Level 1: Foundations of Python & Math for Data Science

Key Concepts:

- Python Basics: Variables, Data Types, Loops, Functions
- Object-Oriented Programming (OOP) Basics
- Numpy for Numerical Computing
- Pandas for Data Manipulation
- Basic Linear Algebra (Vectors, Matrices, Dot Product)
- Basic Statistics (Mean, Median, Mode, Variance, Standard Deviation)

Key Libraries, Classes & Functions:

- numpy: array, mean(), dot(), reshape()
- pandas: DataFrame, read_csv(), describe(), groupby()
- math: sqrt(), exp(), log()
- matplotlib.pyplot: plot(), hist(), scatter()

Project: Exploratory Data Analysis (EDA) on a Real Dataset

- **Dataset**: Titanic Dataset (https://www.kaggle.com/c/titanic/data)
- **Task**: Load the dataset, clean missing values, and analyze survival rates based on gender, age, and ticket class.

Level 2: Data Wrangling & Visualization

Key Concepts:

- Handling Missing Data
- Data Cleaning and Transformation
- Data Visualization (Histograms, Boxplots, Scatterplots)
- Feature Engineering Basics

Key Libraries, Classes & Functions:

- pandas: fillna(), dropna(), merge(), apply()

- seaborn: sns.boxplot(), sns.pairplot(), sns.heatmap()
- matplotlib.pyplot: subplot(), bar()

Project: Customer Segmentation Visualization

- **Dataset**: Mall Customer Segmentation Dataset (https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python)
- **Task**: Visualize spending behavior based on age, income, and spending score.

Level 3: Statistics & Probability for Data Science

Key Concepts:

- Probability Distributions (Normal, Binomial, Poisson)
- Hypothesis Testing (t-test, Chi-Square, ANOVA)
- Confidence Intervals

Key Libraries, Classes & Functions:

- scipy.stats: norm(), ttest_ind(), chi2_contingency()
- numpy.random: randn(), binomial()
- statsmodels.api: ols(), anova_lm()

Project: A/B Testing for Website Performance

- **Dataset**: Online A/B Testing Dataset (https://www.kaggle.com/zhangluyuan/ab-testing)
- **Task**: Conduct hypothesis testing to determine if a new website design improves conversion rates.

Level 4: Machine Learning Fundamentals

Key Concepts:

- Supervised vs. Unsupervised Learning
- Train-Test Split, Cross-Validation
- Performance Metrics (Accuracy, Precision, Recall, F1-score)

Key Libraries, Classes & Functions:

- sklearn.model_selection: train_test_split(), cross_val_score()
- sklearn.metrics: classification_report(), confusion_matrix()
- sklearn.preprocessing: StandardScaler(), MinMaxScaler()

Project: Predicting House Prices

- **Dataset**: California Housing Prices (https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data)
- **Task**: Train a simple Linear Regression model to predict housing prices.

Level 5: Advanced Machine Learning (Regression & Classification)

Key Concepts:

- Linear & Logistic Regression
- Decision Trees, Random Forests, Support Vector Machines
- Hyperparameter Tuning (GridSearch, RandomizedSearch)

Key Libraries, Classes & Functions:

- sklearn.linear_model: LinearRegression(), LogisticRegression()
- sklearn.ensemble: RandomForestClassifier()
- sklearn.svm: SVC()

Project: Credit Card Fraud Detection

- **Dataset**: Credit Card Fraud Dataset (https://www.kaggle.com/mlg-ulb/creditcardfraud)
- **Task**: Build a classifier to detect fraudulent transactions.

Level 6: Unsupervised Learning & Dimensionality Reduction

Key Concepts:

- K-Means Clustering, DBSCAN
- Principal Component Analysis (PCA)
- Anomaly Detection

Key Libraries, Classes & Functions:

- sklearn.cluster: KMeans(), DBSCAN()
- sklearn.decomposition: PCA()

Project: Customer Segmentation Using Clustering

- **Dataset**: Wholesale Customers Dataset (https://archive.ics.uci.edu/ml/datasets/Wholesale+customers)
- **Task**: Cluster customers into different segments based on their purchase behavior.

Level 7: Deep Learning & Neural Networks

Key Concepts:

- Artificial Neural Networks (ANNs)
- Convolutional Neural Networks (CNNs)
- Activation Functions (ReLU, Sigmoid, Softmax)

Key Libraries, Classes & Functions:

- tensorflow.keras.models: Sequential()
- tensorflow.keras.layers: Dense(), Conv2D(), Flatten()
- tensorflow.keras.optimizers: Adam()

Project: Handwritten Digit Recognition (MNIST)

- **Dataset**: MNIST Dataset (https://www.kaggle.com/c/digit-recognizer/data)
- **Task**: Train a CNN to recognize handwritten digits.

Level 8: Natural Language Processing (NLP)

Key Concepts:

- Tokenization, Lemmatization, Stopwords
- Word Embeddings (Word2Vec, GloVe)
- Sentiment Analysis

Key Libraries, Classes & Functions:

- nltk: word_tokenize(), stopwords
- spacy: nlp()
- sklearn.feature_extraction.text: TfidfVectorizer()

Project: Sentiment Analysis on Movie Reviews

- **Dataset**: IMDB Reviews (https://www.kaggle.com/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews)
- **Task**: Train a model to classify reviews as positive or negative.

Level 9: Time Series Analysis

Key Concepts:

- Autoregressive (AR), Moving Average (MA), ARIMA
- Seasonal Patterns
- Forecasting Techniques

Key Libraries, Classes & Functions:

- statsmodels.tsa: ARIMA(), seasonal_decompose()
- pandas: rolling(), resample()

Project: Stock Price Prediction

- **Dataset**: Apple Stock Prices (https://www.kaggle.com/borismarjanovic/price-volume-data-for-all-us-stocks-etfs)
- **Task**: Forecast future stock prices using ARIMA.

Level 10: Big Data & Deploying Models

Key Concepts:

- Handling Large Datasets (Dask, Spark)
- Deploying ML Models (Flask, FastAPI)
- Model Monitoring & Optimization

Key Libraries, Classes & Functions:

- pyspark.sql: SparkSession()
- flask: Flask(), request
- dask: dataframe

Project: Deploying a Machine Learning Model as an API

- **Dataset**: Any dataset from previous projects
- **Task**: Train a model and deploy it using Flask or FastAPI.

Final Goal: Build an End-to-End Data Science Portfolio

- Select 2–3 projects, improve them, and deploy them on a personal website or GitHub.

This roadmap will take you from beginner to expert, giving you hands-on experience with real-world data. 🚀

Level 1: Foundations of Python & Math for Data Science

Project: Exploratory Data Analysis (EDA) on the Titanic Dataset

- **Python Basics (Variables, Data Types, Loops, Functions):** Used to load, process, and analyze the dataset (e.g., defining functions to compute survival rates).
- **OOP Basics:** Structuring data processing functions into classes for cleaner code organization.
- **NumPy for Numerical Computing:** Used for fast array operations and statistical calculations.
- **Pandas for Data Manipulation:** Reading the dataset (read_csv()), handling missing values (fillna()), and filtering passengers by criteria (e.g., groupby('Sex').mean()).
- **Basic Linear Algebra:** Understanding numerical relationships, such as computing survival probabilities via matrix operations.
- **Basic Statistics:** Calculating mean age, median fare, survival rates, and standard deviation to summarize the dataset.

Level 2: Data Wrangling & Visualization

Project: Customer Segmentation Visualization

- **Handling Missing Data:** Using fillna() and dropna() to manage missing customer demographics.
- **Data Cleaning & Transformation:** Formatting inconsistent values (e.g., standardizing age groups).
- **Data Visualization:** Creating histograms (plt.hist()) to analyze spending behavior, boxplots (sns.boxplot()) to identify income distribution, and scatterplots (plt.scatter()) to show age vs. spending score.
- **Feature Engineering:** Creating new features such as "age category" or "income brackets" to enhance segmentation.

Level 3: Statistics & Probability for Data Science

Project: A/B Testing for Website Performance

- **Probability Distributions:** Modeling user behaviors with normal and binomial distributions.
- **Hypothesis Testing:** Running ttest_ind() to compare conversion rates of old vs. new website design.
- **Confidence Intervals:** Estimating the range in which the true conversion rate lies with stats.norm.interval().

Level 4: Machine Learning Fundamentals

Project: Predicting House Prices

- **Supervised vs. Unsupervised Learning:** House price prediction is a supervised learning task.
- **Train-Test Split:** Using train_test_split() to divide data into training and testing sets.
- **Performance Metrics:** Evaluating the model’s accuracy using mean_absolute_error() and r2_score().

Level 5: Advanced Machine Learning (Regression & Classification)

Project: Credit Card Fraud Detection

- **Linear & Logistic Regression:** Initial models to classify fraudulent transactions.
- **Decision Trees & Random Forests:** More complex models to improve fraud detection accuracy.
- **Hyperparameter Tuning:** Using GridSearchCV() to optimize the Random Forest model.

Level 6: Unsupervised Learning & Dimensionality Reduction

Project: Customer Segmentation Using Clustering

- **K-Means Clustering:** Grouping customers based on spending habits.
- **PCA:** Reducing feature dimensions for better visualization.
- **Anomaly Detection:** Identifying outlier customers with unusual spending patterns.

Level 7: Deep Learning & Neural Networks

Project: Handwritten Digit Recognition (MNIST)

- **Artificial Neural Networks:** Training a model with Sequential() and Dense() layers.
- **Convolutional Neural Networks:** Using Conv2D() for image feature extraction.
- **Activation Functions:** Implementing ReLU() for non-linearity and Softmax() for multi-class classification.

Level 8: Natural Language Processing (NLP)

Project: Sentiment Analysis on Movie Reviews

- **Tokenization & Lemmatization:** Processing text with nltk.word_tokenize() and WordNetLemmatizer().
- **Word Embeddings:** Using TfidfVectorizer() to represent text numerically.
- **Sentiment Analysis:** Training a model to classify reviews as positive or negative.

Level 9: Time Series Analysis

Project: Stock Price Prediction

- **ARIMA Modeling:** Using ARIMA() to predict future stock prices.
- **Seasonal Patterns:** Decomposing trends with seasonal_decompose().
- **Rolling Windows:** Smoothing fluctuations with rolling().mean().

Level 10: Big Data & Deploying Models

Project: Deploying a Machine Learning Model as an API

- **Handling Large Datasets:** Using Dask or Spark to process big data efficiently.
- **Deploying ML Models:** Creating a REST API with Flask() to serve model predictions.
- **Model Monitoring:** Continuously tracking model performance using logging tools.