PACE Strategy (Plan, Analyze, Construct, Execute)

1. **PLAN**

- **Problem Definition**: Cloud cost overruns and anomalies create financial risk and operational inefficiencies.

- **Scope**:

    - Focus on usage and cost data across services (EC2, RDS, S3, Lambda, etc.).
    - Detect anomalies, optimize spend, and provide dashboards for decision-making.

- **Stakeholders**:

    - Executives (CFO, CTO): Require high-level KPIs, forecasts, and cost-saving estimates.
    - Engineering Managers: Require anomaly root-cause drilldowns.
    - Data Teams: Require reproducible analytics workflows and models.

- **Constraints**:

    - Dataset must simulate real-world messiness (missing values, duplicates, outliers).
    - Models must run efficiently on 1M+ rows in Jupyter/Python.

- **Success Metrics**:

    - ≥90% accuracy in anomaly detection.
    - Forecasting error ≤10% (MAPE).
    - Interactive dashboard usable by technical and non-technical audiences.

2. **ANALYZE**

- **Data Understanding**:

    - Fields: timestamp, resource_id, service, region, team, usage, cost, currency, owner, source_ip.
    - Issues: inconsistent service tags, missing team/cost values, duplicates, outliers.

- **Statistical Exploration**:

- Descriptive: total spend, spend by service/team, 95th percentile costs, utilization patterns.
- Hypothesis Testing: e.g., Do deployments correlate with spend spikes?
- Correlation: usage vs. cost, region vs. anomaly frequency.

- **Machine Learning Analysis**:

  - Isolation Forest / Local Outlier Factor → detect abnormal costs.
  - SARIMAX/Prophet → forecast monthly spend & detect deviations.

- **Insights Goal**:

  - Identify who, what, when, and why behind anomalies.
  - Quantify potential savings if anomalies were prevented.

3. **CONSTRUCT**

- **Data Pipeline (Python):**

  - Load → clean (timestamps, nulls, canonicalization, deduplication) → aggregate → save as Parquet.

- **Feature Engineering:**

  - Rolling averages, z-scores, cost deltas, anomaly flags.

- **Model Development:**

  - Train unsupervised anomaly detection models.
  - Build time-series forecasts for daily/weekly/monthly spend.

- **Visualization Prep:**

  - Aggregate outputs (service-level daily spend, anomaly labels, savings estimates).
  - Export to CSV/Parquet for Power BI integration.

- **Power BI Dashboard:**

  - Page 1: Executive KPIs & trends.
  - Page 2: Root-cause analysis drilldown.
  - Page 3: Model performance & what-if savings analysis.

4. EXECUTE

- **Implementation Steps**:
    - Generate synthetic dataset (cloud_usage_synthetic_1M.csv).
    - Run cleaning & EDA notebooks → produce cleaned dataset.
    - Apply anomaly detection & forecasting models.
    - Extract top anomalies and savings estimates.
    - Build Power BI dashboard using aggregated dataset.
    - Prepare README/report for recruiters.

- **Testing & Validation**:
    - Compare detected anomalies against injected spikes (ground truth in synthetic dataset).

- **Deployment/Presentation**:
    - Publish notebooks + CSV + Power BI .pbix file on GitHub/portfolio.
    - Record a 2–3-minute video walkthrough demonstrating interactive dashboard.

- **Impact Statement**:
    - “Using this pipeline, organizations could save up to 20–30% in cloud costs annually by preventing anomalies and optimizing underutilized resources.”