Machine Learning Modelling

1. Look at the Big Picture: Task and Initial Data Exploration

Define Your Task:
Build a model to predict housing prices using metrics like population, median income, and median housing price.

Initial Data Exploration:
Load data: **data = pd.read_csv("california_housing.csv")**.
Explore data: **data.head()**, **data.info()**, **data.describe()**, **data['category_column'].value_counts()**, **data.corr()**.

Visual Data Analysis:
Histograms and scatter plots: **data.hist()**, **data.plot(kind="scatter", x="longitude", y="latitude")**.

Setting Analysis Scope & Planning Further Analysis:
Identify target and predictors.
Consider feature engineering, handling missing values/outliers, and data transformation

2. Get the Data: Sourcing and Preparing Your Dataset

Identify Data Sources:
Look for reliable open datasets relevant to your project (examples include public repositories and data portals).

Data Acquisition:
Implement methods for downloading or accessing the data, such as scripts for web scraping or APIs for database access.

Organize Data Storage:
Create a structured directory system for storing the datasets.

Load Data for Analysis:
Use Python libraries (like pandas) to load the data into a suitable format for analysis: **data = pd.read_csv("your_dataset.csv")**.

3. Discover and Visualize Data for Gaining Insights

Data Visualization:
Utilize scatter plots to observe spatial relationships and patterns in data.

Attribute Correlation Analysis:
Investigate how different data attributes correlate with each other, potentially using correlation matrices.

Combining Attributes (Feature engineering):
Explore creating new informative features by combining existing attributes (e.g., calculating ratios or aggregate metrics).

4. Prepare Data for Machine Learning Algorithms

Handle Missing Values:
Employ strategies like imputation, removing rows, or column elimination.

Process Text and Categorical Attributes:
Convert text and categorical data into numerical formats through encoding methods. EX LabelEncoder(Many values), OneHotEncoder(Less values)

Feature Scaling ex.(StandardScaler, RobustScaler):
Apply normalization or standardization to ensure uniformity in feature scales.

Transformation Pipelines:
Create pipelines to streamline data cleaning, feature engineering, and scaling processes.

5. Select and Train Models

Start Simple:
Begin with basic models such as ex. Linear Regression to establish a baseline.

Progress to Complex Models:
Gradually shift to more intricate models like Decision Trees and Random Forests for potentially better performance.

Employ Cross-Validation:
Use cross-validation techniques for a more robust and reliable evaluation of model performance.

6. Fine-Tune Models

Hyperparameter Tuning:
Utilize tools like **GridSearchCV or RandomSearch** to optimize model hyperparameters.

Analyze Best Models:
Examine top-performing models, study their characteristics and errors for insights.

Test Set Evaluation:
Assess the final model's performance on a previously unseen test dataset to gauge real-world efficacy.

7. Launch, Monitor, and Maintain the System

Deployment:
Integrate the model into a production environment for real-world use.

Monitoring:
Set up systems to continuously assess the model's performance, ensuring it remains effective over time.

Regular Updates:
Periodically refresh the model with new data to maintain its relevance and accuracy.