Exploratory Data Analysis (EDA)

1. Data Cleaning

- Handle **missing values** (fill with mean/median for numerical, "Unknown" for categorical).
- Remove **duplicates** and standardize data types.

2. Univariate Analysis

- **Numerical:** Histograms, Boxplots (identify distributions & outliers).
- **Categorical:** Bar Charts, Pie Charts (analyze ad types, demographics, device usage).

3. Bivariate & Multivariate Analysis

- **Correlation Heatmap** – Identify relationships (e.g., Ad Spend vs. Revenue).
- **Scatter & Box Plots** – Analyze CPC, CTR, and ROI trends.
- **Pivot Tables** – Compare campaign performance across features.

4. Outlier Detection

- **Boxplots & Z-Score** – Spot anomalies in revenue, clicks, CPC.
- **Winsorization** – Cap extreme values to prevent skewed insights.

5. Time-Series Analysis

- **Trend & Seasonality Analysis** – Identify peak engagement times.
- **Rolling Averages** – Smoothen daily variations.

6. KPI Evaluation

- What features impact **Click-Through Rate (CTR)** the most?
- How does **Ad Spend correlate with Revenue & Conversions**?
- Are there **underperforming campaigns** with low engagement?
- Which **locations, devices, or demographics** perform best?
- Are there **seasonal trends** in ad performance?

Data Preprocessing

1. Handling Missing Values

- Fill **numerical** values using mean/median.
- Fill **categorical** values with mode or "Unknown".
- Drop columns if missing data >40%.

2. Standardization & Normalization

- **Standardization (Z-score):** For models sensitive to scale (e.g., regression, SVM).
- **Normalization (Min-Max Scaling):** For distance-based models (e.g., KNN, neural networks).

3. Encoding Categorical Variables

- **One-Hot Encoding:** For non-ordinal categories (device type, location).
- **Label Encoding:** For ordinal categories (ad ranking levels).

4. Handling Time-Series Data

- Convert timestamps to **datetime format**.
- Extract **hour, day, week, month, season** for trend analysis.
- Create **lag features & moving averages** for forecasting.

1. Feature Extraction (Deriving New Features)

We create additional features that provide better insights into **ad performance & user behavior**.

✅ **Engagement Metrics:**

- **CTR (Click-Through Rate)** = (Clicks / Impressions) × 100 → Measures ad effectiveness.
- **Bounce Rate** = (Users leaving without action / Total users) × 100 → Identifies poor-performing ads.
- **Avg. Session Duration** → Helps in understanding user retention.

✅ **Financial Performance Metrics:**

- **ROI (Return on Investment)** = (Revenue - Ad Cost) / Ad Cost → Evaluates profitability.
- **CPC (Cost per Click)** = Total Ad Spend / Clicks → Helps in budget optimization.
- **CAC (Customer Acquisition Cost)** = Total Ad Spend / Number of Conversions.

✅ **User Behavior & Demographics:**

- **Engagement Time (Peak Hours, Day of Week, Seasonality)** → Identifies the best times for ads.
- **Location-based Conversion Rate** → Helps in targeted marketing strategies.
- **Device-Based CTR** → Determines which devices perform better.

✅ **Campaign Effectiveness:**

- **Ad Fatigue Score** (CTR decay over time) → Detects if an ad is losing impact.
- **Repeat User Ratio** = Returning Users / Total Users → Helps in loyalty assessment.

---

2. Feature Selection (Choosing the Best Features)

To avoid **redundancy & improve model accuracy**, we select the most relevant features.

🔹 **Correlation Analysis:**

- Check relationships between features (remove highly correlated features).

🔹 **Variance Inflation Factor (VIF):**

- If **VIF > 5**, the feature may be redundant (like Total Ad Spend vs. CPC).

🔹 **Dimensionality Reduction (If Needed):**

- Apply **PCA (Principal Component Analysis)** if too many features cause overfitting.

1. Click-Through Rate (CTR) Prediction

Goal: Estimate how likely users are to click on an ad.
Best Model: XGBoost / LightGBM

🔹 Handles categorical + numerical data well.
🔹 Works great for structured ad campaign data.

---

2. Conversion Rate Prediction

Goal: Predict how many users will take the desired action (purchase, signup).
Best Model: Logistic Regression / Random Forest
🔹 Logistic Regression works well if data is **linear**.
🔹 Random Forest captures complex **non-linear relationships** (e.g., how age, location, device affect conversions).

---

3. Cost per Click (CPC) Optimization

Goal: Predict how much an advertiser will pay per click.
Best Model: Gradient Boosting (CatBoost, XGBoost)
🔹 Handles pricing data variations well.
🔹 Good for feature importance analysis (e.g., impact of ad type, audience demographics).

---

4. Return on Investment (ROI) Forecasting

Goal: Estimate campaign profitability.
Best Model: Linear Regression / ARIMA
🔹 **Linear Regression** is effective for short-term ROI analysis.
🔹 **ARIMA** works well for long-term **trend forecasting** (predicting future ROI based on past data).

---

5. Ad Spend Optimization

Goal: Predict the ideal budget allocation for max conversions.
Best Model: SARIMAX (for time-series) / Reinforcement Learning (RL-based budget allocation)
🔹 **SARIMAX** accounts for seasonality & ad performance over time.
🔹 **Reinforcement Learning (Multi-Armed Bandit)** dynamically adjusts budgets based on live performance.

---

6. Ad Fatigue Detection (When an ad loses effectiveness)

Goal: Identify when engagement drops over time.
Best Model: LSTM / Transformer Models

🔹 **LSTM (Long Short-Term Memory)** captures how CTR changes over time.
🔹 **Transformers** handle large-scale ad datasets (multi-campaign tracking).

1. Click-Through Rate (CTR) Prediction

🔹 **Metric: ROC-AUC Score** (how well the model distinguishes between clicked & non-clicked ads)
🔹 **Secondary Metrics:** Precision, Recall, F1-Score

---

2. Conversion Rate Prediction

🔹 **Metric: Precision & Recall** (especially important if conversions are rare)
🔹 **F1-Score** for balancing false positives & false negatives

---

3. Cost per Click (CPC) Optimization

🔹 **Metric: Mean Absolute Error (MAE) / Root Mean Squared Error (RMSE)** (to measure pricing accuracy)

---

4. Return on Investment (ROI) Forecasting

🔹 **Metric: Mean Squared Error (MSE)** (for predicting financial performance)
🔹 **R^2 Score** (to check how well the model explains variance in ROI)

---

5. Ad Spend Optimization

🔹 **Metric: Mean Absolute Percentage Error (MAPE)** (for predicting optimal spend)
🔹 **Hit Ratio** (percentage of times the model correctly predicts budget allocation)

---

6. Ad Fatigue Detection

🔹 **Metric: Time-to-Failure (TTF) Prediction Accuracy** (when engagement starts declining)
🔹 **F1-Score** (to detect early signs of ad fatigue)

Validation Techniques

✔ **Train-Test Split (80-20)** – General case
✔ **Time-Based Validation** – For time-dependent ad data
✔ **Cross-Validation (K-Fold / Time-Series Split)** – Ensures model stability

1. Model Serialization (Saving the Trained Model)

🔹 **Formats:**

- **Pickle (.pkl)** – For traditional ML models.
- **HDF5 (.h5)** – For deep learning models (TensorFlow/Keras).
- **ONNX (.onnx)** – For cross-framework compatibility.

🔹 **Storage Options:**

- **Local Storage** – For initial testing.
- **Cloud Storage (AWS S3, Google Cloud Storage, Azure Blob)** – For scalability.

---

2. Containerization (Ensuring Portability)

🔹 **Docker** – Packages the model with all dependencies.
🔹 **Docker Compose** – Manages multiple services like APIs & databases.
🔹 **Kubernetes** – For scalable deployment across multiple instances.

---

3. Model Deployment (Making It Accessible)

🔹 **API Deployment:**

- **FastAPI / Flask** – To expose the model via an API.
- **TorchServe / TensorFlow Serving** – For efficient deep learning model hosting.

🔹 **Cloud Deployment:**

- **AWS (Elastic Beanstalk, Lambda, SageMaker, EKS)**
- **GCP (Vertex AI, Cloud Run, Kubernetes Engine)**
- **Azure (ML Studio, AKS, Functions)**

🔹 **Edge Deployment:**

- **TensorFlow Lite / ONNX Runtime** – For mobile & IoT devices.

---

4. Monitoring & Logging

🔹 **Grafana + Prometheus** – To track model performance in production.
🔹 **MLflow** – For experiment tracking & model registry.
🔹 **Elastic Stack (ELK: Elasticsearch, Logstash, Kibana)** – For logging & visualization.

---

5. Version Control & Continuous Integration

🔹 **GitHub / GitLab** – For code & model versioning.
🔹 **DVC (Data Version Control)** – To track changes in datasets & models.
🔹 **CI/CD Pipelines (Jenkins, GitHub Actions, GitLab CI/CD)** – For automated deployment.