of clusters (KKK) in advance.
* Sensitive to **outliers** and **noise**, which can distort cluster boundaries.
* Assumes clusters are spherical and equally sized, which may not always hold.
* Results can vary based on the initialization of centroids.

**Application of K-Means:**

1. Market Segmentaiton: Grouping customers based on purchasing bahavior.
2. Document clustering: Organizing text data into topic-based groups.
3. Image compression: Reducing color in an image by clustering similar pixel colors.
4. Anomaly Detection : Identifiying unsual patters or bahaviors

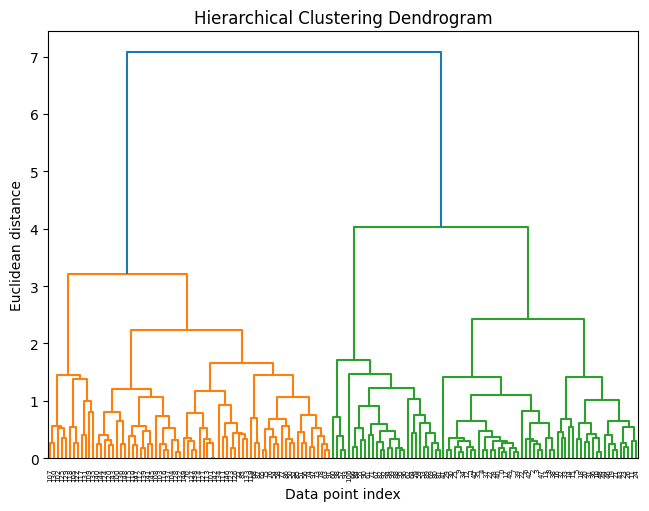
* **Clustering- Hierarchical Clustering**

Hierarchical clustering is **an unsupervised learning** algorithm used to group data into a tree-like structure based on their similarity. It is different K-means because it does not require to specify the number of clusters in advance. Instead, it creates a hierarchy of clusters that can be visualized as a dendrogram.

1. More advanced clustering method.
2. Flexible approach to choosing the amount of clusters.
3. Bottom-up mechanic.

Each individual point starts off as a unique cluster, at the end there is only 1 cluster left.

1. Clusters are formed by merging the clusters which are closest together.

 [Click here. To know more](https://www.geeksforgeeks.org/hierarchical-clustering-with-scikit-learn/?ref=header_outind)

**Dendogram :**

* 1. Visual representation of all clusters **from max to 1**
  2. X-axis show the datapoints
  3. Y-axis shows the distance between clusters
  4. Lines show the grouping of clusters (linkage )
  5. Define colours for each cluster

**Linkage criteria:**

1. A way to figure out how good a cluster is
2. Excat inner working differ per model
3. Most common one ward
4. Calculates inertia ( inner cluster variance, how close are datapoints within a cluster) for each cluster pair.
5. Similar working as Kmeans -> Good for spherical data
6. Dendogram: y-axis plots inertia

**Overfitting :**

1. We have to take it in mind
2. Overfittting is clustering : too many clusters
3. Like kmeans this is easy to counter
4. Elbow method .

**Linkage Methods**

When calculating the distance between clusters, there are several approaches:

**🔹 Single Linkage**

The distance between two clusters is the **shortest distance** between any two points in the clusters.

* Tends to create long, "chain-like" clusters.

**🔹 Complete Linkage**

The distance between two clusters is the **longest distance** between any two points in the clusters.

* Tends to create compact, spherical clusters.

**🔹 Average Linkage**

The distance between two clusters is the **average distance** between all pairs of points in the two clusters.

**🔹 Centroid Linkage**

The distance is measured between the **centroids** (mean positions) of the clusters.

**Advantages and Disadvantages**

**✅ Advantages**

1. **No need to pre-specify the number of clusters.**
2. Produces a dendrogram, providing insights into the hierarchy of the data.
3. Works well with small to medium-sized datasets.

**❌ Disadvantages**

1. Computationally expensive (O(n3)O(n^3)O(n3)) for large datasets.
2. Sensitive to outliers, as they can form their own clusters.
3. Does not work well for **non-hierarchical** data structures.

**Applications**

* **Genomics**: Grouping genes with similar expression patterns.
* **Social Network Analysis**: Understanding communities or relationships.
* **Market Segmentation**: Grouping customers based on purchasing behavior.
* **Document Clustering**: Organizing text documents by topic.
* **Dimonsionaliiy reduction: Feature Analysis + PCA**

**Dimensionality Reduction:**

Dimensionality reduction is a technique used to reduce the number of features (dimensions) in a dataset while preserving as much information as possible. It is crucial in machine learning and data analysis to combat the "curse of dimensionality," simplify models, and improve computational efficiency.

1. Sometimes data is too big
2. We want to keep all data points
3. It’s a **unsupervised** reduction
4. Reduce the amount of variables in a dataset
5. Ideal scenrio no information loss.
6. Realstic scenerio minimal information loss.

**PCA: Principal Component Analysis:**

PCA is one of the most popular techniques for dimensionality reduction. It transforms the data into a new set of uncorrelated features called **principal components**, which are linear combinations of the original features.

1. Reduce the variables to a few principal components
2. Only one of the many dimensionality reduction techniques available
3. Bundle variables with **high correlations** together
4. Create a new variable to replace the two previous ones
5. Try to keep as much explained variance in the dataset as possible

A graph with green dots and red lines

Description automatically generated

**PCA Example:**

* 1. Consider a dataset with multiple variables, two of which have a linear correlation
  2. Plot a linear regression model to predict the values of the first variable based off the second variable
  3. Predict this value for all datasets
  4. This list becomes the new variable which explains both old variables.

**Eigenvalues:**

1. Reducing the number of components is nice, but how do we know when to stop?

**Finding the Best amount of components:**

1. Plot the eigenvalues in function of the number of components
2. Elbow method

**Overfitting:**

1. It’s a dimensionality reduction technique, overfitting **is not as matter**
2. Just be sure don’t reduce the data or loss valuable information.
3. Same as with clustering just use elbow method.

**Key Concepts of PCA**

**a. Variance and Principal Components**

* **Principal Components (PCs)**:
  + Each PC is a direction in the feature space that captures the maximum variance in the data.
  + The first PC captures the **most variance**, the second PC captures the **second most** variance, and so on.
* **Goal**:
  + Project the data onto a lower-dimensional subspace while retaining most of the variability.

**b. Orthogonality**

* PCs are orthogonal (uncorrelated), ensuring no redundancy between components.

**c. Dimensionality Reduction**

* By keeping only the top kkk PCs (where k<number of features < \text {number of features}k<number of features), PCA reduces the dimensionality of the data.
* **Regularization + Lasso Regression**

Lasso regression is a type of **linear regression** that incorporates L1 regulation to prevent overfitting and perform feature selection. It modifies the cost function of linear regression by adding a penalty proportional to the sum of the absolute values of the coefficients.

Feature Selection:

1. Selecting the most important (or least redundant) features of a dataset.
2. Reduces size and variety of dataset.
3. Improves processing time when creating ML models
4. Finding a balance between dataset size and model accuracy
5. Similar goals to PCA but remember PCA =/= feature selection!
6. PCA creates new features, feature selection only drops redundant ones.

A graph showing a growth rate

Description automatically generated with medium confidence

**Lasso Regression:**

* 1. Least absolute shrinkage and selection operator
  2. Takes a base linear or polynomial regression model
  3. Adjusts the coefficients using penalization.

**Penalization:**

1. Adding a penalty term to punish complex models
2. Promotes simple models that catch a broader sense of the dataset.
3. The higher your penalty, the simpler final model

Careful about underfitting

**Inner work of Lasso regression:**

1. Start with linear regression: minimize prediction errors
2. Add a penalty: shrink large coefficient down
3. Penalty strength controlled by parameter alpha
4. Lasso forces irrelevant features coefficients to zero
5. Important features stay unimportant ones are removed
6. Final model is simpler and avoids overfitting.

**Finding the best alpha:**

**Two methods:**

* 1. Visually identify it by drawing a plot

Loop over values of alpha, pick the best one

More intuitive but hyper specific

* 1. Using grid search cross validation

Just call a method

Pass on parameter

**Why Use Lasso Regression?**

1. **Feature Selection**:
   * Lasso can shrink some coefficients to **exactly zero**, effectively removing those features from the model.
   * Useful when you have many features, but only a few are important.
2. **Prevents Overfitting**:
   * Reduces the complexity of the model by penalizing large coefficients.
3. **Improves Interpretability**:
   * By reducing the number of features, it simplifies the model and makes it easier to interpret.

**Advantages of Lasso**

1. Simultaneously performs feature selection and regularization.
2. Reduces model complexity and prevents overfitting.
3. Improves interpretability by excluding irrelevant features.

**Limitations of Lasso**

1. **Multicollinearity**:
   * Lasso may arbitrarily select one feature out of a group of correlated features.
2. **Performance with Small λ\lambda**:
   * When λ\lambda is small, Lasso behaves like ordinary linear regression.
3. **High Dimensionality**:
   * May not work well when the number of features exceeds the number of samples.

**Supervised and Unsupervised:**

**Supervised:** A type of learning where the model is trained on labeled data, meaning each input comes with a corresponding target/output.

**Unsupervised:** A type of learning where the model is trained on unlabeled data, meaning no specific output is provided.

|  |
| --- |
| **Comparison Table** |
| |  |  |  | | --- | --- | --- | | **Aspect** | **Supervised Learning** | **Unsupervised Learning** | | **Data Type** | Labeled | Unlabeled | | **Goal** | Predict outputs | Identify patterns | | **Examples** | Regression, Classification | Clustering, Dimensionality Reduction | | **Algorithms** | Linear Regression, Decision Trees | K-Means, PCA, Hierarchical Clustering | | **Output** | Known during training | Unknown during training | |

**QUESTIONS:**

* + 1. **What are the limitations of linear regression, and when should you avoid using it?**

Limitations:

* + - 1. Assumes a linear relationship between independent and dependent variables.
      2. Sensitive to outliners, which can skew results.
      3. Assumes no multicollinearity among independent variables.
      4. Does not perform well non-linear data.

When to avoid:

1. When the relationship between variables is not linear.
2. When data contains significant outliners.
3. When there is high multicollinearity or heteroscedasticity.
   * 1. **How would you explain the formula of linear regression in both one-dimensional and multi-dimensional cases?**

One-dimensional:

Y=*B0+b1x+€*

Here, y is the dependent variable, x is the independent variable, b0 is the intercept, b1 is the slope and € is the error term.

Multi-dimensional:

y=B0+B1x+B2x2+-----+BnXn+€

The formula expands to include multiple independent variables x1, x2, x3 and

* + 1. **How do you choose the polynomial degree for your regression model?**

**.** Use cross-validation to evaluate model performance for different degrees.

**.** Start with a lower degree and incrementally increase it, monitoring for overfitting.

**.** Use domain knowledge to set a reasonable maximum degree.

**.** Evaluate using metrics like Mean squared error (MSE) or R-Squared.

* + 1. **Can you explain what logistic regression is and when or how to is used, including its limitations?**

Definition:

Logistic regression is used for binary or multi-class classification problems. It models the probability of a categorical outcome using a sigmoid function

Application: Spam detection, disease diagnosis, credit risk assessment.

Limitations:

. Assumes linear separability in the feature space.

. Sensitive to outliers.

. Limited to classification problems and does not provide continuous predictions.

* + 1. **Can you explain what recall is and why it is important?**

**Recall:**

Recall =Ture Positives/True Positives + False Negative

It measures the proportion of actual positive correctly identified.

Importance:

Crucial in applications were missing a positive case it costly, such as disease or fraud detection.

* + 1. **What is a confusion matrix, and how does it work in classification problems?**

**.** A confusion Metrix is a table summarizing the performance of a classification model by showing True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN)

It helps calculate metrics like accuracy, precision, recall and F1 Score.

* + 1. **What are some tweaks or improvement that can be made to a logistic regression model, and how do they work?**

**Feature Engineering**: Create or transform features to capture complex relationships.

**Feature Scaling**: Standardize inputs for faster convergence.

**Regularization:** Use L1 and L2 to reduce overfitting.

**Adjust Decision Threshold:** Change the default 0.5 threshold to suit your needs (e.g. improve recall)

**Cross-Validation:** Remove irrelevant or redundant features to reduce noise.

**Data Argumentation:** generates additional data for small or imbalanced datasets.

**Outlier Handling:** Address outliers to improve robustness.

* + 1. **which classification method is the most prone to overfitting, and why?**

High Variance: Decision trees tend to memorize the training data by creating

very specific splits, especially if the grows too deep.

Complexity: They capture every detail of the training data, including noise leading to poor generalization on unseen data.

Small data risk: With limited data, the tree can create overly specific patterns that don’t apply to new data.

Solution: Use pruning, limit tree depth or employ ensemble methods like random forests or gradient boosting to mitigate overfitting.

* + 1. **How does a decision trees model differ from a logistic regression model? when would you use which?**

Decision trees: No-linear, prone to overfitting, handles complex relationships.

Logistic Regression: Linear, less prone to overfitting, interpretable coefficients.

When to use:

Decision trees: For non-linear data or interactions.

Logistic Regression: For linear relationships or when simplicity is key.

* + 1. **what method or alternative models can address the overfitting problem in decision trees?**

Methods to address Overfitting in decision trees:

Pruning: limit tree depth or remove branches based on performance.

Regularization: Set constraints like maximum depth or minimum samples per leaf.

Ensemble Methods: Use Random Forest or gradient Boosting to combine multiple trees for better generalization.

Cross-Validation: Tune hyperparameters to balance complexity and accuracy.

Add Noise: Randomize data splits slightly to reduce overfitting.

* + 1. **How can the KNN model overfit or underfit, and what factors contribute to these issues?**

Overfitting KNN:

. Happens when k is too small, making the model overly sensitive to noise in the training data.

Contributing Factors: low values of k, high dimensionality, and noise data.

Underfitting in KNN:

. Happens when k is too large, causing the model to oversimplify by averaging too many neighbors.

Contributing Factors: High value of k and insufficient features to distinguish classes.

Solutions: Use cross-validation to find the optimal value of k and scale features properly.

* + 1. **what is cross-validation and how does it help in KNN?**

**Cross-Validation:**  
It is a technique to split the dataset into training and validation sets multiple times to evaluate model performance.

**How it Helps in KNN:**

1. **Optimal k Selection:** Determines the best number of neighbors by testing multiple values.
2. **Prevents Overfitting/Underfitting:** Ensures the model generalizes well to unseen data.
3. **Reliable Evaluation:** Reduces bias in performance estimates by using all data for both training and validation.
   * 1. **Compare hierarchical clustering and k-means when would you see each?**

**Comparison:**

1. Algorithm Type:
   * Hierarchical Clustering: Builds a tree-like structure (dendrogram) by merging or splitting clusters.
   * K-Means: Divides data into a predefined number of clusters by minimizing distance to centroids.
2. Cluster Shape:
   * Hierarchical: Handles non-spherical clusters well.
   * K-Means: Works best with spherical, evenly sized clusters.
3. Scalability:
   * Hierarchical: Computationally expensive, better for small datasets.
   * K-Means: Scales well for large datasets.
4. Number of Clusters:
   * Hierarchical: No need to specify clusters beforehand.
   * K-Means: Requires predefining the number of clusters.

When to Use:

* Hierarchical Clustering: For small datasets or when you want to visualize cluster hierarchy (e.g., exploratory analysis).
* K-Means: For large datasets or when you know the approximate number of clusters.
  + 1. **How do I determine the best polynomial degree here?**

To determine the best polynomial degree, we can use cross-validation and evaluate the model's performance (e.g., using mean squared error or R-squared) for different degrees. The best degree is typically the one that minimizes error without overfitting, which is often identified when the model starts to perform poorly on the validation data compared to the training data.