TOP SELLER

YOUR GUIDE TO ARTIFICIAL INTELLIGENCE & DATA SCIENCE

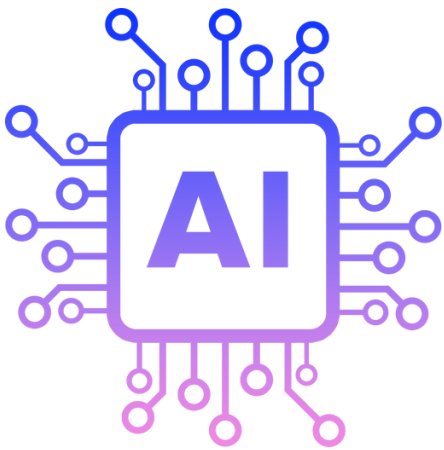
Written By
Mohammad Arshad

DECODING DATA SCIENCE

Table of Contents

Introduction

Welcome to "Your Guide to Artificial Intelligence & Data Science," an all-encompassing manual created to guide you through the intricacies and nuances of Artificial Intelligence (AI) and Data Science. This eBook is an indispensable resource for learners, professionals, and entrepreneurs keen on leveraging the transformative capabilities of AI and Data Science, offering both foundational knowledge and advanced insights.



Objective

The primary aim of this guide is to simplify and unravel the often-intimidating terminologies and concepts associated with AI and Data Science. Whether you are an aspiring data analyst, a machine learning enthusiast, or a decision-maker in your organization, this eBook equips you with the conceptual and practical tools necessary for a fulfilling journey in this ever-evolving field.

Structure and Content
The eBook is organized into the following sections, each designed to offer in-depth knowledge and actionable insights:

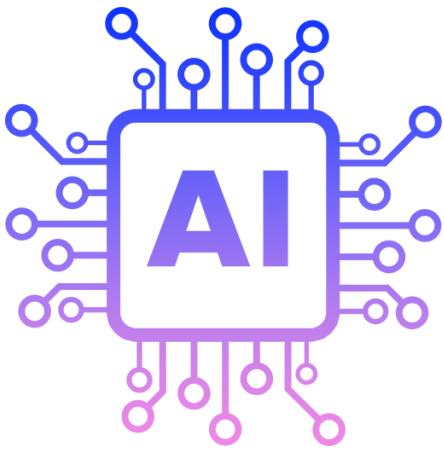
- **Roadmap to get into Artificial Intelligence: Setting the stage for your AI journey.**
- **Numpy and Pandas Cheat Sheets: Quick-reference guides to these essential libraries.**
- **Loc and Iloc in Panda: A focused look at data selection techniques in Pandas.**
- **Data Visualization with Seaborn and Matplotlib Cheat Sheet: Mastering the art of visualizing data.**
- **Regression Analysis and Logistic Regression Explained: Fundamental statistical methods for predictive modeling.**
- **Introduction to Decision Tree and KMeans Cheat Sheet: Exploring classification and clustering algorithms.**
- **Hypothesis Testing: A primer on inferential statistics in Data Science.**
- **Data Science Roadmap: A comprehensive path for aspiring data scientists.**
- **SQL Operators and SQL Interview: Preparing you for data manipulation and interviews.**
- **DAX in PowerBI: A guide to Data Analysis Expressions in Power BI.**
- **Roadmap to Mastering Generative AI: Charting the course for your foray into Generative AI.**
- **NLP Cheat Sheet: A quick guide to Natural Language Processing.**
- **Parameters of OpenAI: Understanding the mechanics of OpenAI platforms.**
- **AIGuild Premium Community: An invitation to join our exclusive community for further learning and networking opportunities.**

Audience

Whether you're a student just starting out, a professional looking to pivot into AI and Data Science, or a C-level executive keen on leveraging these technologies for business advancement, this eBook is tailored for you.

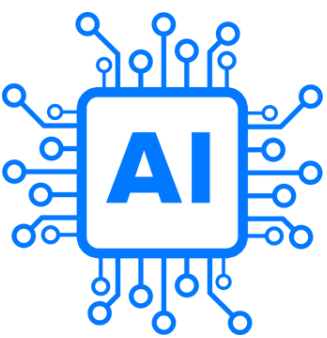
Concluding Remarks

We stand at an unprecedented crossroads in technological advancement. AI and Data Science are more than just buzzwords; they are transformative forces capable of redefining industries, economies, and lives.



Thank you for choosing this eBook as your companion in this intellectual expedition. We trust it will be a valuable asset on your path to mastering Artificial Intelligence and Data Science.

Roadmap to Get into Artificial Intelligence



6 Steps to Master

1. **Excel**
2. **PowerBI**
3. **SQL for Data Science**
4. **Python for Data Science**
5. **Maths & Stats for DS**
6. **Capstone Project**

Master Excel

Mastering Excel for Data Science involves learning how to effectively use Excel as a tool to collect, clean, analyze, and visualize data.

PowerBI

Power BI is an important tool for Data Science that offers a wide range of benefits.

It helps Data Scientists to collect, manage, and visualize data, making it easier to communicate insights to stakeholders and drive business decisions

SQL For DS

SQL (Structured Query Language) is a programming language that is widely used for Data Science. It is used to manipulate and retrieve data stored in relational databases, making it an essential tool for data analysis

SQL is a fundamental skill that is necessary for anyone working in Data Science. Here are some of the key benefits of SQL for Data Science

Python For DS

Python is a popular programming language used extensively in the field of Data Science.

It is a versatile language that offers a wide range of libraries and tools for Data Scientists to analyze, manipulate and visualize data

Panda , Numpy & Matplotlib

Maths/Stats For DS

Mathematics and statistics are fundamental to the field of Data Science.

They provide the foundation for data analysis and modeling, enabling Data Scientists to draw insights from data and make informed decisions

Capstone Project

Project work is an essential part of Data Science education.

Projects help Data Science students to apply the concepts they have learned in a practical setting, and gain hands-on experience working with real-world datasets.

Projects provide a platform for Data Science students to showcase their skills, and help them to build a strong portfolio of work.

NUMPY

Cheat Sheet

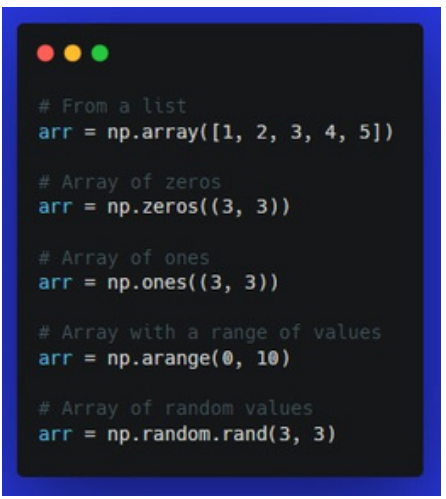
1. Basic Commands

Importing NumPy and checking its version:

```
import numpy as np

print(np.__version__)
```

2. Array Creation

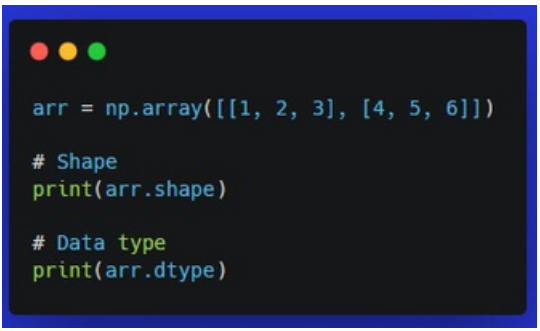
Creating NumPy arrays from lists and with initial placeholders:



```
# From a list
arr = np.array([1, 2, 3, 4, 5])

# Array of zeros
arr = np.zeros((3, 3))

# Array of ones
arr = np.ones((3, 3))

# Array with a range of values
arr = np.arange(0, 10)

# Array of random values
arr = np.random.rand(3, 3)
```

3. Array Attributes

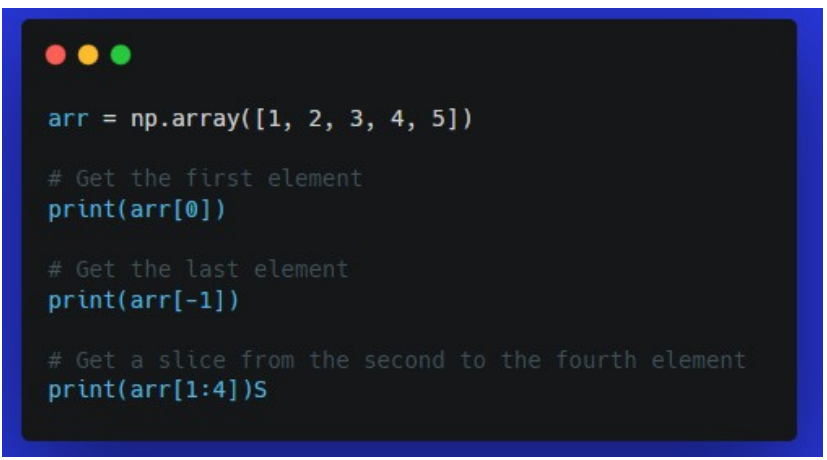
Getting an array's shape and data type:



```
arr = np.array([[1, 2, 3], [4, 5, 6]])

# Shape
print(arr.shape)

# Data type
print(arr.dtype)
```

4. Indexing and Slicing

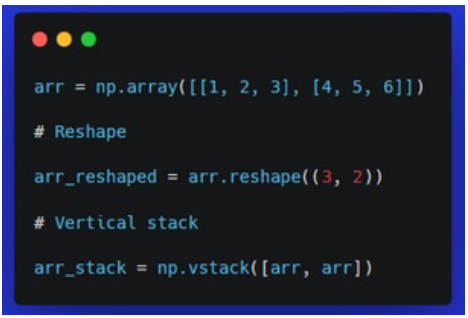
Indexing and slicing one-dimensional and multi-dimensional arrays:



```
arr = np.array([1, 2, 3, 4, 5])

# Get the first element
print(arr[0])

# Get the last element
print(arr[-1])

# Get a slice from the second to the fourth element
print(arr[1:4])S
```

5. Array Manipulation

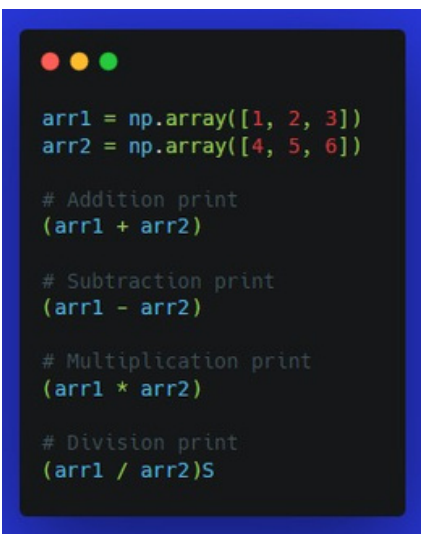
Various ways to manipulate arrays such as reshaping, stacking, and splitting:



```
arr = np.array([[1, 2, 3], [4, 5, 6]])

# Reshape

arr_reshaped = arr.reshape((3, 2))

# Vertical stack

arr_stack = np.vstack([arr, arr])
```

6. Arithmetic Operations

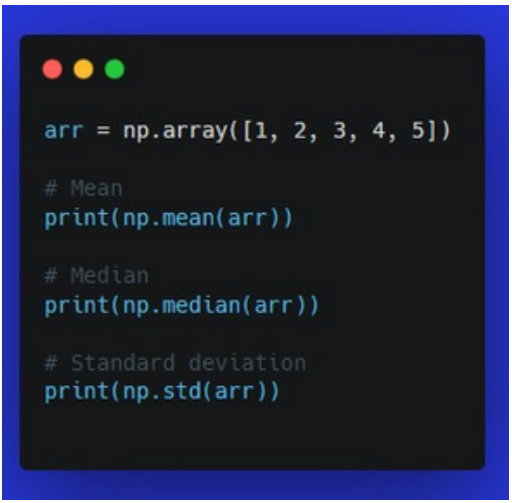
Performing addition, subtraction, multiplication, division, and dot product on arrays:



```
arr1 = np.array([1, 2, 3])
arr2 = np.array([4, 5, 6])

# Addition print
(arr1 + arr2)

# Subtraction print
(arr1 - arr2)

# Multiplication print
(arr1 * arr2)

# Division print
(arr1 / arr2)S
```

7. Statistical Operations

Calculating the mean, median, and standard deviation of an array:



PANDAS
Cheat Sheet

1. Basic Commands

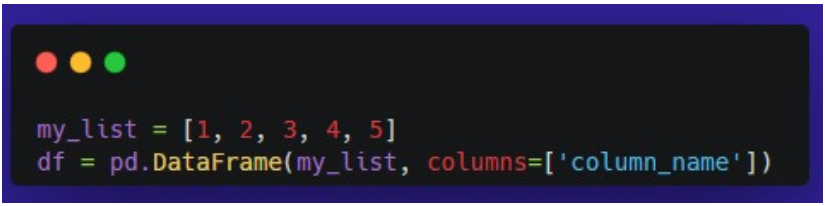
Pandas is a software library for Python that provides tools for data manipulation and analysis. It's important to ensure that the correct version of pandas is installed for compatibility with your code.

- Importing Pandas:

- Checking Pandas Version:

2. Dataframe Creation

Dataframes are two-dimensional labeled data structures with columns potentially of different types.
You can think of it like a spreadsheet or SQL table.
- From a list:



- From a Dictionary:

3. Data Selection

Pandas provides different methods for data selection.
- Selecting a column:

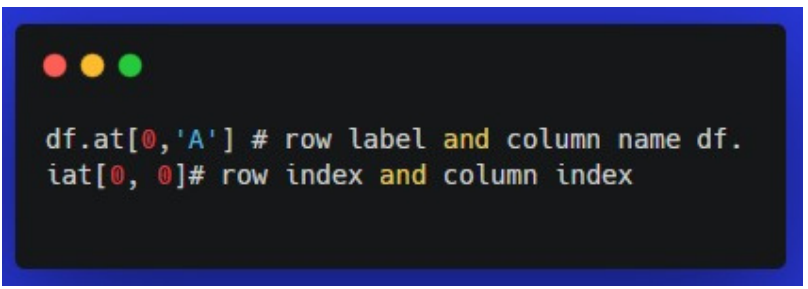
- Selecting multiple columns:

```
df[['A','B']]
```

- Selecting rows:

```
df.loc[0] # row label
df.iloc[0] # row index
```

- Selecting specific value:



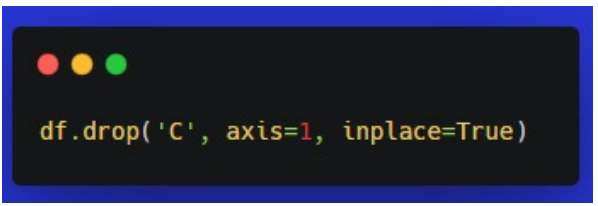
```
df.at[0,'A'] # row label and column name df.
iat[0, 0]# row index and column index
```

4. Data Manipulation

Pandas provide various ways to manipulate a dataset.

- Adding a column:

- Deleting a column:



- Renaming columns:

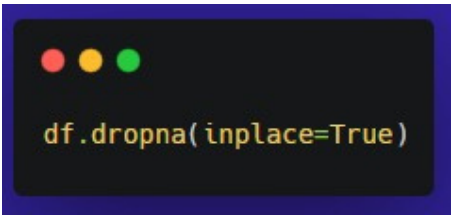
- Applying a function to a column:

5. Data Cleaning

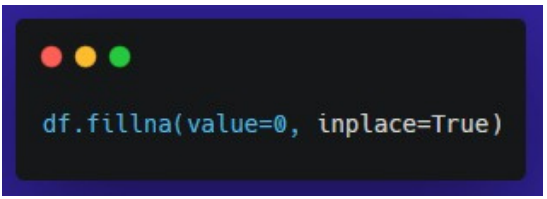
Data cleaning is detecting and correcting (or removing) corrupt or inaccurate records from a dataset.

- Checking for null values:

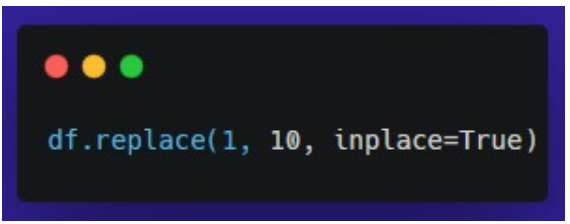
- Dropping null values:



- Filling null values:



- Replacing values:

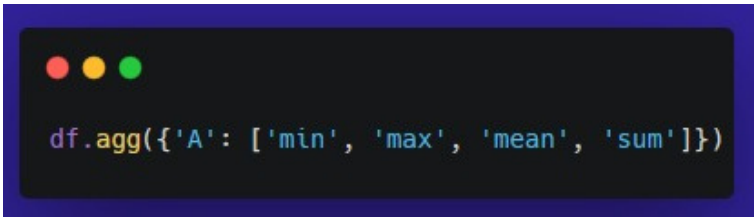


6. Grouping & Aggregation

Grouping involves combining data based on some criteria, while aggregation is the process of turning the results of a query into a single row.

- Group by:

- Aggregation:

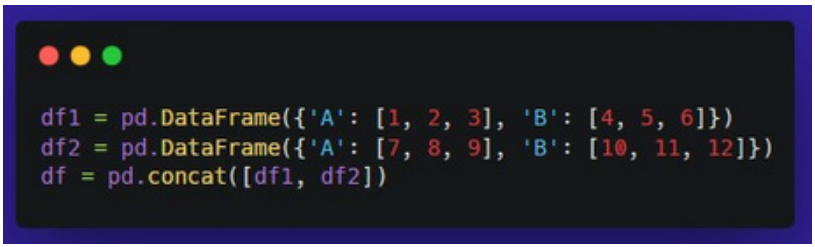


```
df.agg({'A': ['min', 'max', 'mean', 'sum']})
```

7. Merging, Joining, and Concatenating

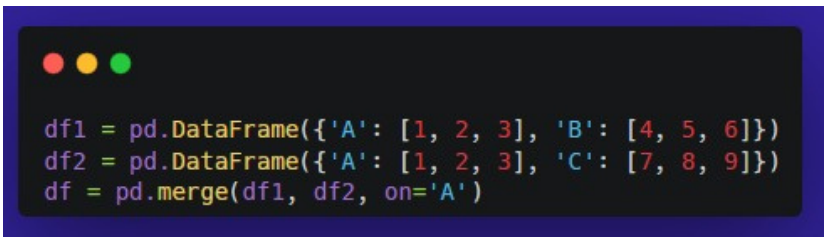
Pandas provides various ways to combine DataFrames including merge and join.

- Concatenating:



```
df1 = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
df2 = pd.DataFrame({'A': [7, 8, 9], 'B': [10, 11, 12]})
df = pd.concat([df1, df2])
```

- Merging:



```
df1 = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
df2 = pd.DataFrame({'A': [1, 2, 3], 'C': [7, 8, 9]})
df = pd.merge(df1, df2, on='A')
```

- Joining:

```
df1 = pd.DataFrame({'A': [1, 2, 3], 'B': [4, 5, 6]})
df2 = pd.DataFrame({'C': [7, 8, 9]})
df = df1.join(df2)
```

8. Working with Dates

Pandas provides powerful functionalities for working with dates.

- Convert to datetime:

```
df['date'] = pd.to_datetime(df['date'])
```

- Extracting date parts:

```
df['year'] = df['date'].dt.year
df['month'] = df['date'].dt.month
df['day'] = df['date'].dt.day
```

9. File I/O

Pandas can seamlessly read from and write to a variety of file formats.

- Reading a CSV file:

```
df = pd.read_csv('file.csv')
```

- Writing to a CSV file:

```
df.to_csv('file.csv', index=False)
```

- Similarly for other file formats like

```
Excel (read_excel, to_excel), JSON (read_json, to_json), SQL (read_sql, to_sql), etc.
```

ILOC AND LOC IN PANDA

- Loc allows you to select data based on the label or name of the row or column, while iloc uses the number or index of the row or column.

- Understanding the difference between these two methods is crucial for effectively working with data in Python.

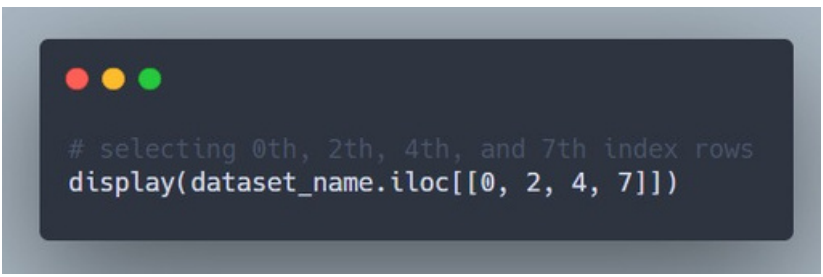
- Whether you're a seasoned data analyst or just starting out, understanding how to use loc and ilocwill give you the skills you need to effectively analyze data in Python.

LOC

THE LOC() FUNCTION IS LABEL BASED DATA SELECTING METHOD WHICH MEANS THAT WE HAVE TO PASS THE NAME OF THE ROW OR COLUMN WHICH WE WANT TO SELECT.

ILOC

THE ILOC() FUNCTION IS AN INDEXED-BASED SELECTING METHOD WHICH MEANS THAT WE HAVE TO PASS AN INTEGER INDEX IN THE METHOD TO SELECT A SPECIFIC ROW/COLUMN.



Python Pandas Selections and Indexing

.iloc selections - position based selection

data.iloc[<row selection], <column selection>]

Integer list of rows: [0,1,2]
Slice of rows: [4:7]
Single values: 1

Integer list of columns: [0,1,2]
Slice of columns: [4:7]
Single column selections: 1

loc selections - position based selection

data.loc[<row selection], <column selection>]

Index/Label value: 'john'
List of labels: ['john', 'sarah']
Logical/Boolean index: data['age'] == 10

Named column: 'first_name'
List of column names: ['first_name', 'age']
Slice of columns: 'first_name':'address'

	loc[] - By Label	iloc[] - By Index
Select Single Row	df.loc['r2']	df.iloc[1]
Select Single Column	df.loc[:, "Courses"]	df.iloc[:, 0]
Select Multiple Rows	df.loc[['r2','r3']]	df.iloc[[1,2]]
Select Multiple Columns	df.loc[:, ["Courses","Fee"]]	df.iloc[:, [0,1]]
Select Rows Range	df.loc['r1':'r4']	df.iloc[0:4]
Select Columns Range	df.loc[:,'Fee':'Discount']	df.iloc[:,1:4]
Select Alternate Rows	df.loc['r1':'r4':1]	df.iloc[0:4:1]
Select Alternate Columns	df.loc[:,'Fee':'Discount':1]	df.iloc[:,1:4:1]
Using Condition	df.loc[df['Fee'] >= 24000]	df.iloc[list(df['Fee'] >= 24000)]
Using Lambda Function	df.loc[lambda x: x[3]]	df.iloc[lambda x: x[3]]

Difference Between pandas DataFrame loc vs iloc

Data Visualization with Seaborn

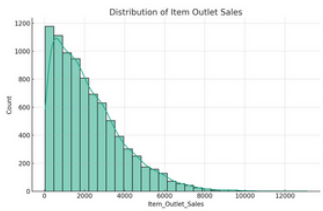
Introduction to Seaborn, its various functionalities, and sample graphs using the provided dataset.

2. Distribution Plots

Description:
Distribution plots are used to visualize the distribution of a dataset. Common distribution plots in Seaborn include the histogram, KDE (Kernel Density Estimation), and the rug plot.

Code Snippet:

```
# Create a simple distribution plot
data = sns.load_dataset('tips')
sns.histplot(data['total_bill'])
plt.show()
```

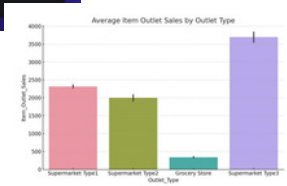


3. Categorical Plots

Description:
Categorical plots are used to visualize categorical data. Examples include bar plots, box plots, and violin plots.

Code Snippet:

```
# Create a simple bar plot
sns.barplot(x='day', y='total_bill', data=data)
plt.show()
```

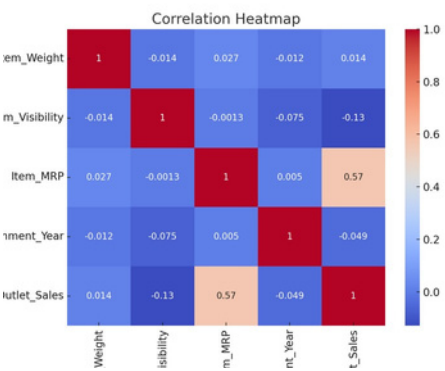


4. Matrix Plots

Description:
Matrix plots are used to display data in a matrix format.
The heatmap is a common matrix plot used to represent data in a color-encoded matrix format.

Code Snippet:

```
# Create a heatmap of a correlation matrix
correlation = data.corr()
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.show()
```

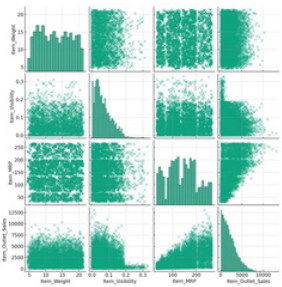


5. Pair Plots

Description:
Pair plots are used to visualize relationships between multiple variables in a dataset.
It plots pairwise relationships in a dataset.

Code Snippet:

```
# Create a pair plot
sns.pairplot(data)
plt.show()
```

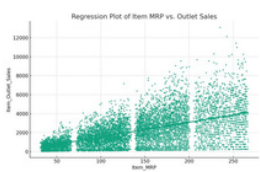


6. Regression Plots

Description:
Regression plots are used to visualize the relationship between two variables and fit a regression line.

Code Snippet:

```
# Create a regression plot
sns.regplot(x='total_bill', y='tip', data=data)
plt.show()
```

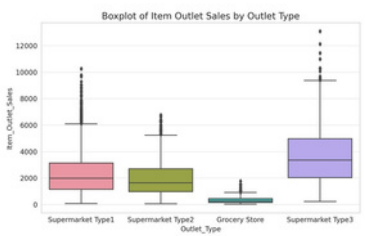


7. Styling and Themes

Description:
Seaborn allows for the customization of plots using various styles and themes.
This ensures that plots are both informative and aesthetically pleasing.

Code Snippet:

```
# Set a theme and create a plot
sns.set_style('whitegrid')
sns.barplot(x='day', y='total_bill', data=data)
plt.show()
```



MATPLOTLIB
Cheat Sheet

1. Basic Commands

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
- Importing Matplotlib:

```
import matplotlib.pyplot as plt
```

- Checking Matplotlib version:

```
import matplotlib
print(matplotlib.__version__)
```

2. Basic Plotting

Matplotlib provides functionalities for various types of plots.

```
- Line Plot: plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
- Scatter Plot: plt.scatter([1, 2, 3, 4], [1, 4, 9, 16])
- Bar Plot: plt.bar(['group_a', 'group_b', 'group_c'], [1, 10, 5])
- Histogram: plt.hist([1, 2, 2, 3, 4, 4, 4, 5, 5, 5, 5])
```

3. Figure and Axes

A figure in matplotlib means the whole window in the user interface. Axis are the number-line-like objects and they take care of generating the graph limits.

- Creating Figure and Axes:

```
fig, ax = plt.subplots()
```

- Setting Figure Size:

```
fig, ax = plt.subplots()
```

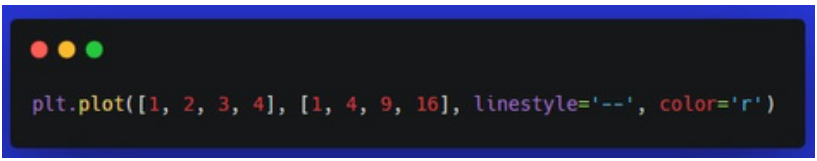
-Setting Axis Labels and Title:

```
ax.set_xlabel('x')
ax.set_ylabel('y')
ax.set_title('Title')
```

4. Customizing Plots

Matplotlib allows you to customize various aspects of your plots.

- Changing Line Style and Color:

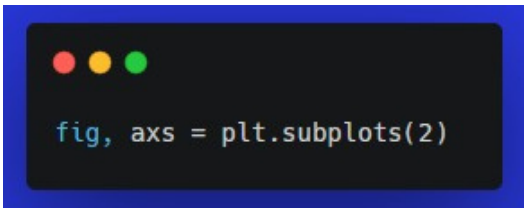


- Adding Grid:

- Setting Axis Limits:

5. Multiple Plots

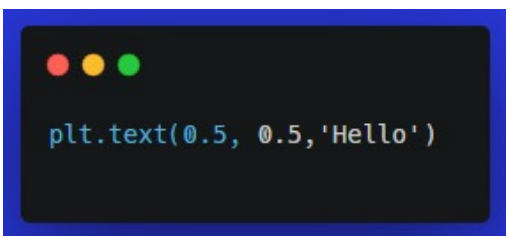
Matplotlib provides functionalities to create multiple plots in a single figure.
- Subplots:



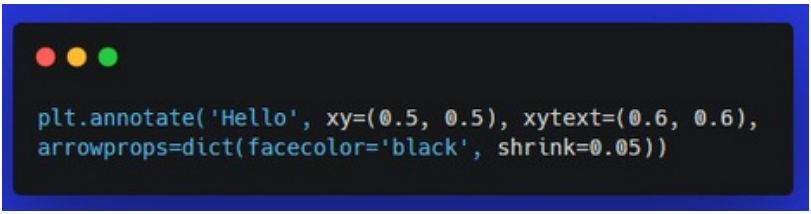
- Sharing Axis:

6. Text and Annotations

Matplotlib provides functionalities to add text and annotations to the plots.
- Adding Text:



- Adding Annotations:



7. Saving Figures

Matplotlib provides the savefig() function to save the current figure to a file.
- Saving Figures as PNG, PDF, SVG, and more:

REGRESSION ANALYSIS

Agenda

- **Introduction to Regression Analysis**
  - **–What is Regression Analysis**
  - **–Why do we need Regression Analysis in Business**
  - **– Introduction to Modeling**
- **Introduction to OLS Regression**
- **Introduction to Modeling Process**

What is Regression Analysis?

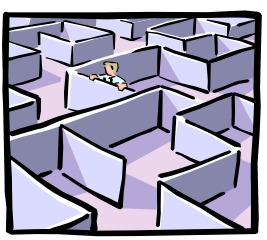
Regression Analysis captures the relationship between one or more response variables (dependent/predicted variable –denoted by Y) and the its predictor variables (independent/explanatory variables –denoted by X) using historical observations of both.

Hence its estimates the functional relationship between a set of independent variables X1, X2, ..., Xpwith the response variable Y which estimate of the functional form best fits the historical data.

```
Y = f (X1, X2,.., Xp) + Є
where Є denotes the "Residual" or unexplained part of Y
```

Historical Data

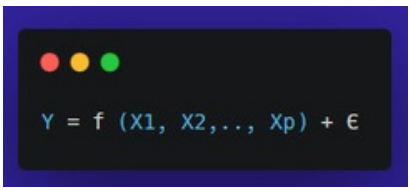
Statistical Analy ses



Predict Future Events

Types of Regression Analysis

There are various kinds of Regressions based on the nature of : - •the functional form of the relationship
•the residual
•the dependent variable
•the independent variables



Functional Form	Residual	Dependent Var	Independent Var
▪ Linear ▪ Non-Linear –*Out of scope for this presentation*	• Based on the distribution of the residual –normal, binomial, poisson, exponential	• Single • Continuous • Discrete • Binary • Multiple –*Out of scope for this presentation*	• Numerical • Discrete • Continuous • Categorical • Ordinal • Nominal

Types of Linear Regression

Dependent Variable Type	Residual Distribution	Types of Regression
Continuous	Normal (with constant variance)	Ordinary Least Squares (OLS)
Continuous	Normal (without constant variance)	Generalized Least Square
Binary	Binomial	Logistic Regression
Discrete	Poisson	Poisson Regression
Rational	Exponential Family of Distributions	Generalized Least Squares

Other Types of Regression Related Techniques

- **Simultaneous Equation Models**

–When both X & Y are dependent on each other

- **Structural Equation Modeling / Pathways**

–Captures the inter-relations between Xs i.e. captures how Xs affect each other before affecting Y

- **Survival Analysis**

–Predicts a decay curve for a probability of an event

- **Hierarchal Bayesian**

–Estimates a non-linear equation

Agenda

- **Introduction to Regression Analysis**

–What is Regression Analysis

–Why do we need Regression Analysis in Business – Introduction to Modeling

- **Introduction to OLS Regression**
- **Introduction to Modeling Process**

What is Modeling?

Is based on Regression Analysis. It can be used for the following two distinct but related purposes

1. **Predict certain events**
2. **Identify the drivers of certain events based on some explanatory variables**

Isolates individual effects and then quantifies the magnitude of that driver to its impact on the dependent variable. It is required because

1. **Knowledge of Y is crucial for decision making but is not deterministic**
2. **X is available at the time of decision making and is related to Y**

Volume = Base Sales + b2 (GRPs) + b3 (Dist) ... + bn (Price)

Example of Modeling in Business

- **Predict the sales that a customer would contribute, given a certain set of attributes like demographic information, credit history, prior purchase behavior, etc.**
- **Predict the probability of response from a direct mail thus saving cost and acquire potential customers.**
- **Identify high responsive and high profit segments and targeting only these segments for direct mail campaigns**
- **Identify the most effective marketing levers & quantify their impact**
- **To find out what differentiates between buyers and non buyers based on their past 3 months usage of the product and the age group**

Agenda

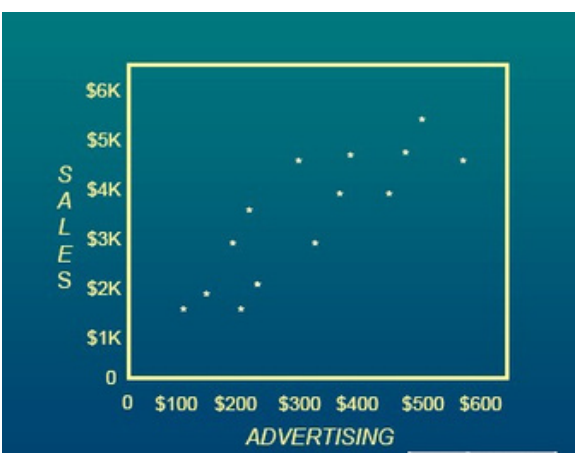
- Introduction to Regression Analysis
- Introduction to OLS Regression
- Introduction to Modeling Process

Introduction to Ordinary Least Squares

Dependent Variable Type	Residual Distribution	Types of Regression
Continuous	Normal (with constant variance)	Ordinary Least Squares (OLS)
Continuous	Normal (without constant variance)	Generalized Least Square
Binary	Binomial	Logistic Regression
Discrete	Poisson	Poisson Regression
Rational	Exponential Family of Distributions	Generalized Least Squares

Introduction to Ordinary Least Squares – Simple Regression

Advertising	Sales
$120	$1,503
$160	$1,755
$205	$2,971
$210	$1,682
$225	$3,497
$230	$1,998
$290	$4,528
$315	$2,937
$375	$3,622
$390	$4,402
$440	$3,844
$475	$4,470
$490	$5,492
$550	$4,398

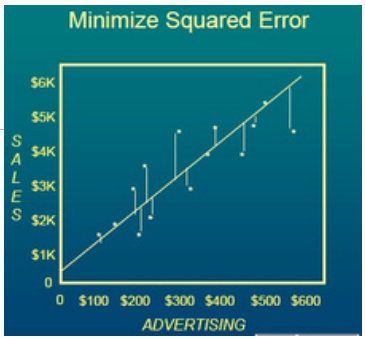


Goal: characterize relationship between advertising and sales

Introduction to Ordinary Least Squares – Simple Regression

Result: equation that predicts sales dollars based on advertising dollars spent

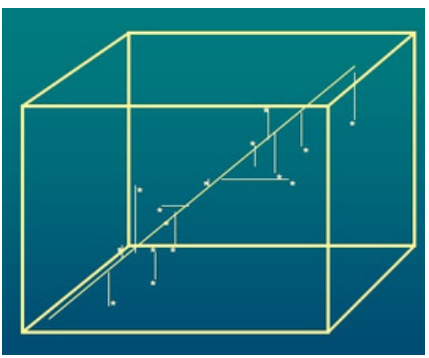
Sales = B0+ B1*Adv.



Minimizes Error sum of squares ,Hence the name "Ordinary Least Square Regression"

Introduction to Ordinary Least Squares – Multiple Regression

•Credit card balances –payment amount – years –gender (0/1) •Minimizes squared error in N-dimensional space



Balances = 2.1774 +.0966*Payment + 1.2494*Months + .4412*Gender

OLS Model Assumptions

1. Linearity -Model is linear in parameters
2. Spherical Errors - Error distribution is Normal with mean 0 & constant variance
3. Zero Expected Error - The expected value (or mean) of the errors is always zero
4. Homoskedasticity - The errors have constant variance
5. Non-Autocorrelation - The errors are statistically independent from one another. This implies the data is a random sample of the population
6. Non-Multicollinearity - The independent variables are not collinear

$Yi=a+b1X1i+b2X2i+...+bpXpi+ei$

$e2i \sim Normal(0, \sigma)$

$E(ei)=0$ for all i

$Variance(ei)=constant$ for all i

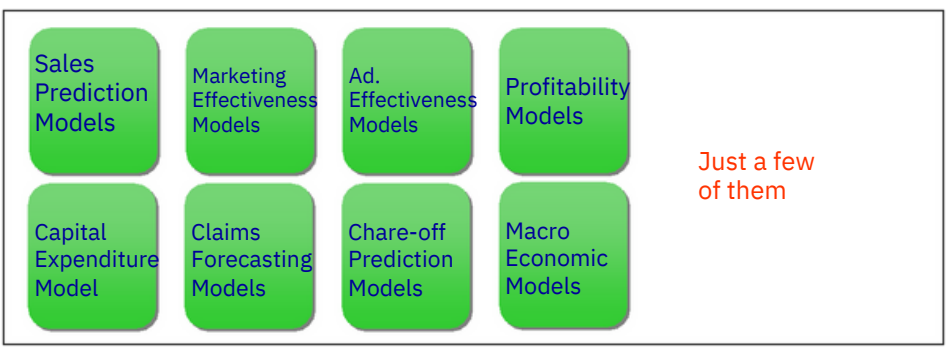
$corr(ei, ej)=0$ for all $i \neq j$

$Covariance\ (Xi,Xj) = 0$

Steps in OLS Regression

- Assume all OLS assumptions hold
- Run regression in software (R/Python)
- Check if assumptions really hold
- Check if Fit is good
- Check Hypothesis testing results i.e. variable significance
- Iterate to make “BEST” model

Applications of OLS Regression in Business



Logistics Regression Explained

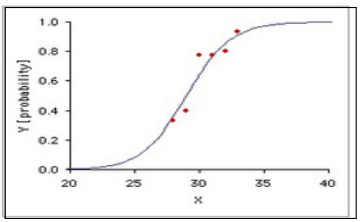
Logistic Regression – Introduction

In Linear regression, the outcome variable is continuous and the predictor variables can be a mix of numeric and categorical. But often there are situations where we wish to evaluate the effects of multiple explanatory variables on a binary outcome variable

For example, the effects of a number of factors on the development or otherwise of a disease. A patient may be cured or not; a prospect may respond or not, should we grant a loan to particular person or not, etc.

When the outcome or dependent variable is binary, and we wish to measure the effects of several independent variables on it, we uses Logistic Regression

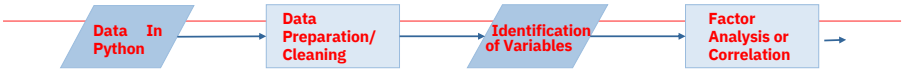
- **The binary outcome variable can be coded as 0 or 1.**
- **The logistic curve is shown in the figure below:**



We estimate the probability of success by the equation:

$$P = \frac{e^{a+bX}}{1+e^{a+bX}}$$

Process Flow

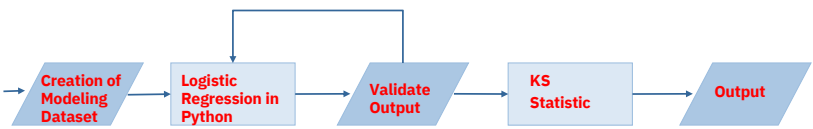


Data is obtained in pandas dataframe

- Missing Value Imputation
- Trash value
- Outlier Treatment

Independent and dependent variables should be identified

- FA is done in order to get the variables into groups
- Good to choose factor solution near the Eigen value of 1
- As a further Check Correlation Analysis is done



Divide data into Development and Validation Sample in ratio 70:30 or 80:20

- Assume all assumptions hold
- Check for the significance of the variables
- Run Regression on Development sample

- Validate the output on the Validation sample , by running the same model on the Validation sample

Results will be presented in PowerPoint

Python code

Step 1: Importing the dataset

```
dataset = pd.read_csv('car_purchase_Ads.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
```

Step 2: Splitting the dataset into the Training set and Test set

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
```

Step 3: Feature Scaling

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Step 4 : Training the Logistic Regression model on the Training set

```
from sklearn.linear_model import LogisticRegression
classifier = LogisticRegression(random_state = 0)
classifier.fit(X_train, y_train)
```

Step 5:Predicting a new result

```
print(classifier.predict(sc.transform([[30,87000]])))
```

Step 6: Predicting the Test set results

```
y_pred = classifier.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

Step 7: Making the Confusion Matrix

```
from sklearn.metrics import confusion_matrix, accuracy_score
cm =confusion_matrix(y_test, y_pred)
print(cm)
accuracy_score(y_test, y_pred)
```

Practice

For location of code and dataset

https://github.com/arshad831/Modelling-Exercise/blob/main/logistic_regression.ipynb

INTRODUCTION TO DECISION TREE

Decision Trees

A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility. It is one way to display an algorithm that only contains conditional control statements.

What is Decision Trees ?

A decision tree is a flowchart-like tree structure where an internal node represents a "test" on an attribute (e.g.whether a coin flip comes up heads or tails), each branch represents the outcome of the test, and each leaf node represents a class label (decision taken after computing all attributes).

Importance of decision tree:

Decision trees are a popular method for various classification and regression tasks. For example, in medical diagnosis, decision trees have been used to classify diseases based on symptoms. In credit scoring, decision trees are used to predict the probability of default.

Decision tree learning is a predictive modelling approach that can be used for both classification and regression tasks. The goal is to create a model that predicts the value of a target variable based on several input variables.

Decision Trees

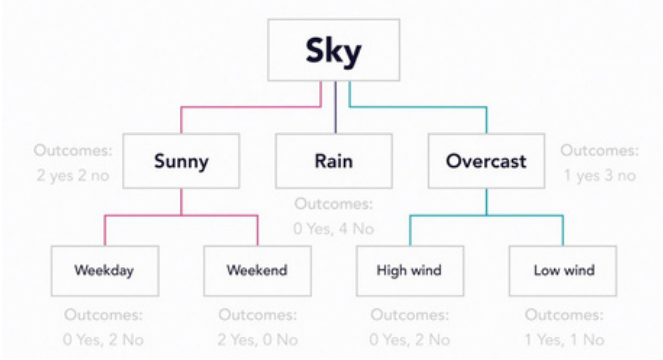
We were creating a decision tree to predict whether or not someone is likely to go to the beach.

Predictors		Outcome
Sky Weekend Wind Speed		Arshad goes to the beach?

Data

Predictors			Outcome
Sky	Weekend	Wind	Yash goes to the beach
Sunny	Weekday	Low	No
Sunny	Weekday	High	No
Overcast	Weekday	Low	Yes
Rain	Weekday	Low	No
Rain	Weekend	Low	No
Rain	Weekend	High	No
Overcast	Weekend	High	No
Sunny	Weekend	Low	Yes
Sunny	Weekend	Low	Yes
Rain	Weekend	Low	No
Overcast	Weekend	Low	No
Overcast	Weekday	High	No

Decision Tree Outcome



Advantages of decision trees:

1. Decision trees are easy to interpret and explain.
2. They can handle both numeric and categorical data.
3. They are resistance to overfitting.
4. They can be used for feature selection.
5. They are non-parametric, meaning they make no assumptions about the underlying data distribution.

Disadvantages of decision trees

1. They are prone to overfitting with large amounts of data.
2. They can be unstable because small changes in the data can result in large changes in the structure of the tree.
3. They are not as accurate as some other methods, such as neural networks.
4. They can be difficult to tune, especially when there are many parameters to consider.

6 detailed modules to be a successful data scientist

1. Excel for Data Science

- Formulas and Functions
- Visualization
- Dashboarding
- Data Analysis

2. Business Intelligence

- Tableau
- Power BI

3. Deep Dive into SQL

- Filtering Data
- Functions in Database
- Displaying Data from Multiple tables
- Grouping Data and Computing Aggregates
- Subqueries and Nested queries in SQL
- Windows Function

4. Statistics and Mathematics

- Descriptive Statistics
- Hypothesis Testing
- Fundamentals of Probability
- Linear Algebra
- ANOVA and Covariance

5. Deep Dive into Python

- Conditionals and Loops
- Operation and Operator
- Functions and Classes
- Data Wrangling – NumPy & Panda
- Visualization: Matplotlib & Seaborn

6. Decode Machine Learning

- Linear Regression
- Logistic Regression
- Decision Trees
- Clustering

K-Means Clustering Detailed Steps with Code

Introduction to K-Means Clustering

Definition and Description of K-Means Clustering: K- Means is a type of partitioning clustering that separates the data into K non-overlapping subsets (or clusters) without any cluster-internal structure.
Overview of Unsupervised Learning: In unsupervised learning, the goal is to identify useful patterns and structure from the input data. K-Means is an unsupervised learning algorithm as it forms clusters based on the input data without referring to known, or labelled, outcomes.
Use Cases for K-Means Clustering: Applications in various fields like market segmentation, image segmentation, anomaly detection, etc.

The K-Means Clustering Algorithm

Explanation of the K-Means Algorithm:
Detail the iterative process of assigning each data point to the nearest centroid, updating the centroids based on the data points assigned, and repeating until convergence. Choice of K (the number of clusters): Discuss methods to choose the optimal number of clusters, like the Elbow Method, Silhouette Analysis, etc.
Centroid Initialization Methods: Discuss different methods for initializing centroids, including random initialization, k-means++ and their impact on the final result.

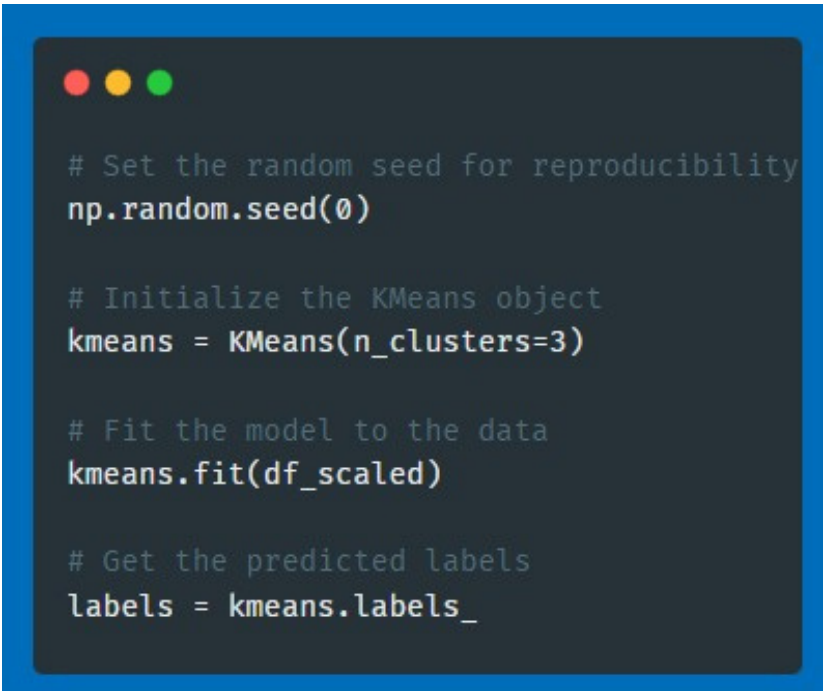
Python setup and data preparation

Required Python Libraries: Detail libraries such as pandas for data manipulation, numpy for numerical operations, matplotlib and seaborn for visualization, and scikit-learn for the K-Means algorithm. Data Preparation: Discuss the importance of data cleaning, normalization, and dealing with missing values. Include code examples of these tasks using pandas and scikit-learn.

```python
# Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import silhouette_score

# Load the data
df = pd.read_csv('data.csv')

# Data cleaning (e.g., removing duplicates)
df.drop_duplicates(inplace=True)

# Handling missing values (e.g., fill with mean)
df.fillna(df.mean(), inplace=True)

# Data normalization
scaler = StandardScaler()
df_scaled = pd.DataFrame(scaler.fit_transform(df), columns=df.columns)
```

Implementing K-Means Clustering in Python

Detailed Code Example: Provide a step-by-step walkthrough of a Python implementation of the K- Means algorithm using scikit-learn. Discuss each step in detail, including the importance of setting the random seed for reproducibility.



```
# Set the random seed for reproducibility
np.random.seed(0)

# Initialize the KMeans object
kmeans = KMeans(n_clusters=3)

# Fit the model to the data
kmeans.fit(df_scaled)

# Get the predicted labels
labels = kmeans.labels_
```

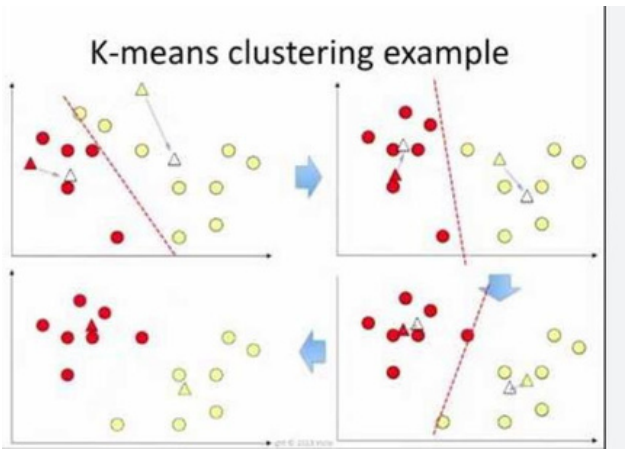
Evaluating K-Means Clustering

Evaluation Metrics: Discuss how to evaluate the clustering result using metrics like Within Cluster Sum of Squares (WCSS), between cluster sum of squares (BCSS), and silhouette score.
The Elbow Method: Explain and provide a code snippet to demonstrate the Elbow Method, a visual tool to estimate the optimal number of clusters by plotting the explained variation as a function of the number of clusters.

```
# Calculate Within Cluster Sum of Squares (WCSS)
wcss = kmeans.inertia_

# Calculate Between Cluster Sum of Squares (BCSS)
total_variance = np.sum(np.var(df_scaled))
bcss = total_variance - wcss

# Calculate silhouette score
silhouette = silhouette_score(df_scaled, labels)

# Elbow Method
wcss_values = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters=i)
    kmeans.fit(df_scaled)
    wcss_values.append(kmeans.inertia_)

plt.plot(range(1, 11), wcss_values)
plt.title('Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```

Visualizing K-Means Clustering

Visualization Techniques: Discuss and provide Python code examples for visualizing K-Means Clustering results. This could include scatter plots of the data points colored by cluster and indicating the centroids, as well as pair plots for multi-dimensional data.



Limitations and Considerations

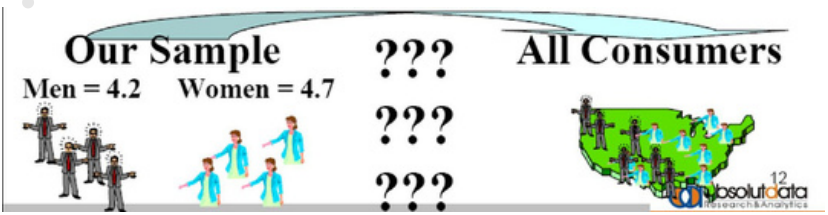
Limitations of K-Means: Discuss limitations, such as sensitivity to initialization and difficulty handling clusters of different shapes and sizes. Considerations: Discuss considerations for using K- Means effectively, such as preprocessing steps (normalization, PCA for dimensionality reduction) and the importance of understanding your data before clustering.

Hypothesis Testing

Why Significance Testing ?

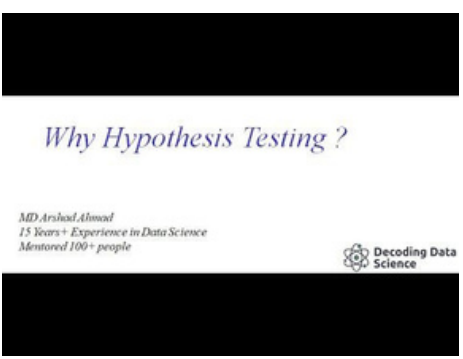
-Significance testing allows us to examine results from a sample of people, and make inferences about the total population.

-Significance tests ask: Given the differences observed in our sample, what are the odds that there are no true differences in the population?



Why Hypothesis Testing ?

https://youtu.be/hq40GN4HmP4



Video Explaining the scenarios where we need to use it

Hypothesis Testing

Medicine Experiment →

Average duration of cold = 8.5 days

n = 250

Recovery time = 7.3 days

Questions →

Is the result significant?

Could this sample just be the result of chance or did this drug have an impact?

Should the drug be tested further?

Does this mean this new drug should be approved for use?

How to test a hypothesis in four steps

Medicine Experiment →

Town population = 35,000
Percentage men = 50%
Percentage women = 50%

→ Next Sample 14 Men

Step 1a) Design Hypothesis

Step 1a) Develop Hypotheses

H_0 = Null hypothesis

Jury numbers happened by chance

Shows odds of woman being picked as 50%

H_0 is $p \leq 0.50$

H_a = Alternative hypothesis

Jury numbers not by chance

Shows odds of women being picked as higher than 50%

H_a is $p > 0.50$

Step 1b) State Significance Level

Set threshold for test

Significance level = 5%. If 36 or more women ending up on a jury have less than a 5% chance of occurring at random, then we will reject our null hypothesis.

Step 2) Identify Test Statistic

Binomial probability

p= 0.50

Step 3) Determine P-Value

Looking for the probability that the number of women chosen for the jury-panel would be 36 or more

Probability = 0.13%

Step 4) Compare P-Value to Significance Level

There was only a 0.13% chance that at random 36 or more women would be chosen for a panel of 50 potential jurors.

Alpha = 0.05 or 5%

p-value < significance level

It much more likely for a woman to be chosen versus a man.

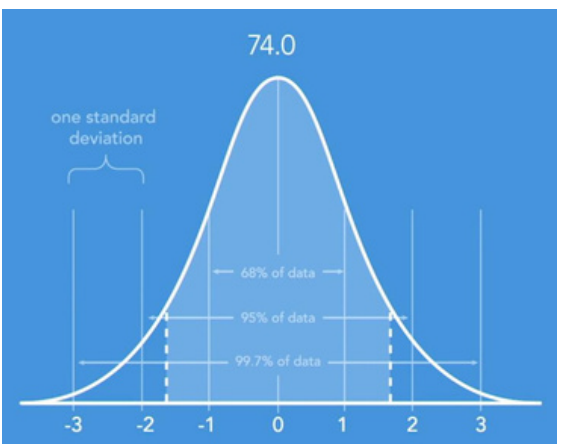
Scenarios for hypothesis testing

A recent national studyfound that the average Americanbetween the ages of 18 and 24checks their phone 74 times per day.A mobile service provider questions these results

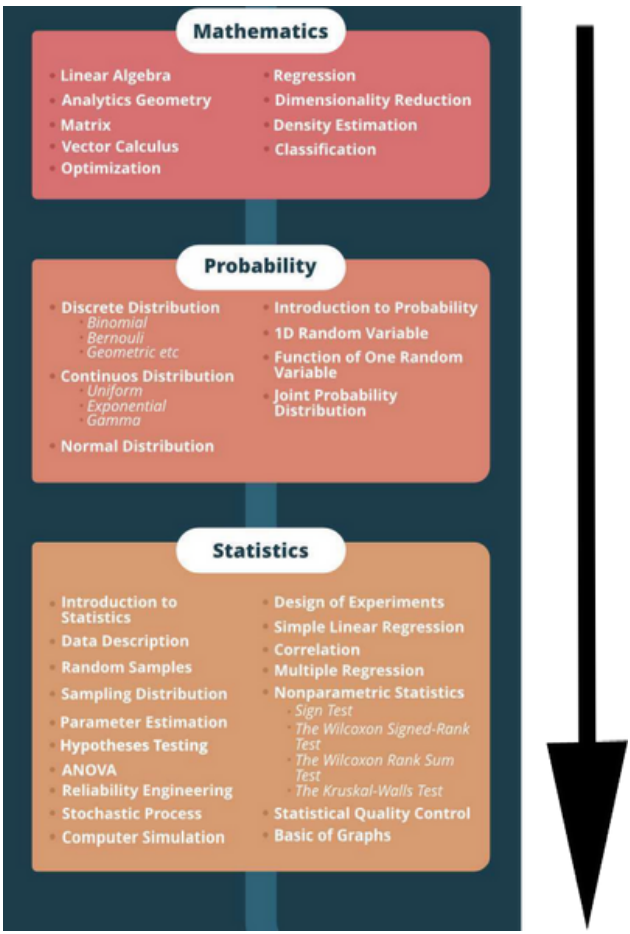
$$H_0: \mu = 74.0$$

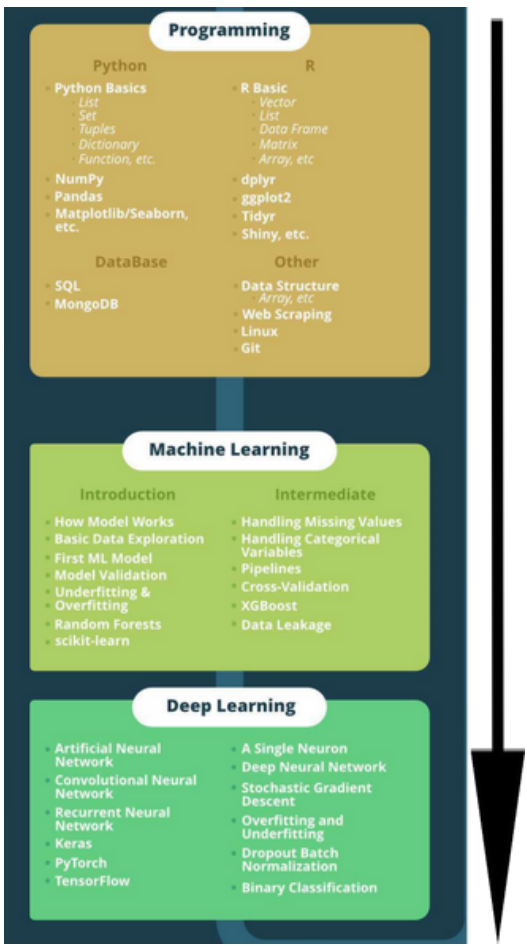
$$H_a: \mu \neq 74.0$$

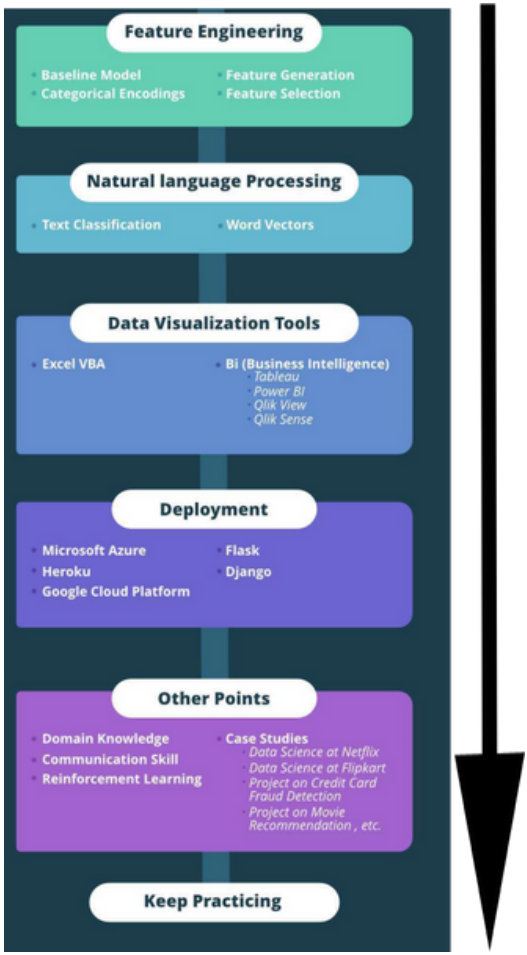
As you can see,we have two rejections areas here,one rejection area in the positive direction,greater than the mean.The other in the negative direction,less than the mean.This is considered a two tailed testbecause the null hypothesis is tested in both directions



Data Science Road Map


Mathematics
Linear Algebra
Analytics Geometry
Matrix
Vector Calculus
Optimization
Regression
Dimensionality Reduction
Density Estimation
Classification
Probability
Discrete Distribution
Binomial
Bernouli
Geometric etc
Continuos Distribution
Uniform
Exponential
Gamma
Normal Distribution
Introduction to Probability
1D Random Variable
Function of One Random Variable
Joint Probability Distribution
Statistics
Introduction to Statistics
Data Description
Random Samples
Sampling Distribution
Parameter Estimation
Hypotheses Testing
ANOVA
Reliability Engineering
Stochastic Process
Computer Simulation
Design of Experiments
Simple Linear Regression
Correlation
Multiple Regression
Nonparametric Statistics
Sign Test
The Wilcoxon Signed-Rank Test
The Wilcoxon Rank Sum Test
The Kruskal-Walls Test
Statistical Quality Control
Basic of Graphs

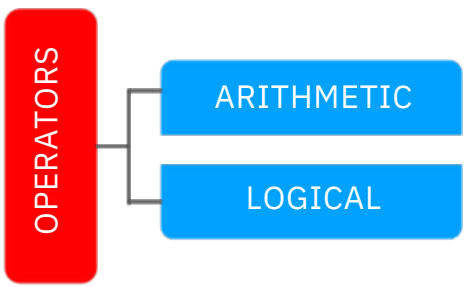

Programming
Python
Python Basics
List
Set
Tuples
Dictionary
Function, etc.
NumPy
Pandas
Matplotlib/Seaborn, etc.
R
R Basic
Vector
List
Data Frame
Matrix
Array, etc
dplyr
ggplot2
Tidyr
Shiny, etc.
DataBase
SQL
MongoDB
Other
Data Structure
Array, etc
Web Scraping
Linux
Git
Machine Learning
Introduction
How Model Works
Basic Data Exploration
First ML Model
Model Validation
Underfitting & Overfitting
Random Forests
scikit-learn
Intermediate
Handling Missing Values
Handling Categorical Variables
Pipelines
Cross-Validation
XGBoost
Data Leakage
Deep Learning
Artificial Neural Network
Convolutional Neural Network
Recurrent Neural Network
Keras
PyTorch
TensorFlow
A Single Neuron
Deep Neural Network
Stochastic Gradient Descent
Overfitting and Underfitting
Dropout Batch Normalization
Binary Classification


Feature Engineering
Baseline Model
Categorical Encodings
Feature Generation
Feature Selection
Natural language Processing
Text Classification
Word Vectors
Data Visualization Tools
Excel VBA
Bi (Business Intelligence)
Tableau
Power BI
Qlik View
Qlik Sense
Deployment
Microsoft Azure
Heroku
Google Cloud Platform
Flask
Django
Other Points
Domain Knowledge
Communication Skill
Reinforcement Learning
Case Studies
Data Science at Netflix
Data Science at Flipkart
Project on Credit Card Fraud Detection
Project on Movie Recommendation , etc.
Keep Practicing

SQL OPERATORS

SQL OPERATORS

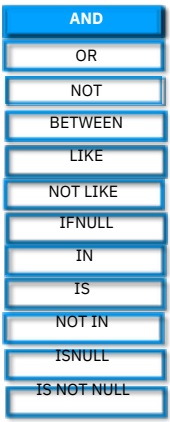
Operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement.



ARITHMETIC OPERATORS

Arithmetic Operators		Let us assume X and Y are two variables
Operator	Expression	Description
+	X + Y	To perform addition
-	X - Y	To perform subtraction
*	X * Y	To perform multiplication
/	X / Y	To perform division
%	X % Y	To perform modulus

LOGICAL OPERATORS



AND operator is generally used in the WHERE Clause to apply multiple filters on the records returned by SELECT Statement.

•This operator returns values only if all the conditions are satisfied.

Syntax:

AND()

Example:

```
SELECT First_Name
FROM customers
AND Yearly_Income> 75000 WHERE Profession LIKE '%Developer'
```

SQL

IMPORTANT INTERVIEW QUESTIONS

Answers taken from online portal for education persons .
Not for commercial use.

SQL INTERVIEW QUESTIONS

1.What is database normalization?

Ans: It is a process of analyzing the given relation schemas based on their functional
dependencies and primary keys to achieve the following desirable properties:
1)Minimizing Redundancy
2) Minimizing the Insertion, Deletion, And Update Anomalies
Relation schemas that do not meet the properties are decomposed into smaller relation
schemas that could meet desirable properties.

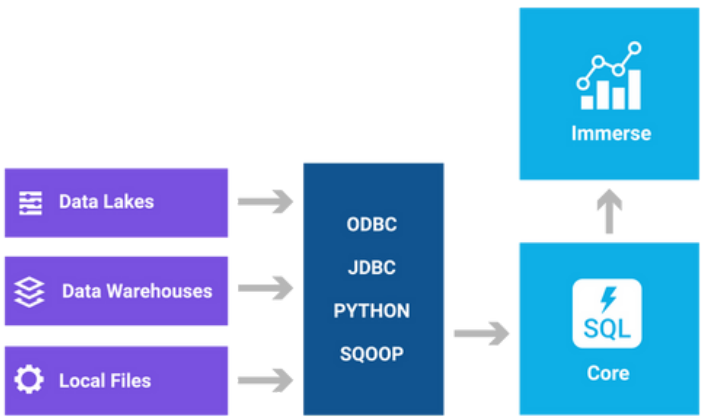
2.What is SQL?

SQL is Structured Query Language designed for inserting and modifying in a relational
database management system.

3.What are the differences between DDL, DML and DCL in SQL?

Ans: Following are some details of three.
DDL stands for Data Definition Language. SQL queries like CREATE, ALTER, DROP and
RENAME come under this.
DML stands for Data Manipulation Language. SQL queries like SELECT, INSERT and
UPDATE come under this.
DCL stands for Data Control Language. SQL queries like GRANT and REVOKE come under
this.

4.What is the difference between having and where clause? *Ans: HAVING is used to specify a condition for a group or an aggregate function used in select statement. The WHERE clause selects before grouping. The HAVING clause selects rows after grouping. Unlike HAVING clause, the WHERE clause cannot contain aggregate functions.*



5.What is Join?

Ans: An SQL Join is used to combine data from two or more tables, based on a common
field between them. For example, consider the following two tables.
Student Table
EnrollNo StudentName Address
1000 *geek1 geeksquiz1*
1001 geek2 geeksquiz2
1002 geek3 geeksquiz3

StudentCourse Table

CourseID EnrollNo
1 *1000*
2 *1000*
3 *1000*
1 *1002*
2 *1003*

Following is join query that shows names of students enrolled in different courseIDs.

```
SELECT StudentCourse.CourseID, Student.StudentName
FROM StudentCourse
INNER JOIN Customers
ON StudentCourse.EnrollNo = Student.EnrollNo
ORDER BY StudentCourse.CourseID;
```

The above query would produce following result.

CourseID StudentName
1 geek1
1 *geek2*
2 *geek1*
2 *geek3*
3 *geek1*

14.What is Identity?
Ans: Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers; the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

15. What is a view in SQL? How to create one
Ans: A view is a virtual table based on the result set of an SQL statement. We can create using the create view syntax.

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name WHERE
condition
```

16. What are the uses of view?

1. *Views can represent a subset of the data contained in a table; consequently, a view can Limit the degree of exposure of the underlying tables to the outer world: a given user*
may have permission to query the view, while denied access to the rest of the base table.
2. *Views can join and simplify multiple tables into a single virtual table*
3. *Views can act as aggregated tables, where the database engine aggregates data (sum, average etc.) and presents the calculated results as part of the data*
4. *Views can hide the complexity of data; for example, a view could appear as Sales2000 or Sales2001, transparently partitioning the actual underlying table.*
5. *Depending on the SQL engine used, views can provide extra security.*

17. What is a Trigger?

Ans: A Trigger is a code that is associated with insert, update, or delete operations. The
code is executed automatically whenever the associated query is executed on a table. Triggers can be useful to maintain integrity in a database.

18. What is a stored procedure?

Ans: A stored procedure is like a function containing a set of operations. It contains a set of operations that are commonly used in an application to do
some common database tasks.

19.What is the difference between Trigger and Stored Procedure?

Ans: Unlike Stored Procedures, Triggers cannot be called directly. They can only be associated with queries.

20.What is a transaction? What are ACID properties?

Ans: A Database Transaction is a set of database operations that must be treated as whole, means either all operations are executed or none of them.
An example can be bank transaction from one account to another account. Either both debit and credit operations must be executed or none of them.

ACID (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably.

21.What are indexes?

Ans: A database index is a data structure that improves the speed of data retrieval operations on a database table at the cost of additional writes and the use of more storage space to maintain the extra copy of data.
Data can be stored only in one order on disk. To support faster access according to different values, faster search like binary search for different values is desired, For this purpose, indexes are created on tables. These indexes need extra space on disk, but they allow faster search according to different frequently searched values.

22.What are clustered and non-clustered Indexes?

Ans: Clustered indexes is the index according to which data is physically stored on disk. Therefore, only one clustered index can be created on a given database table.
Non-clustered indexes don't define physical ordering of data, but logical ordering. Typically, a tree is created whose leaf point to disk records. B-Tree or B+ tree are used for this purpose.

23.What are Primary Keys and Foreign Keys?

Ans: Primary keys are the unique identifiers for each row. They must contain unique values and cannot be null. Due to their importance in relational databases, Primary keys are the most fundamental aspect of all keys and constraints. A table can have only one primary
key. Foreign keys are a method of ensuring data integrity and manifestation of the relationship between tables.

24.What is SQL ?

Ans: Structured Query Language(SQL) is a language designed specifically for communicating with databases. SQL is an ANSI (American National Standards Institute) standard .

25. What are the different type of SQL or different commands in SQL?

Ans: Frequently Asked SQL Interview Questions
1.DDL – Data Definition Language. DDL is used to define the structure that holds the data.
2. DM– Data Manipulation Language DML is used for the manipulation of the data itself. Typical operations are Insert, Delete, updating, and retrieving the data from the table
3. DCL–Data Control Language DCL is used to control the visibility of data like grantingdatabase access and setting privileges to create tables etc.
4.TCL-TransactionControl Language

26. What are the Advantages of SQL?

1. *SQL is not a proprietary language used by specific database vendors. Almost every Major DBMS supports SQL, so learning this one language will enable programmers to interact with any database like ORACLE, SQL, MYSQL, etc.*
2. *SQL is easy to learn. The statements are all made up of descriptive English words, and there aren't that many of them.*
3. *SQL is actually a very powerful language and by using its language elements you can perform very complex and sophisticated database operations.*

27. What is a field in a database?

A field is an area within a record reserved for a specific piece of data.
Examples: Employee Name, Employee ID, etc

28. What is a Record in a database?

A record is the collection of values/fields of a specific entity: i.e. a Employee, Salary etc.

29. What is a Table in a database ?

A table is a collection of records of a specific type. For example, employee table , salary table etc.

30. What is a database transaction?

Database transactions take a database from one consistent state to another. At the end of the transaction, the system must be in the prior state if the transaction fails or the status of the system should reflect the successful completion if the transaction goes through.

31. What are the properties of a transaction?

Properties of the transaction can be summarized as ACID Properties.
1. Atomicity
A transaction consists of many steps. When all the steps in a transaction are completed, it will get reflected in DB or if any step fails, all the transactions are rolled back.
2. Consistency
The database will move from one consistent state to another if the transaction succeeds and remain in the original state if the transaction fails.
3. Isolation
Every transaction should operate as if it is the only transaction in the system
4. Durability
Once a transaction has been completed successfully, the updated rows/records must be available for all other transactions on a permanent basis

32. What is a Database Lock?

Database lock tells a transaction if the data item in question is currently being used by other transactions.

33. What are the types of locks?

1. *Shared Lock*

When a shared lock is applied on data item, other transactions can only read the item, but can't write into it.

2. *Exclusive Lock*

When a exclusive lock is applied on data item, other transactions can't read or write into the data item.

34.What are the different type of normalization?

Frequently asked SQL Interview Questions

In database design, we start with one single table, with all possible columns. Lot of redundant data would be present since it's a single table. The process of removing the redundant data, by splitting up the table in a well-defined fashion is called normalization.

1. First Normal Form (1NF)

A relation is said to be in first normal form if and only if all underlying domains contain atomic values. After 1NF, we can still have redundant data

2. Second Normal Form (2NF)

A relation is said to be in 2NF if and only if it is in 1NF and every non-key attribute is fully dependent on the primary key. After 2NF, we can still have redundant data

3. Third Normal Form (3NF)

A relation is said to be in 3NF, if and only if it is in 2NF and every non-key attribute is nontransitively dependent on the primary key

35. What is a primary key?

Frequently Asked SQL Interview Questions

A primary key is a column whose values uniquely identify every row in a table. Primary key values can never be reused. If a row is deleted from the table, its primary key may not be assigned to any new rows in the future. To define a field as a primary key, the following conditions had to be met :

1. *No two rows can have the same primary key value.*
2. *Every row must have a primary key value*
3. *Primary key field cannot be null*
4. *Values in primary key columns can never be modified or updated*

36.What is a Composite Key ?

A Composite primary key is a type of candidate key, which represents a set of columns whose values uniquely identify every row in a table.

For example - if "Employee_ID" and "Employee Name" in a table is combined to uniquely identifies a row its called a Composite Key.

37. What is a Composite Primary Key?

A Composite primary key is a set of columns whose values uniquely identify every row in a table. What it means is that, the table which contains a composite primary key will be indexed based on columns specified in the primary key. This key will be referred in Foreign Key tables.

For example - if the combined effect of columns, "Employee_ID" and "Employee Name" in a table is required to uniquely identify a row, it's called a Composite Primary Key. In this

case, both columns will be represented as primary keys.

38. What is a Foreign Key?

Frequently Asked SQL Interview Questions

When a "one" table's primary key field is added to a related "many" table in order to create The common field that relates the two tables, is called a foreign key in the "many" table.

For example, the salary of an employee is stored in a salary table. The relation is established via foreign key column "Employee_ID_Ref" which refers "Employee_ID" field in an Employee table

39. What is a Unique Key?

The unique key is the same as the primary with the difference being the existence of null. Unique key field

allows one value as a NULL value.

40. Define SQL Insert Statement?

SQL INSERT statement is used to add rows to a table. For a full row insert, SQL Query should start with "insert into " statement followed by a table name and values command, followed by the values that need to be inserted into the table. Insert can be used in several ways:

1. *To insert a single complete row*
2. *To insert a single partial row*

41. Define SQL Update Statement ?

SQL Update is used to update data in a row or set of rows specified in the filter condition. The basic format of an SQL UPDATE statement is ,Update command followed by table to be updated and SET command followed by column names and their new values followed by filter condition that determines which rows should be updated

42. Define SQL Delete Statement ?

SQL Delete is used to delete a row or set of rows specified in the filter condition.
The basic format of an SQL DELETE statement is, DELETE FROM command followed by table name followed by filter condition that determines which rows should be updated.
43.What are wild cards used in database for Pattern Matching ?
SQL Like operator is user for pattern matching. SQL 'Like' command takes more time to process. So before using like operator, consider suggestions given below on when and where to use wild card search.
1) *Don't overuse wild cards. If another search operator will do, use it instead.*
2) *When you do use wild cards, try not to use them at the beginning of the search pattern,*
unless absolutely necessary. Search patterns that begin with wild cards are the slowest to process.
3) *Pay careful attention to the placement of the wild card symbols. If they are misplaced,*
you might not return the data you intended

44. Define Join and explain different type of joins?

In order to avoid data duplication, data is stored in related tables . Join keyword is used to fetch data from related table. Join return rows when there is at least one match in both tables. Type of joins are
Right Join
Return all rows from the right table, even if there are no matches in the left table .Outer Join Left Join
Return all rows from the left table, even if there are no matches in the right table .
Full Join
Return rows when there is a match in one of the tables .

45.What is Self-Join?

Self-join is query used to join a table to itself. Aliases should be used for the same table comparison.

46. What is Cross Join?

Cross Join will return all records where each row from the first table is combined with each row from the second table.

SQL Interview Questions and answers on Database Views

47. What is a view?

Views are virtual tables. Unlike tables that contain data, views simply contain queries that dynamically retrieve data when used.

48. What is a materialized view?

Materialized views is also a view but are disk based. Materialized views get updated on specific duration, base upon the interval specified in the query definition. We can index materialized view.

49. What are the advantages and disadvantages of views in a database?

Advantages:

1. *Views doesn't store data in a physical location.*
2. *View can be use to hide some of the columns from the table*
3. *Views can provide Access Restriction, since data insertion, update and deletion is not possible on the view.*

Disadvantages:

1. *When a table is dropped, associated view become irrelevant.*
2. *Since view are created when a query requesting data from view is triggered, its bit slow*
3. *When views are created for large tables, it occupy more memory .*

SQL Interview Questions and answers on Stored Procedures and Triggers

50. What is a stored procedure?

Stored Procedure is a function which contain collection of SQL Queries. Procedure can take inputs , process them and send back output.

51.What are the advantages a stored procedure?

Stored Procedures are pre-complied and stored in database. This enable the database to execute the queries much faster. Since many queries can be included in a stored procedure, round trip time to execute multiple queries from source code to database and back is avoided.

52.What is a trigger?

Database are set of commands that get executed when an event(Before Insert, After Insert, On Update, On delete of a row) occurs on a table, views.

53. ***Explain the difference between DELETE , TRUNCATE and DROP commands?***

Once delete operation is performed, Commit and Rollback can be performed to retrieve data.

Once truncate statement is executed, Commit and Rollback statement cant be performed. Where condition can be used along with delete statement but it cant be used with truncate statement.

Drop command is used to drop the table or keys like primary,foreign from a table.

54. ***What is the difference between Cluster and Non cluster Index?***

A clustered index reorders the way records in the table are physically stored. There can be only one clustered index per table. It make data retrieval faster.

A non clustered index does not alter the way it was stored but creates a complete separate object within the table. As a result insert and update command will be faster.

55. What is Union, minus and Interact commands?

MINUS operator is used to return rows from the first query but not from the second query. INTERSECT operator is used to return rows returned by both the queries.

56.What's the difference between a primary key and a unique key?

Both primary key and unique enforce uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the column, where are unique creates a non clustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

62. What is a transaction and what are ACID properties?

A transaction is a logical unit of work in which, all the steps must be performed or none. ACID stands for Atomicity, Consistency, Isolation, Durability. These are the properties of a transaction. For more information and explanation of these properties, see SQL Server books online or any RDBMS fundamentals text book.

63.Explain different isolation levels

An isolation level determines the degree of isolation of data between concurrent transactions. The default SQL Server isolation level is Read Committed. Here are the other isolation levels (in the ascending order of isolation): Read Uncommitted, Read

Committed, Repeatable Read, Serializable. See SQL Server books online for an explanation of the isolation levels. Be sure to read about SET TRANSACTION ISOLATION LEVEL, which lets you customize the isolation level at the connection level.

64.CREATE INDEX myIndex ON myTable(myColumn)

What type of Index will get created after executing the above statement?
Non-clustered index. Important thing to note: By default a clustered index gets created on the primary key, unless specified otherwise.

65.What's the maximum size of a row?

8060 bytes. Don't be surprised with questions like 'what is the maximum number of columns per table'. Check out SQL Server books online for the page titled: "Maximum Capacity Specifications".

66. Explain Active/Active and Active/Passive cluster configurations

Hopefully you have experience setting up cluster servers. But if you don't, at least be familiar with the way clustering works and the two clustering configurations Active/Active and Active/Passive. SQL Server books online has enough information on this topic and there is a good white paper available on Microsoft site.

67. Explain the architecture of SQL Server

This is a very important question and you better be able to answer it if consider yourself a DBA. SQL Server books online is the best place to read about SQL Server architecture. Read up the chapter dedicated to SQL Server Architecture.

68. What is lock escalation?

Lock escalation is the process of converting a lot of low level locks (like row locks, page locks) into higher level locks (like table locks). Every lock is a memory structure too many locks would mean, more memory being occupied by locks. To prevent this from happening,
SQL Server escalates the many fine-grain locks to fewer coarse-grain locks. Lock escalation
threshold was definable in SQL Server 6.5, but from SQL Server 7.0 onwards it's dynamically managed by SQL Server.

69.What is the difference between DELETE TABLE and TRUNCATE TABLE commands?

DELETE TABLE is a logged operation, so the deletion of each row gets logged in the transaction log, which makes it slow. TRUNCATE TABLE also deletes all the rows in a table, but it won't log the deletion of each row, instead, it logs the de-allocation of the data pages of the table, which makes it faster. Of course, the TRUNCATE TABLE can be rolled back.

70. Explain the storage models of OLAP

Check out MOLAP, ROLAP, and HOLAP in SQL Server books online for more information.

71. What are the new features introduced in SQL Server 2000 (or the latest release of SQL Server at the time of your interview)?

What changed between the previous version of SQL Server and the current version? This question is generally asked to see how current is your knowledge. Generally there is a section in the beginning of the books online titled "What's New", which has all such information. Of course, reading just that is not enough, you should have tried those things to better answer the questions. Also check out the section titled "Backward Compatibility" in books online which talks about the changes that have taken place in the new version.

72. What are constraints?

Explain different types of constraints

Constraints enable the RDBMS enforce the integrity of the database automatically, without needing you to create triggers, rule or defaults.

Types of constraints: NOT NULL, CHECK, UNIQUE, PRIMARY KEY, FOREIGN KEY

For an explanation of these constraints see books online for the pages titled: "Constraints" and "CREATE TABLE", "ALTER TABLE"

73. What is an index? What are the types of indexes? How many clustered indexes can be created on a table?

I create a separate index on each column of a table. what are the advantages and disadvantages of this approach?

Indexes in SQL Server are similar to the indexes in books. They help SQL Server retrieve the data quicker. Indexes are of two types. Clustered indexes and non-clustered indexes. When you createe a clustered index on a table, all the rows in the table are stored in the order of the clustered index key. So, there can be only one clustered index per table. Non-clustered indexes have their own storage separate from the table data storage. Non-clustered indexes are stored as B-tree structures (so do clustered indexes), with the leaf level nodes having the index key and it's row locater. The row located could be the RID or the Clustered index key, depending up on the absence or presence of clustered index on the table. If you create an index on each column of a table, it improves the query performance, as the query optimizer can choose from all the existing indexes to come up with an efficient execution plan. At the same time, data modification operations (such as INSERT, UPDATE, DELETE) will become slow, as every time data changes in the table, all the indexes need to be updated. Another disadvantage is that, indexes need disk space, the more indexes you have, more disk space is used.

74. What is RAID and what are different types of RAID configurations?
RAID stands for Redundant Array of Inexpensive Disks, used to provide fault tolerance to database servers. There are six RAID levels 0 through 5 offering different levels of performance, fault tolerance. MSDN has some information about RAID levels and for detailed information, check out the RAID advisory board's homepage.

75. What are the steps you will take to improve performance of a poor performing query?
This is a very open ended question and there could be a lot of reasons behind the poor performance of a query. But some general issues that you could talk about would be: No indexes, table scans, missing or out of date statistics, blocking, excess recompilations of stored procedures, procedures and triggers without SET NOCOUNT ON, poorly written query with unnecessarily complicated joins, too much normalization, excess usage of cursors and temporary tables.

Some of the tools/ways that help you troubleshooting performance problems are: SET SHOWPLAN_ALL ON, SET SHOWPLAN_TEXT ON, SET STATISTICS IO ON, SQL Server Profiler, Windows NT /2000 Performance monitor, Graphical execution plan in Query Analyzer.

76. What are the steps you will take, if you are tasked with securing an SQL Server?
Again this is another open ended question. Here are some things you could talk about: Preferring NT authentication, using server, database and application roles to control access to the data, securing the physical database files using NTFS permissions, using an unguessable SA password, restricting physical access to the SQL Server, renaming the Administrator account on the SQL Server computer, disabling the Guest account, enabling auditing, using multiprotocol encryption, setting up SSL, setting up firewalls, isolating SQL Server from the web server etc.

77. What is a deadlock and what is a live lock? How will you go about resolving deadlocks?
Deadlock is a situation when two processes, each having a lock on one piece of data, attempt to acquire a lock on the other's piece. Each process would wait indefinitely for the other to release the lock, unless one of the user processes is terminated. SQL Server detects deadlocks and terminates one user's process.
A live lock is one, where a request for an exclusive lock is repeatedly denied because a series of overlapping shared locks keeps interfering. SQL Server detects the situation after four denials and refuses further shared locks. A live lock also occurs when read transactions monopolize a table or page, forcing a write transaction to wait indefinitely

78. *What is blocking and how would you troubleshoot it?*

Blocking happens when one connection from an application holds a lock and a second connection requires a conflicting lock type. This forces the second connection to wait, blocked on the first.

79. *Explain CREATE DATABASE syntax*

Many of us are used to creating databases from the Enterprise Manager or by just issuing the command: CREATE DATABAE MyDB. But what if you have to create a database with two file groups, one on drive C and the other on drive D with log on drive E with an initial size of 600 MB and with a growth factor of 15%? That's why being a DBA you should be familiar with the CREATE DATABASE syntax. Check out SQL Server books online for moreinformation.

80. *How to restart SQL Server in single user mode? How to start SQL Server in minimal configuration mode?*

SQL Server can be started from command line, using the SQLSERVR.EXE. This EXE has some very important parameters with which a DBA should be familiar with. -m is used for starting SQL Server in single user mode and -f is used to start the SQL Server in minimal configuration mode. Check out SQL Server books online for more parameters and their explanations.

81. *As a part of your job, what are the DBCC commands that you commonly use for database maintenance?*

DBCC CHECKDB, DBCC CHECKTABLE, DBCC CHECKCATALOG, DBCC CHECKALLOC, DBCC SHOWCONTIG, DBCC SHRINKDATABASE, DBCC SHRINKFILE etc. But there are a whole load of DBCC commands which are very useful for DBAs. Check out SQL Server books online for more information.

82. *What are statistics, under what circumstances they go out of date, how do you update them?*

Statistics determine the selectivity of the indexes. If an indexed column has unique values then the selectivity of that index is more, as opposed to an index with non-unique values. Query optimizer uses these indexes in determining whether to choose an index or not while executing a query.

Some situations under which you should update statistics:

1) *If there is significant change in the key values in the index*

2) If a large amount of data in an indexed column has been added, changed, or removed (that is, if the distribution of key values has changed), or the table has been truncated using the TRUNCATE TABLE statement and then repopulated
3) Database is upgraded from a previous version

108.Define constraints.
Constraints enforce integrity of the database. Constraints can be of following types Not Null Check
Unique
Primary key Foreign key

109.Define stored procedure.
Stored procedure is a set of pre-compiled SQL statements, executed when it is called in the program.

110.Define Trigger.
Triggers are similar to stored procedure except it is executed automatically when any operations are occurred on the table.

111. ***What is RDBMS?***
Relational Data Base Management Systems (RDBMS) are database management systems that maintain data records and indices in tables. Relationships may be created and maintained across and among the data and tables. In a relational database, relationships between data items are expressed by means of tables. Interdependencies among these tables are expressed by data values rather than by pointers. This allows a high degree of data independence. An RDBMS has the capability to recombine the data items from different files, providing powerful tools for data usage.

112. ***What are the properties of the Relational tables?***
Relational tables have six properties:
1. *Values are atomic.*
2. *Column values are of the same kind.*
3. *Each row is unique.*
4. *The sequence of columns is insignificant.*
5. *The sequence of rows is insignificant.*
6. *Each column must have a unique name.*

113. *What is Normalization ?*

Database normalization is a data design and organization process applied to data structures based on rules that help building relational databases. In relational database design, the process of organizing data to minimize redundancy is called normalization. Normalization usually involves dividing a database into two or more tables and defining relationships between the tables. The objective is to isolate data so that additions, deletions, and modifications of a field can be made in just one table and then propagated through the rest of the database via the defined relationships.

114. What is De-normalization?

De-normalization is the process of attempting to optimize the performance of a database by adding redundant data. It is sometimes necessary because current DBMSs implement the relational model poorly. A true relational DBMS would allow for a fully normalized database at the logical level, while providing physical storage of data that is tuned for high performance. De-normalization is a technique to move from higher to lower normal forms of database modeling in order to speed up database access.

115. What are different normalization forms?

1. 1NF: Eliminate Repeating Groups Make a separate table for each set of related attributes, and give each table a primary key. Each field contains at most one value from its attribute domain.
2. 2NF: Eliminate Redundant Data If an attribute depends on only part of a multivalued key, remove it to a separate table.
3. 3NF: Eliminate Columns Not Dependent On Key If attributes do not contribute to a description of the key, remove them to a separate table. All attributes must be directly dependent on the primary key.
4. BCNF: Boyce-Codd Normal Form If there are non-trivial dependencies between candidate key attributes, separate them out into distinct tables.
5. 4NF: Isolate Independent Multiple Relationships No table may contain two or more 1:n or n:m relationships that are not directly related.
6. 5NF: Isolate Semantically Related Multiple Relationships There may be practical constrains on information that justify separating logically related many-to-many relationships.

7. ONF: Optimal Normal Form A model limited to only simple (elemental) facts, as expressed in Object Role Model notation.
8. *DKNF: Domain-Key Normal Form A model free from all modification anomalies is said to be in DKNF.*
Remember, these normalization guidelines are cumulative. For a database to be in 3NF, it must first fulfill all the criteria of a 2NF and 1NF database.

116. *What is Stored Procedure?*
A stored procedure is a named group of SQL statements that have been previously created and stored in the server database. Stored procedures accept input parameters so that a single procedure can be used over the network by several clients using different input
data. And when the procedure is modified, all clients automatically get the new version. Stored procedures reduce network traffic and improve performance. Stored procedures can be used to help ensure the integrity of the database. e.g. sp_helpdb, sp_renamedb, sp_depends etc.

117. What is Trigger?
A trigger is a SQL procedure that initiates an action when an event (INSERT, DELETE or UPDATE) occurs. Triggers are stored in and managed by the DBMS. Triggers are used to maintain the referential integrity of data by changing the data in a systematic fashion. A trigger cannot be called or executed; DBMS automatically fires the trigger as a result of a data modification to the associated table. Triggers can be viewed as similar to stored procedures in that both consist of procedural logic that is stored at the database level. Stored procedures, however, are not event-drive and are not attached to a specific table as triggers are. Stored procedures are explicitly executed by invoking a CALL to the procedure while triggers are implicitly executed. In addition, triggers can also execute stored procedures.

118. What is Nested Trigger?
A trigger can also contain INSERT, UPDATE and DELETE logic within itself, so when the trigger is fired because of data modification it can also cause another data modification, thereby firing another trigger. A trigger that contains data modification logic within itself is called a nested trigger.

119. What is View?

A simple view can be thought of as a subset of a table. It can be used for retrieving data, as well as updating or deleting rows. Rows updated or deleted in the view are updated or deleted in the table the view was created with. It should also be noted that as data in the original table changes, so does data in the view, as views are the way to look at part of the original table. The results of using a view are not permanently stored in the database. The data accessed through a view is actually constructed using standard T-SQL select command and can come from one to many different base tables or even other views.

120. *What is Index?*

An index is a physical structure containing pointers to the data. Indices are created in an existing table to locate rows more quickly and efficiently. It is possible to create an index on one or more columns of a table, and each index is given a name. The users cannot see the indexes; they are just used to speed up queries. Effective indexes are one of the best ways to improve performance in a database application. A table scan happens when there is no index available to help a query. In a table scan SQL Server examines every row in the table to satisfy the query results. Table scans are sometimes unavoidable, but on large tables, scans have a terrific impact on performance.

121. What is a Linked Server?

Linked Servers is a concept in SQL Server by which we can add other SQL Server to a Group and query both the SQL Server dbs using T-SQL Statements. With a linked server, you can create very clean, easy to follow, SQL statements that allow remote data to be retrieved, joined and combined with local data. Stored Procedure sp_addlinkedserver, sp_addlinkedserverlogin will be used add new Linked Server.

122. What is Cursor?

Cursor is a database object used by applications to manipulate data in a set on a row-by- row basis, instead of the typical SQL commands that operate on all the rows in the set at one time. In order to work with a cursor we need to perform some steps in the following order: 1. Declare cursor
2. Open cursor
3. Fetch row from the cursor
4. Process fetched row
5. Close cursor 6. De-allocate cursor

123. *What is Collation?*

Collation refers to a set of rules that determine how data is sorted and compared. Character data is sorted using rules that define the correct character sequence, with options for specifying case sensitivity, accent marks, kana character types and character width.

124. *What is Difference between Function and Stored Procedure?*

UDF can be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section where as Stored procedures cannot be. UDFs that return tables can be treated as another row set. This can be used in JOINs with other tables. Inline UDF's can be thought of as views that take parameters and can be used in JOINs and other Row set operations.

125. *What is sub-query? Explain properties of sub-query?*

Sub-queries are often referred to as sub-selects, as they allow a SELECT statement to be executed arbitrarily within the body of another SQL statement. A sub-query is executed by enclosing it in a set of parentheses. Sub-queries are generally used to return a single row as an atomic value, though they may be used to compare values against multiple rows with the IN keyword. A subquery is a SELECT statement that is nested within another T-SQL statement. A subquery SELECT statement if executed independently of the T-SQL statement, in which it
is nested, will return a resultset. Meaning a subquery SELECT statement can standalone and is not depended on the statement in which it is nested. A subquery SELECT statement
can return any number of values, and can be found in, the column list of a SELECT statement, a FROM, GROUP BY, HAVING, and/or ORDER BY clauses of a T-SQL statement.
A Subquery can also be used as a parameter to a function call. Basically a subquery can be used anywhere an expression can be used.

126. What are different Types of Join?

1. Cross Join A cross join that does not have a WHERE clause produces the Cartesian product of the tables involved in the join. The size of a Cartesian product result set is the number of rows in the first table multiplied by the number of rows in the second table. The common example is when company wants to combine each product with a pricing table to analyze each product at each price.
2. Inner Join A join that displays only the rows that have a match in both joined tables is known as inner Join. This is the default type of join in the Query and View Designer.

3. *Outer Join A join that includes rows even if they do not have related rows in the joined table is an Outer Join. You can create three different outer join to specify the unmatched rows to be included:*
1. *Left Outer Join: In Left Outer Join all rows in the first-named table i.e. "left" table, which appears leftmost in the JOIN clause are included. Unmatched rows in the right table do not appear.*
2. *Right Outer Join: In Right Outer Join all rows in the second-named table i.e. "right" table, which appears rightmost in the JOIN clause are included. Unmatched rows in the left table are not included.*
3. *Full Outer Join: In Full Outer Join all rows in all joined tables are included, whether they are matched or not.*
4. *Self Join This is a particular case when one table joins to itself, with one or two aliases to avoid confusion. A self join can be of any type, as long as the joined tables are the same. A self join is rather unique in that it involves a relationship with only one table. The common example is when company has a hierarchal reporting structure whereby one member of staff reports to another. Self Join can be Outer Join or Inner Join.*

127. What are primary keys and foreign keys?

Primary keys are the unique identifiers for each row. They must contain unique values and cannot be null. Due to their importance in relational databases, Primary keys are the most fundamental of all keys and constraints. A table can have only one Primary key. Foreign keys are both a method of ensuring data integrity and a manifestation of the relationship between tables.

128. What is User Defined Functions? What kind of User-Defined Functions can be created?

User-Defined Functions allow defining its own T-SQL functions that can accept 0 or more parameters and return a single scalar data value or a table data type. Different Kinds of User-Defined Functions created are:
1. Scalar User-Defined Function A Scalar user-defined function returns one of the scalar data types. Text, image and timestamp data types are not supported. These are the type of user-defined functions that most developers are used to in other programming languages. You pass in 0 to many parameters and you get a return value.
2. Inline Table-Value User-Defined Function An Inline Table-Value user-defined function returns a table data type and is an exceptional alternative to a view as the user-defined function can pass parameters into a T-SQL select command and in essence provide us with a parameterized, non-updateable view of the underlying tables.

3. *Multi-statement Table-Value User-Defined Function A Multi-Statement TableValue user- defined function returns a table and is also an exceptional alternative to a view as the function can support multiple T-SQL statements to build the final result where the view is limited to a single SELECT statement. Also, the ability to pass parameters into a TSQL select command or a group of them gives us the capability to in essence create a parameterized, non-updateable view of the data in the underlying tables. Within the create function command you must define the table structure that is being returned. After creating this type of user-defined function, It can be used in the FROM clause of a T-SQL command unlike the behavior found when using a stored procedure which can also return record sets.*

129. *What is Identity?*

Identity (or AutoNumber) is a column that automatically generates numeric values. A start and increment value can be set, but most DBA leave these at 1. A GUID column also generates numbers; the value of this cannot be controlled. Identity/GUID columns do not need to be indexed.

130. Which TCP/IP port does SQL Server run on? How can it be changed?

SQL Server runs on port 1433. It can be changed from the Network Utility TCP/IP properties.

131. What are the difference between clustered and a non-clustered index?

1. A clustered index is a special type of index that reorders the way records in the table are physically stored. Therefore table can have only one clustered index. The leaf nodes of a clustered index contain the data pages.
2. A non clustered index is a special type of index in which the logical order of the index does not match the physical stored order of the rows on disk. The leaf node of a non clustered index does not consist of the data pages. Instead, the leaf nodes contain index rows.

132. What are the different index configurations a table can have?

A table can have one of the following index configurations:
1. No indexes
2. A clustered index
3. A clustered index and many nonclustered indexes
4. A non clustered index
5. Many non clustered indexes

134. What are different types of Collation Sensitivity?

1. Case sensitivity - A and a, B and b, etc.
2. Accent sensitivity
3. Kana Sensitivity - When Japanese kana characters Hiragana and Katakana are treated differently, it is called Kana sensitive.
4. Width sensitivity - A single-byte character (half-width) and the same character represented as a double-byte character (full-width) are treated differently than it is width sensitive.

135. *What is OLTP (Online Transaction Processing)?*

In OLTP - online transaction processing systems relational database design use the discipline of data modeling and generally follow the Codd rules of data normalization in order to ensure absolute data integrity. Using these rules complex information is broken down into its most simple structures (a table) where all of the individual atomic level elements relate to each other and satisfy the normalization rules.

136. What's the difference between a primary key and a unique key?

Both primary key and unique key enforces uniqueness of the column on which they are defined. But by default primary key creates a clustered index on the column, where are unique creates a non clustered index by default. Another major difference is that, primary key doesn't allow NULLs, but unique key allows one NULL only.

137. What is difference between DELETE and TRUNCATE commands?

Delete command removes the rows from a table based on the condition that we provide with a WHERE clause. Truncate will actually remove all the rows from a table and there will be no data in the table after we run the truncate command.
1. TRUNCATE:
1. TRUNCATE is faster and uses fewer system and transaction log resources than DELETE.
2. TRUNCATE removes the data by deallocating the data pages used to store the table's data, and only the page deallocations are recorded in the transaction log.
3. TRUNCATE removes all rows from a table, but the table structure, its columns, constraints, indexes and so on, remains. The counter used by an identity for new rows is reset to the seed for the column.
4. You cannot use TRUNCATE TABLE on a table referenced by a FOREIGN KEY constraint. Because TRUNCATE TABLE is not logged, it cannot activate a trigger.

5. *TRUNCATE cannot be rolled back.*
6. *TRUNCATE is DDL Command.*
7. *TRUNCATE Resets identity of the table*
2. *DELETE:*
1. *DELETE removes rows one at a time and records an entry in the transaction log for each deleted row.*
2. *If you want to retain the identity counter, use DELETE instead. If you want to remove table definition and its data, use the DROP TABLE statement.*
3. DELETE Can be used with or without a WHERE clause
4. *DELETE Activates Triggers.*
5. *DELETE can be rolled back.*
6. *DELETE is DML Command.*
7. *DELETE does not reset identity of the table.*

Note: DELETE and TRUNCATE both can be rolled back when surrounded by TRANSACTION if the current session is not closed. If TRUNCATE is written in Query Editor surrounded by TRANSACTION and if session is closed, it can not be rolled back but DELETE can be rolled back.

138. When is the use of UPDATE_STATISTICS command?

This command is basically used when a large processing of data has occurred. If a large amount of deletions any modification or Bulk Copy into the tables has occurred, it has to update the indexes to take these changes into account. UPDATE_STATISTICS updates the indexes on these tables accordingly.

139. What is the difference between a HAVING CLAUSE and a WHERE CLAUSE?

They specify a search condition for a group or an aggregate. But the difference is that HAVING can be used only with the SELECT statement. HAVING is typically used in a GROUP BY clause. When GROUP BY is not used, HAVING behaves like a WHERE clause. Having Clause is basically used only with the GROUP BY function in a query whereas WHERE
Clause is applied to each row before they are part of the GROUP BY function in a query.

140. What are the properties and different Types of Sub-Queries?

1. Properties of Sub-Query
1. A sub-query must be enclosed in the parenthesis.
2. A sub-query must be put in the right hand of the comparison operator, and
3. A sub-query cannot contain an ORDER-BY clause.
4. A query can contain more than one sub-query.

2. Types of Sub-Query
1. Single-row sub-query, where the sub-query returns only one row.
2. Multiple-row sub-query, where the sub-query returns multiple rows,. and 3. Multiple column sub-query, where the sub-query returns multiple columns

141. What is SQL Profiler?
SQL Profiler is a graphical tool that allows system administrators to monitor events in an instance of Microsoft SQL Server. You can capture and save data about each event to a file or SQL Server table to analyze later. For example, you can monitor a production environment to see which stored procedures are hampering performances by executing too slowly. Use SQL Profiler to monitor only the events in which you are interested. If traces are becoming too large, you can filter them based on the information you want, so that only a subset of the event data is collected. Monitoring too many events adds overhead to the server and the monitoring process and can cause the trace file or trace table to grow very large, especially when the monitoring process takes place over a long period of time.

142. What are the authentication modes in SQL Server? How can it be changed?
Windows mode and Mixed Mode - SQL and Windows. To change authentication mode in SQL Server click Start, Programs, Microsoft SQL Server and click SQL Enterprise Manager to run SQL Enterprise Manager from the Microsoft SQL Server program group. Select the server then from the Tools menu select SQL Server Configuration Properties, and choose the Security page.

143. Which command using Query Analyzer will give you the version of SQL server and operating system?
SELECT SERVERPROPERTY ('productversion'), SERVERPROPERTY ('productlevel'), SERVERPROPERTY ('edition').

144. What is SQL Server Agent?
SQL Server agent plays an important role in the day-to-day tasks of a database administrator (DBA). It is often overlooked as one of the main tools for SQL Server management. Its purpose is to ease the implementation of tasks for the DBA, with its full- function scheduling engine, which allows you to schedule your own jobs and scripts.

145. Can a stored procedure call itself or recursive stored procedure? How much level SP nesting is possible?

Yes. Because Transact-SQL supports recursion, you can write stored procedures that call themselves. Recursion can be defined as a method of problem solving wherein the solution is arrived at by repetitively applying it to subsets of the problem. A common application of recursive logic is to perform numeric computations that lend themselves to repetitive evaluation by the same processing steps. Stored procedures are nested when one stored procedure calls another or executes managed code by referencing a CLR routine, type, or aggregate. You can nest stored procedures and managed code references up to 32 levels

.

146. What is Log Shipping?

Log shipping is the process of automating the backup of database and transaction log files on a production SQL server, and then restoring them onto a standby server. Enterprise Editions only supports log shipping. In log shipping the transactional log file from one server is automatically updated into the backup database on the other server. If one server fails, the other server will have the same db and can be used this as the Disaster Recovery plan. The key feature of log shipping is that it will automatically backup transaction logs throughout the day and automatically restore them on the standby server at defined interval.

147. Name 3 ways to get an accurate count of the number of records in a table?

*SELECT * FROM table1*
SELECT COUNT(*) FROM table1
SELECT rows FROM sysindexes WHERE id = OBJECT_ID(table1) AND indid < 2

148. What does it mean to have QUOTED_IDENTIFIER ON? What are the implications of having it OFF?

When SET QUOTED_IDENTIFIER is ON, identifiers can be delimited by double quotation marks, and literals must be delimited by single quotation marks. When SET QUOTED_IDENTIFIER is OFF, identifiers cannot be quoted and must follow all TransactSQL rules for identifiers.

149. What is the difference between a Local and a Global temporary table?

1. A local temporary table exists only for the duration of a connection or, if defined inside a compound statement, for the duration of the compound statement.

2. A global temporary table remains in the database permanently, but the rows exist only within a given connection. When connection is closed, the data in the global temporary table disappears. However, the table definition remains with the database for access when database is opened next time.

150. *What is the STUFF function and how does it differ from the REPLACE function?*
STUFF function is used to overwrite existing characters. Using this syntax, STUFF (string_expression, start, length, replacement_characters), string_expression is the string that will have characters substituted, start is the starting position, length is the number of characters in the string that are substituted, and replacement_characters are the new characters interjected into the string. REPLACE function to replace existing characters of all occurrences. Using the syntax REPLACE (string_expression, search_string, replacement_string), where every incidence of search_string found in the string_expression will be replaced with replacement_string.

151. What is PRIMARY KEY?
A PRIMARY KEY constraint is a unique identifier for a row within a database table. Every table should have a primary key constraint to uniquely identify each row and only one primary key constraint can be created for each table. The primary key constraints are used to enforce entity integrity.

152. What is UNIQUE KEY constraint?
A UNIQUE constraint enforces the uniqueness of the values in a set of columns, so no duplicate values are entered. The unique key constraints are used to enforce entity integrity as the primary key constraints.

153. What is FOREIGN KEY?
A FOREIGN KEY constraint prevents any actions that would destroy links between tables with the corresponding data values. A foreign key in one table points to a primary key in another table. Foreign keys prevent actions that would leave rows with foreign key values when there are no primary keys with that value. The foreign key constraints are used to enforce referential integrity.

154. What is CHECK Constraint?
A CHECK constraint is used to limit the values that can be placed in a column. The check constraints are used to enforce domain integrity.

155. What is NOT NULL Constraint?

A NOT NULL constraint enforces that the column will not accept null values. The not null constraints are used to enforce domain integrity, as the check constraints.

156. How to get @@ERROR and @@ROWCOUNT at the same time?

If @@Rowcount is checked after Error checking statement then it will have 0 as the value of @@Recordcount as it would have been reset. And if @@Recordcount is checked before the error-checking statement then @@Error would get reset. To get @@error and @@rowcount at the same time do both in same statement and store them in local variable. SELECT @RC = @@ROWCOUNT, @ER = @@ERROR

157. What is a Scheduled Jobs or What is a Scheduled Tasks?

Scheduled tasks let user automate processes that run on regular or predictable cycles. User can schedule administrative tasks, such as cube processing, to run during times of slow business activity. User can also determine the order in which tasks run by creating job steps within a SQL Server Agent job. E.g. back up database, Update Stats of Tables. Job steps give user control over flow of execution. If one job fails, user can configure SQL Server Agent to continue to run the remaining tasks or to stop execution.

158. What are the advantages of using Stored Procedures?

1. Stored procedure can reduced network traffic and latency, boosting application performance.
2. Stored procedure execution plans can be reused, staying cached in SQL Server's memory, reducing server overhead.
3. Stored procedures help promote code reuse.
4. Stored procedures can encapsulate logic. You can change stored procedure code without affecting clients.
5. Stored procedures provide better security to your data.

159. What is a table called, if it has neither Cluster nor Non-cluster Index? What is it used for?

Unindexed table or Heap. Microsoft Press Books and Book on Line (BOL) refers it as Heap. A heap is a table that does not have a clustered index and, therefore, the pages are not linked by pointers. The IAM pages are the only structures that link the pages in a table together. Unindexed tables are good for fast storing of data. Many times it is better to drop all indexes from table and then do bulk of inserts and to restore those indexes after that.

160. *Can SQL Servers linked to other servers like Oracle?*

SQL Server can be linked to any server provided it has OLE-DB provider from Microsoft to allow a link. E.g. Oracle has an OLE-DB provider for oracle that Microsoft provides to add it as linked server to SQL Server group.

161. How to implement one-to-one, one-to-many and many-to-many relationships while designing tables?

One-to-One relationship can be implemented as a single table and rarely as two tables with primary and foreign key relationships. One-to-Many relationships are implemented by splitting the data into two tables with primary key and foreign key relationships. Manyto-Many relationships are implemented using a junction table with the keys from both the tables forming the composite primary key of the junction table.

162. What are the basic functions for master, msdb, model, tempdb and resource databases?

1. The master database holds information for all databases located on the SQL Server instance and is theglue that holds the engine together. Because SQL Server cannot start without a functioning masterdatabase, you must administer this database with care.
2. The msdb database stores information regarding database backups, SQL Agent information, DTS packages, SQL Server jobs, and some replication information such as for log shipping.
3. The tempdb holds temporary objects such as global and local temporary tables and stored procedures.
4. The model is essentially a template database used in the creation of any new user database created in the instance.
5. The resoure Database is a read-only database that contains all the system objects that are included with SQL Server. SQL Server system objects, such as sys.objects, are physically persisted in the Resource database, but they logically appear in the sys schema of every database. The Resource database does not contain user data or user metadata.

163. What is Service Broker?

Service Broker is a message-queuing technology in SQL Server that allows developers to integrate SQL Server fully into distributed applications. Service Broker is feature which provides facility to SQL Server to send an asynchronous, transactional message. it allows a database to send a message to another database without waiting for the response, so the application will continue to function if the remote database is temporarily unavailable.

164. Where SQL server user names and passwords are stored in SQL server?
They get stored in System Catalog Views sys.server_principals and sys.sql_logins.

165. What is Policy Management?
Policy Management in SQL SERVER 2008 allows you to define and enforce policies for configuring and managing SQL Server across the enterprise. Policy-Based Management is configured in SQL Server Management Studio (SSMS). Navigate to the Object Explorer and expand the Management node and the Policy Management node; you will see the Policies, Conditions, and Facets nodes.

166. What is Replication and Database Mirroring?
Database mirroring can be used with replication to provide availability for the publication database. Database mirroring involves two copies of a single database that typically reside on different computers. At any given time, only one copy of the database is currently available to clients which are known as the principal database. Updates made by clients to the principal database are applied on the other copy of the database, known as the mirror database. Mirroring involves applying the transaction log from every insertion, update, or deletion made on the principal database onto the mirror database.

167. What are Sparse Columns?
A sparse column is another tool used to reduce the amount of physical storage used in a database. They are the ordinary columns that have an optimized storage for null values. Sparse columns reduce the space requirements for null values at the cost of more overhead to retrieve nonnull values.

168. What does TOP Operator Do?
The TOP operator is used to specify the number of rows to be returned by a query. The TOP operator has a new addition in SQL SERVER 2008 that it accepts variables as well as literal values and can be used with INSERT, UPDATE, and DELETES statements.

169. What is CTE?
CTE is an abbreviation of Common Table Expression. A Common Table Expression (CTE) is an expression that can be thought of as a temporary result set which is defined within the execution of a single SQL statement. A CTE is similar to a derived table in that it is not stored as an object and lasts only for the duration of the query.

170. *What is MERGE Statement?*

MERGE is a new feature that provides an efficient way to perform multiple DML operations. In previous versions of SQL Server, we had to write separate statements to INSERT, UPDATE, or DELETE data based on certain conditions, but now, using MERGE statement we can include the logic of such data modifications in one statement that even checks when the data is matched then just updates it and when unmatched then inserts it. One of the most important advantages of a MERGE statement is all the data is read and processed only once.

171. What is a Filtered Index?

A filtered Index is used to index a portion of rows in a table which means it applies filter on INDEX which improves query performance, reduces index maintenance costs, and reduce index storage costs compared with full-table indexes. When we see an Index created with somewhere clause then that is actually a FILTERED INDEX.

172. Which are new data types introduced in SQL SERVER 2008?

1. The GEOMETRY Type: The GEOMETRY data type is a system .NET common language runtime (CLR) data type in SQL Server. This type represents data in a two-dimensional Euclidean coordinate system.
2. The GEOGRAPHY Type: The GEOGRAPHY datatype's functions are the same as with GEOMETRY. The difference between the two is that when you specify GEOGRAPHY, you are usually specifying points in terms of latitude and longitude.
3. New Date and Time Datatypes: SQL Server 2008 introduces four new datatypes related to date and time: DATE, TIME, DATETIMEOFFSET, and DATETIME2.
1. DATE: The new DATE type just stores the date itself. It is based on the Gregorian calendar and handles years from 1 to 9999.
2. TIME: The new TIME (n) type stores time with a range of 00:00:00.0000000 through 23:59:59.9999999. The precision is allowed with this type. TIME supports seconds down to 100 nanoseconds. The n in TIME (n) defines this level of fractional second precision, from 0 to 7 digits of precision.
3. The DATETIMEOFFSET Type: DATETIMEOFFSET (n) is the time-zone aware version of a datetime datatype. The name will appear less odd when you consider what it really is: a date + a time + a time-zone offset. The offset is based on how far behind or ahead you are from Coordinated Universal Time (UTC) time.

4. The DATETIME2 Type: It is an extension of the datetime type in earlier versions of SQL Server. This new datatype has a date range covering dates from January 1 of year 1 through December 31 of year 9999. This is a definite improvement over the 1753 lower boundary of the datetime datatype. DATETIME2 not only includes the larger date range, but also has a timestamp and the same fractional precision that TIME type provides

173. *What are the Advantages of using CTE?*

1.Using CTE improves the readability and makes maintenance of complex queries easy.
2. The query can be divided into separate, simple, logical building blocks which can be then used to build more complex CTEs until final result set is generated.
3. CTE can be defined in functions, stored procedures, triggers or even views.
4. After a CTE is defined, it can be used as a Table or a View and can SELECT, INSERT, UPDATE or DELETE Data.

74. *How would you handle error in SQL SERVER 2008?*

SQL Server now supports the use of TRY...CATCH con handling. TRY...CATCH lets us build error handling at the level we need, in the way w to, by setting a region where if any error occurs, it will break out of the region and head to an error handler.
The basic structure is as follows:
BEGIN TRY stmts..
END TRY BEGIN
CATCH stmts..
END CATCH

175. *What is Aggregate Functions?*

Aggregate functions perform a calculation on a set of values and return a single value. Aggregate functions ignore NULL values except COUNT function. HAVING clause is used, along with GROUP BY, for filtering query using aggregate values. Following functions are aggregate functions. AVG, MIN CHECKSUM_AGG, SUM, COUNT, STDEV, COUNT_BIG, STDEVP, GROUPING, VAR, MAX. VARP

176. *What do you mean by Table Sample?*

TABLESAMPLE allows you to extract a sampling of rows from a table in the FROM clause. The rows retrieved are random and they are not in any order. This sampling can be based on a percentage of number of rows. You can use TABLESAMPLE when only a sampling of rows is necessary for the application instead of a full result set.

177. What is the difference between UNION and UNION ALL?

1. UNION The UNION command is used to select related information from two tables, much like the JOIN command. However, when using the UNION command all selected columns need to be of the same data type. With UNION, only distinct values are selected.

***2**. UNION ALL The UNION ALL command is equal to the UNION command, except that UNION ALL selects all values.*

The difference between Union and Union all is that Union all will not eliminate duplicate rows, instead it just pulls all rows from all tables fitting your query specifics and combines them into a table.

178. *What is B-Tree?*

The database server uses a B-tree structure to organize index information. B-Tree generally has following types of index pages or nodes:

1. *root node: A root node contains node pointers to branch nodes which can be only one.*
2. *branch node: A branch node contains pointers to leaf nodes or other branch nodes which can be two or more.*
3. *leaf nodes: A leaf node contains index items and orizantal pointers to other leaf nodes which can be many.*

179. What is a foreign key, and what is it used for?

A foreign key is used to establish relationships among relations in the relational model. Technically, a foreign key is a column (or columns) appearing in one relation that is (are) the primary key of another table. Although there may be exceptions, the values in the foreign key columns usually must correspond to values existing in the set of primary key values. This correspondence requirement is created in a database using a referential integrity constraint on the foreign key.

180. What does it mean when we say that a relation is in Boyce-Codd Normal Form (BCNF)?

A relation is in BCNF when every determinant in the relation is a candidate key. This means that any possible primary key can determine all other attributes in the relation. Attributes may not be determined by non-candidate key attributes or part of a composite candidate key. Thus it is said "I swear to construct my tables so that all nonkey columns are dependent on the key, the whole key and nothing but the key, so help me Codd!"

181. You have been given a set of tables with data and asked to create a new database to store them. When you examine the data values in the tables, what are you looking for?

(1) Multivalued dependencies, (2) Functional dependencies, (3) Candidate keys, (4) Primary keys and (5) Foreign keys.

182. *Explain the difference between attributes and identifiers.*

Entities have attributes. Attributes are properties that describe the entity's characteristics. Entity instances have identifiers. Identifiers are attributes that name, or identify, entity instances.

183. *Name and describe three types of binary relationships.*

1:1 - a single-entity instance of one type is related to a single-entity instance of another type. 1:N - a single entity instance of one type is related to many-entity instances of another type. M:N - many-entity instances of one type relate to many-entity instances of another type.

184. *What are stored procedures, and how do they differ from triggers?*

A stored procedure is a program that is stored within the database and is compiled when used. They can receive input parameters and they can return results. Unlike triggers, their scope is database-wide; they can be used by any process that has permission to use the database stored procedure.

185. *What are the advantages of using stored procedures?*

The advantages of stored procedures are (1) greater security, (2) decreased network traffic, (3) the fact that SQL can be optimized and (4) code sharing which leads to less work, standardized processing, and specialization among developers.

186. *What is the relationship of ODBC, OLE DB, and ADO?*

Developed first, the ODBC standard is for relational databases; while the OLE DB standard provides functionality for both relational and other databases. Finally, ADO was developed
to provide easier access to OLE DB data for the non-object-oriented programmer.

187. *Explain the differences between structured data and unstructured data.*

Structured data are facts concerning objects and events. The most important structured data are numeric, character, and dates. Structured data are stored in tabular form. Unstructured data are multimedia data such as documents, photographs, maps, images, sound, and video clips. Unstructured data are most commonly found on Web servers and Web-enabled databases.

188.What are dimension tables and definition of Fact tables?

These two questions are most commonly asked database interview questions. Fact tables are mainly central tables that are an integral part of data warehousing and dimension tables are used for describing the attributes of the fact tables. Both of these tables are important and play an important role in maintaining the database management system.

189.What is Data Warehouse?

A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection

of data in support of management's decision making process.

Subject-Oriented: A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.

Integrated: A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.

Time-Variant: Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.

Non-volatile: Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

190.What is Data Mining?

Data mining is a term from computer science. Sometimes it is also called knowledge discovery in databases (KDD). Data mining is about finding new information in a lot of data. The information obtained from data mining is hopefully both new and useful.

MOST FREQUENTLY ASKED SQL QUERIES

1.SQL Query to find second highest salary of Employee

Answer : There are many ways to find second highest salary of Employee in SQL, you can either use SQL Join or Subquery to solve this problem. Here is SQL query using Subquery :

```
1. select MAX(Salary) from Employee WHERE Salary NOT IN (select MAX(Salary) from Employee );
```

2. SQL Query to find Max Salary from each department. Answer :

```
SELECT DeptID, MAX(Salary) FROM Employee GROUP BY Dep
```

3.Write SQL Query to display current date.

Ans:SQL has built in function called GetDate() which returns current timestamp.

```
SELECT GetDate();
```

4.Write an SQL Query to check whether date passed to Query is date of given format or not. Ans: SQL has IsDate() function which is used to check passed value is date or not of
specified format ,it returns 1(true) or 0(false) accordingly.

```
SELECT ISDATE('1/08/13') AS "MM/DD/YY
```

It will return 0 because passed date is not in correct format.

5. Write a SQL Query to print the name of distinct employee whose DOB is between 01/01/1960 to 31/12/1975.

Ans:

```
SELECT DISTINCT EmpName FROM Employees WHERE DOB BETWEEN '01/01/1960' AND
31/12/1975
```

6.Write an SQL Query find number of employees according to gender whose DOB is between 01/01/1960 to 31/12/1975.

Answer : SELECT COUNT(*), sex from Employees WHERE DOB BETWEEN '01/01/1960 ' AND '31/12/1975' GROUP BY sex;

7.Write an SQL Query to find employee whose Salary is equal or greater than 10000.

Answer : SELECT EmpName FROM Employees WHERE Salary>=10000;

8.Write an SQL Query to find name of employee whose name Start with 'M'

*Ans: SELECT * FROM Employees WHERE EmpName like 'M%';*

9. find all Employee records containing the word "Joe", regardless of whether it was stored as JOE, Joe, or joe.

*Answer : SELECT * from Employees WHERE upper(EmpName) like upper('joe%');*

10. *Write a SQL Query to find year from date.*

Answer : SELECT YEAR(GETDATE()) as "Year";

11.To fetch ALTERNATE records from a table. (EVEN NUMBERED) *select * from emp where rowid in (select decode(mod(rownum,2),0,rowid, null) from emp);*

12.To select ALTERNATE records from a table. (ODD NUMBERED) *select * from emp where rowid in (select decode(mod(rownum,2),0,null ,rowid) from emp);*

13.Find the 3rd MAX salary in the emp table*. select distinct sal from emp e1 where 3 = (select count(distinct sal) from emp e2 where e1.sal <= e2.sal);*

14.Find the 3rd MIN salary in the emp table*. select distinct sal from emp e1 where 3 = (select count(distinct sal) from emp e2where e1.sal >= e2.sal);*

15.Select FIRST n records from a table*. select * from emp where rownum <= &n;*

16.Select LAST n records from a table *select * from emp minus select * from emp where rownum <= (select count(*) - &n from emp);*

17.List dept no., Dept name for all the departments in which there are no employees in the department.

*select * from dept where deptno not in (select deptno from emp);*
*alternate solution: select * from dept a where not exists (select * from emp b where a.deptno = b.deptno);*
altertnate solution: select empno,ename,b.deptno,dname from emp a, dept b where a.deptno(+) = b.deptno and empno is null;

18.How to get 3 Max salaries ? select distinct sal from emp a where 3 >= (select count(distinct sal) from emp b where a.sal <= b.sal) order by a.sal desc;

19.How to get 3 Min salaries ?

select distinct sal from emp a where 3 >= (select
count(distinct sal) from emp b where a.sal >= b.sal);

20.How to get nth max salaries? *select distinct hiredate from emp a where &n = (select count(distinct sal) from emp b where a.sal >= b.sal);*

21.Select DISTINCT RECORDS from emp table.
*select * from emp a where rowid = (select max(rowid) from emp b where a.empno=b.empno);*

22.How to delete duplicate rows in a table? *delete from emp a where rowid != (select max(rowid) from emp b where a.empno=b.empno);*

23.Count of number of employees in department wise.
select count(EMPNO), b.deptno, dname from emp a, dept b where a.deptno(+)=b.deptno group by b.deptno,dname;

24. Suppose there is annual salary information provided by emp table. How to fetch monthly salary of each and every employee*? select ename,sal/12 as monthlysal from emp;*

25.Select all record from emp table where deptno =10 or 40.
*select * from emp where deptno=30 or deptno=10;*

26.Select all record from emp table where deptno=30 and sal>1500.
*select * from emp where deptno=30 and sal>1500;*

27.Select all record from emp where job not in SALESMAN or CLERK.
*select * from emp where job not in ('SALESMAN','CLERK');*

28. Select all record from emp where ename in 'BLAKE','SCOTT','KING'and'FO
RD'.
select * from emp where ename

in('JONES','BLAKE','SCOTT','KING','FORD');
select * from emp where ename like'S____';
29. Select all records where ename starts with 'S' and its lenth is 6 ch

30. Select all records where ename may be any no of character but it should end with
'R'.
select * from emp where ename like'%R';

31. Count MGR and their salary in emp table.
select ename,(sal+nvl(comm,0)) as totalsal from emp;

33. Select any salary <3000 from emp table.
*select * from emp where sal> any(select sal from emp where sal<3000);*

34. Select all salary <3000 from emp table.
*select * from emp where sal> all(select sal from emp where sal<3000);*

35. Select all the employee group by deptno and sal in descending order.
select ename,deptno,sal from emp order by deptno,sal desc;

36. How can I create an empty table emp1 with same structure as emp?
*Create table emp1 as select * from emp where 1=2;*

37.How to retrive record where sal between 1000 to 2000?
*Select * from emp where sal>=1000 And sal<2000*

38.Select all records where dept no of both emp and dept table matches.
*select * from emp where exists(select * from dept where emp.deptno=dept.deptno)*

39.If there are two tables emp1 and emp2, and both have common record. How canI fetc h all the recods but common records only once?
(Select * from emp) Union (Select * from emp1)

40.How to fetch only common records from two tables emp and emp1?

(Select * from emp) Intersect (Select * from emp1)

41. How can I retrive all records of emp1 those should not present in emp2?
(Select * from emp) Minus (Select * from emp1)

42.Count the totalsa deptno wise where more than 2 employees exist.
SELECT deptno, sum(sal) As totalsal
FROM emp
GROUP BY deptno
HAVING COUNT(empno) > 2

43.Display the names of employees who are working in the company for the past 5 years
. select ename from emp where sysdate-hiredate>5*365;

44.Display the list of employees who have joined the company before 30th June 90 or after 31st dec 90

.select * from emp where hiredate between '30-jun-1990' and '31-dec-1990';

45.Display the names of employees working in department number 10 or 20 or 40 or employees working as clerks, salesman or analyst.

select ename from emp where deptno in (10,20,40) or job in ('CLERK','SALESMAN','ANALYST');

46.Display the names of employees whose name starts with alphabet S

select ename from emp where ename like 'S%';

47.Display employee names for employees whose name ends with alphab

select ename from emp where ename like '%S';

1. What is the difference between "Stored Procedure" and "Function"?

1. *A procedure can have both input and output parameters, but a function can only have input parameters.*
2. *Inside a procedure we can use DML (INSERT/UPDATE/DELETE) statements. But inside a function we can't use DML statements.*
3. *We can't utilize a Stored Procedure in a Select statement. But we can use a function in a Select statement.*
4. *We can use a Try-Catch Block in a Stored Procedure but inside a function we can't use a Try-Catch block.*
5. *We can use transaction management in a procedure but we can't in a function.*
6. *We can't join a Stored Procedure but we can join functions.*
7. *Stored Procedures cannot be used in the SQL statements anywhere in the WHERE/HAVING/SELECT section. But we can use a function anywhere.*
8. *A procedure can return 0 or n values (max 1024). But a function can return only 1 value that is mandatory.*
9. *A procedure can't be called from a function but we can call a function from a procedure.*

2.What is difference between "Clustered Index" and "Non Clustered Index"?

1. A Clustered Index physically stores the data of the table in the order of the keys values and the data is resorted every time whenever a new value is inserted or a value is updated in the column on which it is defined, whereas a non-clustered index creates a separate list of key values (or creates a table of pointers) that points towards the location of the data in the data pages.

2. A Clustered Index requires no separate storage than the table storage. It forces the rows to be stored sorted on the index key whereas a non-clustered index requires separate storage than the table storage to store the index information.
3. A table with a Clustered Index is called a Clustered Table. Its rows are stored in a BTree structure sorted whereas a table without any clustered indexes is called a nonclustered table. Its rows are stored in a heap structure unsorted.
4. The default index is created as part of the primary key column as a Clustered Index.
5. In a Clustered Index, the leaf node contains the actual data whereas in a nonclustered index, the leaf node contains the pointer to the data rows of the table.
6. A Clustered Index always has an Index Id of 1 whereas non-clustered indexes have Index Ids > 1.
7. A Table can have only 1 Clustered Index whereas prior to SQL Server 2008 only 249 non-clustered indexes can be created. With SQL Server 2008 and above 999 nonclustered indexes can be created.
8. A Primary Key constraint creates a Clustered Index by default whereas A Unique Key constraint creates a non-clustered index by default.

3.What is the difference between the "DELETE" and "TRUNCATE" commands?
*1. The **DELETE** command is used to remove rows from a table based on a **WHERE** condition whereas TRUNCATE removes all rows from a table.*
2. So we can use a where clause with DELETE to filter and delete specific records whereas we cannot use a Where clause with TRUNCATE.
*3. **DELETE** is executed using a row lock, each row in the table is locked for deletion whereas **TRUNCATE** is executed using a table lock and the entire table is locked for removal of all records.*
*4. **DELETE** is a DML command whereas **TRUNCATE** is a DDL command.*
*5. **DELETE** retains the identity of the column value whereas in **TRUNCATE**, the Identify column is reset to its seed value if the table contains any identity column.*
6. To use Delete you need DELETE permission on the table whereas to use Truncate on a table you need at least ALTER permission on the table.
*7. **DELETE** uses more transaction space than the **TRUNCATE** statement whereas Truncate uses less transaction space than DELETE statement.*
*8. **DELETE** can be used with indexed views whereas **TRUNCATE** cannot be used with indexed views.*
*9. The **DELETE** statement removes rows one at a time and records an entry in the transaction log for each deleted row whereas **TRUNCATE TABLE** removes the data by deallocating the data pages used to store the table data and records only the page deallocations in the transaction log.*
*10. Delete activates a trigger because the operation is logged individually whereas **TRUNCATE TABLE** can't activate a trigger because the operation does not log individual row deletions.*

4. What is the difference between the "WHERE" clause and the "HAVING" clause?

1. *WHERE clause can be used with a Select, Update, and Delete Statement Clause but the HAVING clause can be used only with a Select statement.*
2. *We can't use an aggregate functions in the WHERE clause unless it is in a sub-query contained in a HAVING clause whereas we can use an aggregate function in the HAVING clause. We can use a column name in the HAVING clause but the column must be contained in the group by clause.*
3. *WHERE is used before the GROUP BY clause whereas a HAVING clause is used to impose a condition on the GROUP Function and is used after the GROUP BY clause in the query.*
4. *A WHERE clause applies to each and every row whereas a HAVING clause applies to summarized rows (summarized with GROUP BY).*
5. *In the WHERE clause the data that is fetched from memory depends on a condition whereas in HAVING the completed data is first fetched and then separated depending on the condition.*

5. What is the difference between "Primary Key" and "Unique Key"?

1. We can have only one Primary Key in a table whereas we can have more than one Unique Key in a table.
2. The Primary Key cannot have a NULL value whereas a Unique Key may have only one null value.
3. By default, a Primary Key is a Clustered Index whereas by default, a Unique Key is a unique non-clustered index.
4. A Primary Key supports an Auto Increment value whereas a Unique Key doesn't support an Auto Increment value.

6. What is the difference between a "Local Temporary Table" and a "Global Temporary Table"?

1. A Local Temporary Table is created by giving it a prefix of # whereas a Global Temporary Table is created by giving it a prefix of ##.
2. A Local Temporary Table cannot be shared among multiple users whereas a Global Temporary Tables can be shared among multiple users.
3. A Local Temporary Table is only available to the current DB connection for the current user and are cleared when the connection is closed whereas a Global
4. Temporary Table is available to any connection once created. They are cleared when the last connection is closed.

7.What are super, primary, candidate and foreign keys?

*Ans: A **super key** is a set of attributes of a relation schema upon which all attributes of the schema are functionally dependent. No two rows can have the same value of super key attributes.*
*A **Candidate key** is minimal super key, i.e., no proper subset of Candidate key attributes can be a super key.*
*A **Primary Key** is one of the candidate keys. One of the candidate keys is selected as most important and becomes the primary key. There cannot be more that one primary keys in a table.*
***Foreign key** is a field (or collection of fields) in one table that uniquely identifies a row of another table.*

8.What is the difference between primary key and unique constraints?
Ans: Primary key cannot have NULL value, the unique constraints can have NULL values. There is only one primary key in a table, but there can be multiple unique constrains.

Introduction to DAX in PowerBI

- Overview: Data Analysis Expressions (DAX) is a formula language primarily used in Power Pivot, Analysis Services, and Power BI. It enables users to harness the power of dynamic calculations and deep data insights.

- Importance: By enabling custom calculations, DAX becomes an essential tool for data professionals striving for deeper data insights.

DAX Fundamentals

- Syntax Distinctions: While bearing similarities to Excel, DAX offers a more powerful and dynamic formula structure.

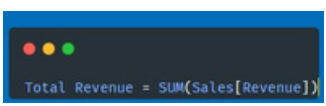
- Data Types: Beyond basic types like Decimal and Integer, DAX introduces complex types that support intricate data operations.

- Contexts: Grasping the concept of Row and Filter Context is vital for accurate data representation and calculation.

Core DAX Functions – Aggregation

- Key Functions: These are the cornerstone of data analysis. Functions like SUM, AVERAGE, and COUNT allow for primary data aggregation.

1. **SUM**
- The `SUM` function calculates the sum of a column's values. -
Example:
- Given a table `Sales` with a column `Revenue`.
- DAX Formula:



- This formula will calculate the total revenue from all sales.

2. **AVERAGE**
- The `AVERAGE` function returns the average (arithmetic mean) of a column's values. -
Example:
- Given the same `Sales` table with a column `Revenue`.
- DAX Formula:

- This formula will calculate the average revenue from all sales.

3. **COUNT**
- The `COUNT` function counts the number of rows in a table where the values in the specified column are not blank.
- Example:
- Using a table `Orders` with a column `OrderID`.
- DAX Formula: `Total Orders = COUNT(Orders[OrderID])`
- This formula will count the total number of orders.

4. **MIN & MAX**
- The `MIN` and `MAX` functions return the smallest and largest values in a column, respectively.
- Example:
- Using the `Sales` table with a column `Revenue`.
- DAX Formula for Minimum Revenue: `Min Revenue = MIN(Sales[Revenue])`
- DAX Formula for Maximum Revenue: `Max Revenue = MAX(Sales[Revenue])`
- These formulas will find the smallest and largest revenue values, respectively.

5. **COUNTA**
- The `COUNTA` function counts the number of rows in a table where the values in the specified column are not blank, and it works on non-numeric data as well.
- Example:
- Using a table `Customers` with a column `Name`.
- DAX Formula: `Total Customers = COUNTA(Customers[Name])`
- This formula will count the total number of customers with a valid name.

Core DAX Functions - Date and Time

- Temporal Calculations: With functions such as DATE, NOW, and TODAY, DAX provides robust support for time-based data operations.

1. **DATE**
- The `DATE` function returns a date in date-time format.
- Example:
- If you want to create a date from year, month, and day values.
- DAX Formula: `Custom Date = DATE(2023, 8, 29)`
- This formula will return a date value of August 29, 2023.

2. **NOW**
- The `NOW` function returns the current date and time.
- Example:
- DAX Formula: `Current DateTime = NOW()`
- This formula will show the current date and time at the moment of execution.

3. **TODAY**
- The `TODAY` function returns the current date.
- Example:
- DAX Formula: `Current Date = TODAY()`
- This formula will return the current date without the time component.

4. **MONTH**
- The `MONTH` function returns the month as a number (1 for January, 2 for February, etc.) from a date.
- Example:
- Given a date column `Sales[Date]`.
- DAX Formula: `Sale Month = MONTH(Sales[Date])`
- This formula will extract the month number from the sale date.

5. **YEAR**
- The `YEAR` function extracts the year from a date.
- Example:
- Given a date column `Sales[Date]`.
- DAX Formula: `Sale Year = YEAR(Sales[Date])`
- This formula will provide the year of the sale date.

6. **DATEDIFF**
- The `DATEDIFF` function returns the difference between two dates, based on a specified interval (day, month, year, etc.).
- Example:
- Given two date columns `Orders[StartDate]` and `Orders[EndDate]`.
- DAX Formula: `Duration in Days = DATEDIFF(Orders[StartDate], Orders[EndDate], DAY)`
- This formula will calculate the number of days between the start and end dates.

Advanced DAX Concepts

- Dynamic Context: Functions like CALCULATE and CALCULATETABLE give users the power to modify existing contexts and create new ones.

- Time Intelligence: Explore historical data trends with functions like DATESYTD and TOTALYTD.

1. **Dynamic Context**
- Dynamic context in DAX enables the modification of existing filter contexts or the creation of new ones.

- **CALCULATE**
- This function evaluates an expression in a modified filter context.
- Example:
- Given a `Sales` table and a need to calculate total sales only for a specific product category, say "Electronics".
- DAX Formula: `
Electronics Sales = CALCULATE(SUM(Sales[Revenue]), Sales[Category] = "Electronics")`
- This formula calculates the total revenue for only the "Electronics" category.

- **CALCULATETABLE**
- This function evaluates a table expression in a modified filter context.
- Example:
- If you want to produce a table that only includes sales data for the year 2023. - DAX Formula: `Sales 2023 = CALCULATETABLE(Sales, YEAR(Sales[Date]) = 2023)`

- This formula provides a table containing sales data exclusively from the year 2023.
2. **Time Intelligence**
- Time intelligence functions in DAX are critical for analyzing data over time, especially for understanding historical trends and making future projections.

- **DATESYTD**
- Returns dates from the beginning of the year to a specified date.
- Example:
- If you're analyzing sales data and need to consider dates from the start of the year to the current date.
- DAX Formula: `Year to Date = DATESYTD(Sales[Date])`
- This formula gives a list of dates from the start of the year to the present date.

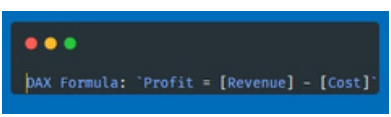
- **TOTALYTD**
- Calculates the cumulative total for a measure from the beginning of the year to a specified date.
- Example:
- To compute the cumulative sales from the start of the year to the current date.
- DAX Formula: `Cumulative Sales = TOTALYTD(SUM(Sales[Revenue]), Sales[Date])`
- This formula calculates the total revenue accumulated from the start of the year to the present date.

DAX in Action

- Calculated Columns: Enhance your data models by defining new columns using DAX.

Calculated columns allow you to add new data to your tables in Power BI. The formula for the column is calculated for each row of data.

1. **Profit Column***
- Let's say you have a sales table with columns "Revenue" and "Cost." You can create a new calculated column called "Profit" using DAX.



2. **Full Name Column**
- If you have a table with "First Name" and "Last Name" columns, you can combine them into a "Full Name" column.
- DAX Formula: `Full Name = [First Name] & " " & [Last Name]`

3. **Age Group Column**
- Given a column "Age," you can categorize individuals into age groups.
- DAX Formula:
```
Age Group =
IF([Age] < 18, "Child",
IF([Age] < 60, "Adult", "Senior"))
```

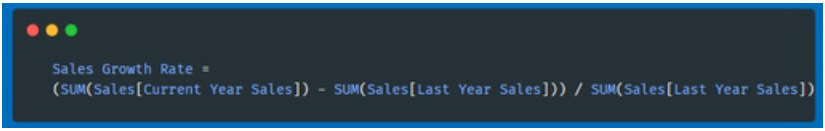
- Measures: These custom aggregations, defined using DAX, provide more profound insights into datasets.

Measures perform calculations on data based on user interactions in reports. They aggregate data as users interact with visuals.

1. **Total Revenue Measure**
- Sum up the revenue from a "Revenue" column in the sales table.
- DAX Formula: `Total Revenue = SUM(Sales[Revenue])`

2. **Average Monthly Sales Measure**
- Calculate the average sales for each month.
- DAX Formula: `Avg Monthly Sales = AVERAGE(Sales[Monthly Sales])`

3. **Sales Growth Rate Measure**
- If you have "Current Year Sales" and "Last Year Sales" columns, you can compute the growth rate.
- DAX Formula:



- KPIs: Monitor business health and trajectory using DAX-defined Key Performance Indicators.

KPIs (Key Performance Indicators)

KPIs provide a clear view of current performance and how it stacks against desired outcomes.

1. **Monthly Sales Target KPI**
- Let's say you have a target of $100,000 in sales every month.
- DAX Formula for Actual Sales: `Total Sales = SUM(Sales[Revenue])`
- DAX KPI Definition: If `Total Sales` is greater than $100,000, the indicator is green. If it's between $90,000 and $100,000, it's yellow. Otherwise, it's red.

2. **Customer Satisfaction KPI**
- You've conducted a survey, and your goal is to achieve a satisfaction rating of above 4.5 out of 5.
- DAX Formula for Average Rating: `Avg Rating = AVERAGE(Survey[Rating])`
- DAX KPI Definition: If `Avg Rating` is greater than 4.5, the indicator is green. If it's between 4.0 and 4.5, it's yellow. Otherwise, it's red.

3. **Inventory Turnover KPI**
- You aim to have an inventory turnover ratio of at least 6 times a year, ensuring products are sold and replaced efficiently.

- DAX Formula: `Turnover Ratio = SUM(Sales[Items Sold]) / AVERAGE(Inventory[Items in Stock])`
- DAX KPI Definition: If `Turnover Ratio` is greater than or equal to 6, the indicator is green. If it's between 5 and 6, it's yellow. Otherwise, it's red.

Performance and Optimization

- **Efficiency Tips:** Writing optimal DAX formulas ensures faster report rendering.
- **Monitoring Techniques:** Using query plans and server timings can help pinpoint and resolve performance bottlenecks.

Best Practices in DAX

- **Naming Conventions:** Adopting a consistent naming strategy enhances formula clarity and maintainability.
- **Documentation:** Always document complex formulas for future reference and team collaboration.
- **Structure:** Decompose intricate formulas into smaller components for better readability and debugging.

RoadMap to Mastering Generative AI

To develop proficiency in Generative AI, the following 5 skills are essential for a comprehensive understanding and practical application of the field.

Fundamentals of Machine Learning and Deep Learning:

Understand the basics of machine learning algorithms, such as supervised and unsupervised learning.

Gain knowledge of deep learning concepts, including neural networks, activation functions, and optimization techniques.

Familiarize yourself with common deep learning frameworks like TensorFlow or PyTorch.

Probability and Statistics:

Acquire a strong foundation in probability theory, including concepts like random variables, probability distributions, and Bayesian inference.

Natural Language Processing (NLP):

Learn the basics of NLP, including tokenization, text preprocessing, and language modeling.

Understand techniques for text generation, such as recurrent neural networks (RNNs) and sequence-to-sequence models.

Explore advanced topics in NLP, such as attention mechanisms, transformer models (e.g., GPT, BERT), and language generation.

Computer Vision:

Develop a solid understanding of computer vision concepts, including image classification, object detection, and image segmentation.

Learn about deep learning architectures for computer vision tasks, such as convolutional neural networks (CNNs).

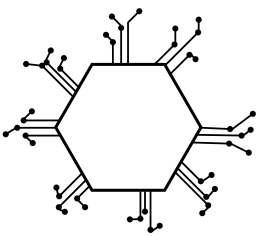
Gain hands-on experience with popular computer vision frameworks like OpenCV and PyTorch's torchvision.

Generative Adversarial Networks (GANs):

Study the fundamentals of GANs, including their architecture and training process.

Explore various GAN variants, such as conditional GANs, cycle-consistent GANs, and style transfer GANs.

Gain practical experience by implementing GANs for tasks like image synthesis, style transfer, and text-to-image generation.



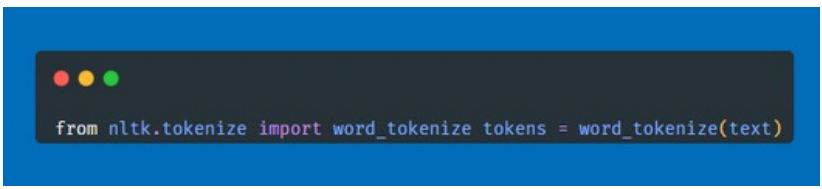
NLP

Cheat Sheet

1. Tokenization

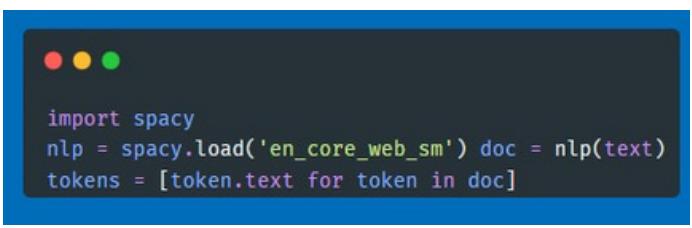
Tokenization is the process of breaking up text into words, phrases, symbols, or other meaningful elements, which are called tokens.

- NLTK Word Tokenization:



```
from nltk.tokenize import word_tokenize tokens = word_tokenize(text)
```

- Spacy Word Tokenization:

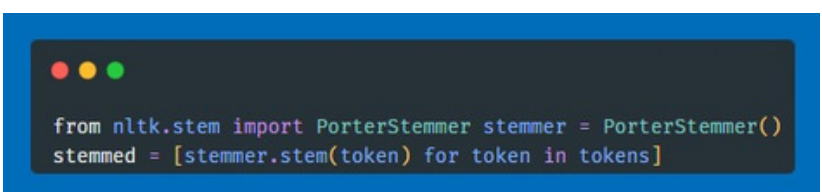


```
import spacy
nlp = spacy.load('en_core_web_sm') doc = nlp(text)
tokens = [token.text for token in doc]
```

2. Stemming and Lemmatization

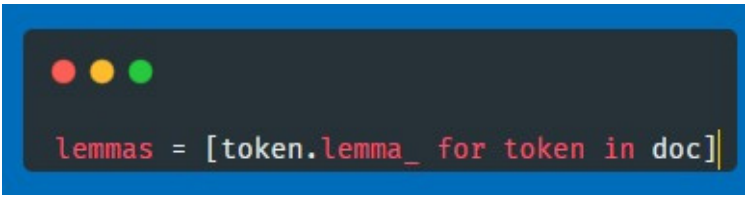
Stemming and Lemmatization are techniques used to extract the base form of the words by removing its inflection.

- NLTK Stemming:



```
from nltk.stem import PorterStemmer stemmer = PorterStemmer()
stemmed = [stemmer.stem(token) for token in tokens]
```

- Spacy Lemmatization:

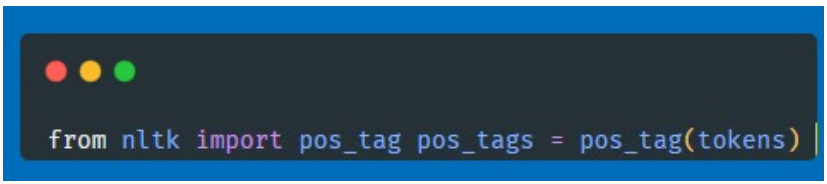


```
lemmas = [token.lemma_ for token in doc]
```

3. Part-of-Speech (POS) Tagging

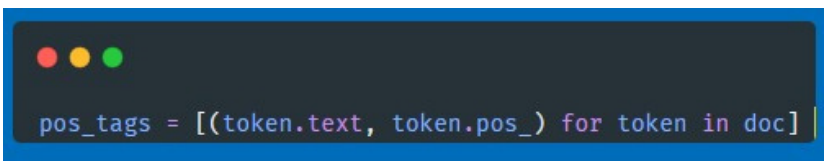
POS tagging is the task of labeling the words in a sentence with their appropriate part of speech.

- NLTK POS Tagging:



```
from nltk import pos_tag pos_tags = pos_tag(tokens)
```

- Spacy POS Tagging:

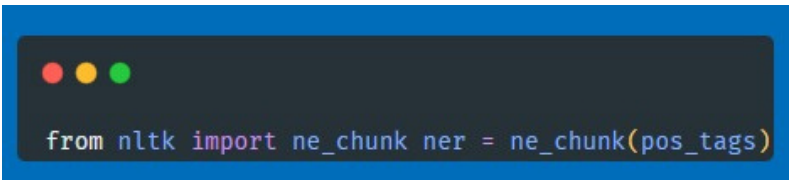


```
pos_tags = [(token.text, token.pos_) for token in doc]
```

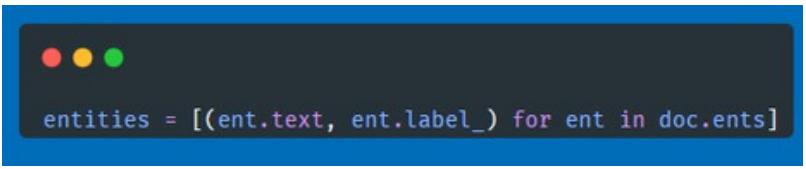
4. Named Entity Recognition (NER)

NER is the process of locating named entities in text and classifying them into predefined categories.

- NLTK NER:



- Spacy NER:

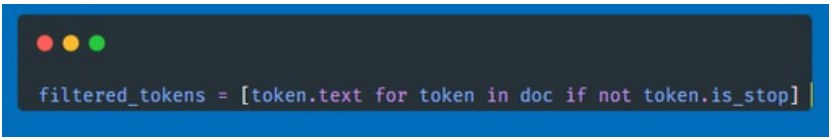


5. Stopword Removal

Stopwords are the most common words in a language that are to be filtered out before processing the text data.

- NLTK Stopword Removal:

- Spacy Stopword Removal:

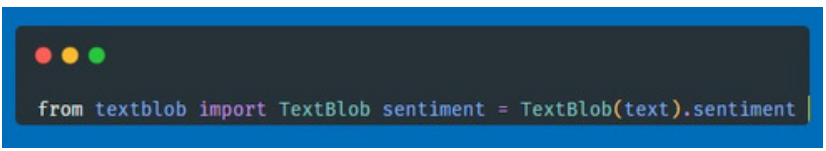


```
filtered_tokens = [token.text for token in doc if not token.is_stop]
```

6. Sentiment Analysis

Sentiment Analysis is the process of determining the sentiment or emotion of a piece of text.

- TextBlob Sentiment Analysis:



```
from textblob import TextBlob sentiment = TextBlob(text).sentiment
```

7. Topic Modeling

Topic Modeling is the process of identifying topics in a set of documents.

- Gensim LDA Topic Modeling:

```
from gensim import corpora, models dictionary = corpora.Dictionary(docs) corpus =
[dictionary.doc2bow(doc) for doc in docs]
lda_model = models.LdaModel(corpus, num_topics=4, id2word=dictionary)
```

Parameters of OPENAI GPT Models

One

Imagine you're playing a game where you have to come up with words that start with the letter 'A'. You can think of many words, like 'apple', 'ant', 'alligator', and so on. But, you want to make the game more interesting, so you add some rules.

Temperature:

This rule decides how creative or unusual the words you come up with can be. If the temperature is low, you'll mostly come up with common words like 'apple' or 'ant'. But if the temperature is high, you might come up with more unusual words like 'abacus' or 'aardvark'.

Two

Top_P

This rule decides how many of the most likely words you can choose from. If top_p is 5, then you can only choose from the 5 most likely words that start with 'A'. If top_p is 10, then you can choose from the 10 most likely words. More the choice, the chances of randomness increases

Three

Frequency Penalty:

This rule decides how often you can use the same word. If the frequency penalty is high, then you can't use the same word too many times. So if you've already used the word 'apple' a few times, it'll become less likely for you to use it again.

Four

Presence Penalty (topic)

This rule decides how closely your words have to be related to a certain topic. If the topic penalty is high and the topic is ‘animals’, then you’ll mostly come up with animal names that start with ‘A’, like ‘ant’ or ‘alligator’. But if the topic penalty is low, then you might also come up with non-animal words like ‘apple’ or ‘airplane’.

AIGuild Premium Community

🎁 Worth $1500/-+ Value of content and updated Weekly

Scan to Access

- ✅ Recording of all workshops, meetups and webinars
- ✅ Doubt Clearing Sessions
- ✅ 4 Live Workshops every month
- ✅ 50% Discount to Live programs
- ✅ 30 mins One on One Mentoring Session with me for all Yearly Premium Members

Long Term Commitment has more benefits and value.

Get value worth $1500+ for just $12 / Month

https://nas.io/aiguild

FOR ACTION TAKERS FIRST 5 USERS USE YEARLY20TO GET 20% OFF FOR 6.4$ / month (if Paid yearly 77$ upfront)

What Does The Community Provide?

Gen AI Courses

- ✅ Generative AI (chatGPT) for Business
- ✅ Prompt Engineering for Developers

Recordings

- ✅ Outcome-based Workshops
- ✅ AI Community Meetup Recordings
- ✅ Python Projects Videos
- ✅ AI & DS Career & Learning Webinar Series

Data Science Courses

- ✅ Basic Excel For Data Science
- ✅ Basic SQL For AI/Data Science
- ✅ Basic Python for AI/Data Jobs
- ✅ Advanced Python for AI/DS Jobs
- ✅ Basic PowerBI for AI/Data Science

Resources

- ✅ Generative AI Resources
- ✅ Sample Datasets
- ✅ Ready to use Resume Template
- ✅ Linkedin Profile Optimization
- ✅ Essential SQL Documents
- ✅ Essential Python Documents
- ✅ Machine Learning Documents

Scan to Access

Mohammad Arshad

Principal Data Scientist | Strategy & Solutions | Generative AI | 18 Years+ Exp | Ex- MAF, Accenture, HP, Dell | Speaker & Mentor | AWS, Azure & GCP

Talks about #careers, #analytics, #datascience, #generativeai, and #artificialintelligence

Dubai, United Arab Emirates · Contact info

Best Community to learn Gen AI

40,118 followers · 500+ connections

mohammad@decodingdatascience.com

2023

Artificial Intelligence

2,200 members

Scan to Access