DATA ANALYTICS PROMPT LIBRARY

60 Production-Ready ChatGPT Prompts for Data Teams

A comprehensive library of carefully engineered prompts covering exploratory analysis, machine learning, business intelligence, customer analytics, and advanced data workflows. Fill in the [bracketed variables] and and use them with ChatGPT.

60 PROMPTS

10 CATEGORIES

32+ DOMAINS

How to Use This Guide

1. **Find** a prompt that matches your current analysis task or goal
2. **Fill in** all [bracketed placeholders] with your specific data context
3. **Paste** the completed prompt into ChatGPT, Claude, or any AI assistant
4. **Iterate** — refine the prompt or ask follow-up questions based on the output

Contents

01 · Exploratory & Data Foundation

7 PROMPTS

#01 EXPLORATION **Full Exploratory Data Analysis (EDA)**

> *Act as a senior data analyst. I have a dataset with the following columns:* [list your columns and data types] .
> *The dataset contains* [X rows] *and covers* [time period or scope] .

Perform a comprehensive EDA that includes:

1. A summary of the data (shape, types, missing values, duplicates)
2. Descriptive statistics for all numeric columns (mean, median, std, min, max, quartiles)
3. Distribution analysis — identify skewed or bimodal variables
4. Correlation matrix — highlight the top 5 strongest correlations with [target variable]
5. Outlier detection using IQR and z-score methods
6. Key patterns, anomalies, or surprising findings
7. A prioritized list of 3–5 next analytical steps you recommend

```
Use: Python (pandas, seaborn, matplotlib) — Provide clean, commented code and a plain-English summary after each section.
```

#02 DATA CLEANING

Intelligent Data Cleaning Pipeline

> *You are a data engineering expert. I have a messy dataset in* [CSV / Excel / SQL table] *format with these known quality issues:*

Missing values in columns: [list columns]

Suspected duplicates in: [column or row identifiers]

Inconsistent formats in: [e.g., date columns, categorical labels]

Possible outliers in: [numeric columns]

Build a full data cleaning pipeline in Python that:

1. Detects and reports missing values with percentage per column
2. Imputes missing values using the most appropriate strategy per column and explains why
3. Removes or flags exact and fuzzy duplicates
4. Standardizes date formats to YYYY-MM-DD
5. Normalizes categorical columns (lowercase, strip whitespace, map variants to canonical labels)
6. Caps or removes outliers based on IQR or z-score, keeping a change log
7. Exports the cleaned dataset and a cleaning summary report

Flag any decisions that require my judgment before applying them.

```
Use: Python — Full pipeline with change log and cleaning summary report.
```

#03 DATA QUALITY

Data Pipeline Audit and Quality Report

> *Act as a data quality engineer. I need to audit the data flowing through* [pipeline name or description] *and generate a quality report for stakeholders. The pipeline processes* [X records per day] *from* [source system] *to* [destination system / data warehouse] *. Known or suspected issues include:* [describe any known problems] *.*

Perform an audit that:

1. Profiles each key table or dataset (row counts, column completeness, cardinality)
2. Detects schema drift between source and destination
3. Validates business rules: [list 3–5 rules, e.g., "no negative revenue", "user_id must be unique"]
4. Measures freshness: time since last update per table
5. Identifies referential integrity violations across joined tables
6. Scores overall data quality on a 1–10 scale with breakdown by dimension
7. Generates a stakeholder-ready report with a priority-ranked issue list

```
Use: Python (Great Expectations or pandas) — Provide code and a formatted report template.
```

#04 DATA QUALITY

Data Audit for a New Dataset

> *Act as a data analyst onboarding a new dataset. I have just received a* [CSV / database table / JSON file] *from* [source: a vendor / internal team / survey platform] *that I have never worked with before. The dataset has* [X rows] *and* [X columns] *.*

Perform a thorough first-pass audit that:

1. Inventories all columns: name, data type, sample values, and suspected meaning
2. Calculates missing value rates and flags columns above [X%] missing
3. Identifies suspicious values: nulls coded as strings, dates out of expected range, negative values in non-negative fields
4. Detects likely duplicate records using fuzzy matching on [key identifier columns]
5. Validates value distributions against business expectations (e.g., "age should be between 18 and 90")
6. Assesses data freshness and coverage: what time period does this actually cover?
7. Produces a data dictionary draft and a data quality scorecard

```
Use: Python — Provide code and the data dictionary + scorecard output.
```

#05 DATA GOVERNANCE

Data Anonymization and Privacy Compliance Check

> *Act as a data privacy engineer. I have a dataset I need to share externally (with a vendor / for analytics / for research) that contains potentially sensitive information. Columns include:* [list all columns] .

Perform a privacy compliance review and anonymization that:

1. Classifies each column by sensitivity level: PII, quasi-identifier, sensitive attribute, non-sensitive
2. Identifies re-identification risk from combinations of quasi-identifiers (k-anonymity check)
3. Applies appropriate anonymization: masking, pseudonymization, generalization, or suppression per column
4. Verifies the anonymized dataset meets k-anonymity where k ≥ [3 / 5 / 10]
5. Tests that anonymized data preserves the statistical properties needed for the intended analysis
6. Documents all transformations for compliance records
7. Produces a privacy impact assessment summary suitable for a DPO review

```
Use: Python — Provide code and the compliance documentation template.
```

#06 DATA GOVERNANCE

Data Catalog and Metadata Management Setup

> *Act as a data governance engineer. My organization has* [X] *datasets across* [list of systems: data warehouse, data lake, operational databases, third-party feeds] *with no centralized catalog or metadata documentation.*

Build a data catalog foundation by:

1. Defining a metadata schema: dataset name, owner, source system, refresh frequency, row count, last updated, sensitivity classification, primary key, description
2. Writing a Python script to auto-inventory all tables in [database system] and populate the schema
3. Classifying all datasets by sensitivity: public, internal, confidential, restricted
4. Identifying datasets with no documented owner and flagging for assignment
5. Detecting datasets that haven't been updated in [X days] and marking as potentially stale
6. Building a searchable HTML or markdown catalog output from the inventory
7. Recommending a governance process: who reviews the catalog, how often, and what triggers an update

```
Use: Python — Provide code, a sample catalog output, and a governance process one-pager.
```

#07 ENGINEERING ANALYTICS

Exploratory Analysis of Unstructured Log Data

> *You are a data engineer and analyst. I have server or application log files in* [format: plain text, JSON, syslog] *covering* [time period] *. The logs contain fields like:* [timestamp, severity level, service name, user_id, error_code, message text] *.*

Parse and analyze the logs by:

1. Parsing raw log lines into a structured DataFrame
2. Counting error frequency by type, service, and time
3. Identifying recurring error patterns or error bursts using time bucketing
4. Correlating error spikes with deployment events or traffic surges
5. Extracting the most common error messages using clustering or frequency analysis
6. Identifying users or sessions most frequently encountering errors
7. Producing an operational health report: top 5 critical issues, their frequency, and recommended fixes

```
Use: Python (pandas, regex) — Provide log parsing code and the health report output.
```

02 · Statistics & Experimentation

7 PROMPTS

#08 STATISTICS

Hypothesis Testing with Interpretation

Act as a statistician. I want to test whether [specific hypothesis, e.g., "customers who receive email campaigns convert at a higher rate than those who don't"] .

Group A (control): [N = X, conversion rate or mean = Y]

Group B (treatment): [N = X, conversion rate or mean = Y]

Metric type: [binary / continuous / ordinal]

Do the following:

1. Recommend the correct statistical test (t-test, chi-square, Mann-Whitney U, ANOVA, etc.) and justify the choice
2. State the null and alternative hypotheses clearly
3. Check all test assumptions with code
4. Run the test and return: test statistic, p-value, confidence interval, and effect size
5. Interpret the result in plain business language
6. Warn me of any limitations or risks of the conclusion

```
Use: Python (scipy, statsmodels) — Include visualizations of the distributions.
```

#09 EXPERIMENTATION

A/B Test Results Analysis

Act as an experimentation analyst. I ran an A/B test with the following setup:

Experiment name: [name]

Control group (A): [N = X users, metric = Y]

Treatment group (B): [N = X users, metric = Y]

Primary metric: [conversion rate / revenue per user / click-through rate]

Secondary metrics: [list any]

Test duration: [X days]

Analyze the results by:

1. Checking for sample ratio mismatch (SRM)
2. Running the appropriate significance test with correction for multiple metrics
3. Calculating p-value, confidence interval, and minimum detectable effect
4. Checking for novelty effect or time-based degradation
5. Assessing practical significance — is the lift worth shipping?
6. Recommending a clear ship / no-ship / run longer decision with justification

```
Use: Python — Include code and a one-page summary suitable for a product review meeting.
```

#10 EXPERIMENTATION **Bayesian A/B Test Analysis**

> *Act as a Bayesian statistician. I have results from an A/B test:*

Control conversions: [X] out of [N]

Treatment conversions: [X] out of [N]

Perform a Bayesian analysis that:

1. Sets up a Beta-Binomial model with an appropriate prior (justify your choice)
2. Calculates the posterior distribution for each variant's conversion rate
3. Plots both posterior distributions on one chart with credible intervals
4. Computes the probability that B is better than A
5. Calculates the expected loss from choosing each variant
6. Determines if a decision can be made now or if more data is needed
7. Summarizes the result in plain language for a non-technical product manager

```
Use: Python (pymc or scipy) — Provide code, the posterior plot, and a decision recommendation.
```

#11 EXPERIMENTATION

Multivariate Testing Analysis

Act as an experimentation analyst. I ran a multivariate test on `[landing page / email / ad]` *with the following element variants:*

Element 1 (headline): `[A, B, C]`

Element 2 (CTA button): `[A, B]`

Element 3 (hero image): `[A, B]`

Primary metric: `[conversion rate]` — Total sessions: `[X]` split across `[N]` combinations

Analyze the results by:

1. Calculating conversion rate and statistical significance for each combination
2. Isolating the marginal effect of each element using factorial analysis
3. Identifying the winning combination and quantifying its improvement over control
4. Checking for interaction effects between elements
5. Correcting for multiple comparisons using Bonferroni or FDR correction
6. Estimating the revenue impact of deploying the winning combination
7. Recommending which elements to lock in and which need further testing

```
Use: Python — Provide code and a results summary table.
```

#12 EXPERIMENTATION

Pricing Elasticity Experiment Design

> *Act as a pricing experimentation expert. I want to run a controlled pricing experiment to measure the price elasticity of* `[product / service / subscription tier]` *.*

Design the experiment by:

1. Defining the price points to test: `[anchor price, treatment prices — at least 3 levels]`
2. Calculating the required sample size per price point to detect a `[X%]` change in conversion rate at 80% power and 95% confidence
3. Specifying the randomization unit: `[user / session / geography / account]` and justifying the choice
4. Identifying and controlling for confounders: day of week, device, acquisition channel, user tenure
5. Setting up a guardrail metric to detect any unintended harm (e.g., support ticket spike, refund rate increase)
6. Designing the holdout strategy and experiment duration
7. Producing a pre-analysis plan: primary metric, secondary metrics, statistical test, stopping rules, and decision criteria

```
Output: Complete experiment brief in structured format ready for engineering and legal review.
```

#13 STATISTICS

Causal Inference Analysis

Act as a causal inference specialist. I want to estimate the causal effect of [intervention: a policy change / feature launch / marketing campaign / price change] *on* [outcome metric] *, using observational data (no randomization was possible).*

Dataset: [X rows] , columns: [treatment indicator, outcome variable, pre-treatment covariates, time variable if panel data]

Perform a causal analysis using:

1. Propensity score matching or inverse probability weighting to balance treatment and control groups — report covariate balance before and after
2. Difference-in-differences if panel data is available — verify the parallel trends assumption
3. Instrumental variable (IV) estimation if a valid instrument exists: [describe candidate instrument]
4. A sensitivity analysis to assess how robust the estimate is to hidden confounding (Rosenbaum bounds)
5. Report the average treatment effect (ATE) and average treatment effect on the treated (ATT) with confidence intervals
6. Interpret the result in plain business language: what actually caused what?
7. Flag the key assumptions made and how likely they are to hold in this context

```
Use: Python (causalinference, econml, linearmodels) — Provide code and a causal evidence summary.
```

#14 STATISTICS

Regression Analysis for Business Drivers

> *Act as a quantitative analyst. I want to understand what drives* [outcome variable, e.g., customer lifetime value / monthly revenue / support ticket volume] *using regression analysis.*

Dataset: [X rows], columns: [list predictor variables and the outcome variable]

Run a full regression analysis that:

1. Checks assumptions: linearity, normality of residuals, homoscedasticity, multicollinearity (VIF)
2. Runs OLS regression and interprets each coefficient in plain business terms
3. Identifies and removes or handles outliers and high-leverage points
4. Tests for interaction effects between [specific variable pairs]
5. Uses stepwise or LASSO feature selection to find the minimal predictive model
6. Reports R-squared, adjusted R-squared, F-statistic, and AIC/BIC
7. Produces a coefficient plot with confidence intervals and a plain-English interpretation of each significant predictor

```
Use: Python (statsmodels, sklearn) — Provide code and a business-friendly results table.
```

03 · Machine Learning & AI

9 PROMPTS

#15 ML & PREDICTION

Predictive Model Build and Evaluation

> *Act as a machine learning engineer. I want to predict* [target variable, e.g., customer churn / revenue / equipment failure] .

Features: [list feature names and types] — Target: [name, type: binary / multiclass / continuous]
Size: [rows x columns] — Class imbalance (if classification): [e.g., 90% / 10%]

Build a complete ML pipeline that:

1. Splits data into train (70%), validation (15%), and test (15%) sets with stratification
2. Preprocesses features in-pipeline
3. Trains and compares at least 3 models
4. Tunes the best model using cross-validated grid search
5. Evaluates on the test set with all relevant metrics and plots
6. Explains top 10 features using SHAP values
7. Saves the final pipeline as a .pkl file

```
Use: Python (scikit-learn, SHAP) — Full pipeline with evaluation metrics and feature importance.
```

#16 ML & PREDICTION

Churn Prediction Model

> *Act as an ML engineer specializing in retention. I want to build a churn prediction model for* [SaaS / e-commerce / subscription] *customers.*

Rows: one per customer — Features: [list behavioral, usage, demographic, and billing features]
Label: [churned = 1, active = 0] — Imbalance: [X% churned] — Prediction horizon: [e.g., next 30 days]

Build a pipeline that:

1. Engineers relevant features (recency, frequency, tenure, usage trends)
2. Handles class imbalance using SMOTE or class weighting
3. Trains Logistic Regression, Random Forest, and XGBoost with cross-validation
4. Selects the threshold that maximizes F1 or recall based on business cost of false negatives
5. Outputs a scored customer list ranked by churn probability
6. Explains drivers for the top 10 highest-risk customers using SHAP force plots

```
Use: Python — Include code, evaluation metrics, and a decision memo for the retention team.
```

#17 ML & PREDICTION

Fraud Detection Model

> *Act as a fraud analytics specialist. I have transaction data with columns:* [transaction_id, user_id, amount, merchant_category, timestamp, device_type, location, is_fraud (label)] *. The fraud rate is approximately* [X%] *.*

Build a fraud detection system that:

1. Performs feature engineering: transaction velocity, amount deviation from user baseline, geographic anomalies, time-of-day patterns
2. Handles extreme class imbalance using SMOTE, class weighting, and anomaly detection approaches
3. Trains and compares: Logistic Regression, Random Forest, XGBoost, and Isolation Forest
4. Optimizes for recall (minimize false negatives / missed fraud) while keeping precision acceptable
5. Selects optimal classification threshold based on the cost matrix: [false negative cost = $X, false positive cost = $Y]
6. Explains flagged transactions using SHAP local explanations
7. Outputs a real-time scoring function returning fraud probability and risk tier

```
Use: Python — Provide evaluation metrics and a risk tier decision guide.
```

#18 ML & PREDICTION

Feature Engineering for Machine Learning

> *Act as a machine learning engineer. I have a raw dataset with columns:* [list all columns and data types] *that I need to prepare for a* [classification / regression / ranking] *model predicting* [target variable] *.*

Perform comprehensive feature engineering that:

1. Creates interaction features between [specific column pairs] and explains the business rationale
2. Encodes categorical variables using the most appropriate method per column (one-hot, target encoding, ordinal, binary) with justification
3. Applies date/time decomposition: extract day of week, hour, month, quarter, days since reference date, is_weekend flag
4. Engineers lag features and rolling window statistics (mean, std, min, max) for any time-dependent columns
5. Creates binned or bucketed versions of skewed numeric columns
6. Detects and removes or combines near-zero variance and perfectly correlated features
7. Produces a final feature importance pre-check using mutual information scores before any model is trained

```
Use: Python (pandas, scikit-learn, feature-engine) — Provide commented code and a feature catalog with business interpretation.
```

#19 ML OPERATIONS

Model Monitoring and Drift Detection

> *Act as an MLOps engineer. A predictive model was deployed* [X weeks / months] *ago. I now have* [X weeks] *of production prediction logs with columns:* [timestamp, input_features, predicted_label or score, actual_label if available] *.*

Set up model monitoring by:

1. Detecting feature drift: compare production feature distributions vs. training distributions using KS test and PSI
2. Detecting label drift: compare production prediction score distributions over time
3. Monitoring model accuracy: calculate performance metrics on any labeled production data available
4. Identifying underperforming data slices: are there specific segments where accuracy has degraded?
5. Setting up statistical process control charts for key metrics with alert thresholds
6. Producing a model health dashboard with weekly rollup
7. Recommending a retraining trigger policy based on drift severity observed

```
Use: Python (evidently, scipy) — Provide code and the monitoring dashboard spec.
```

#20 ML & PREDICTION

Benchmarking Model Performance Across Data Slices

> *Act as a responsible AI and ML evaluation specialist. I have a trained* [classification / regression] *model and a labeled test set with columns:* [features, target, and demographic or segment fields like region, age_group, device_type, customer_tier] .

Evaluate model fairness and slice performance by:

1. Calculating overall model performance: accuracy, F1, AUC, RMSE as appropriate
2. Breaking down performance by each segment dimension: compute the same metrics per slice
3. Identifying slices where performance falls below [X%] of overall performance — flag as underperforming
4. Testing whether performance gaps between slices are statistically significant
5. Checking for disparate impact: does the model systematically over-predict or under-predict for any group?
6. Tracing underperforming slices back to training data: are they underrepresented or have noisier labels?
7. Recommending remediation: targeted data collection, slice-specific retraining, or post-processing calibration

```
Use: Python (sklearn, slicefinder or pandas groupby) — Provide code, a slice performance heatmap, and a model fairness summary memo.
```

#21 NLP & TEXT ANALYTICS

Text Classification Model for Support Tickets

> *You are an NLP engineer. I have* [X] *customer support tickets with columns:* [ticket_id, ticket_text, resolution_time_hours, agent_id, and optionally a category label for a subset] *. I want to automatically classify incoming tickets into categories:* [list your categories, e.g., billing, technical issue, account access, feature request, complaint] .

Build a text classification pipeline that:

1. Cleans and preprocesses ticket text: lowercasing, punctuation removal, stopword filtering, lemmatization
2. Converts text to features using TF-IDF and separately using sentence embeddings (sentence-transformers)
3. Trains classifiers: Logistic Regression, SVM, and a fine-tuned DistilBERT — compare all three
4. Evaluates using macro F1 and per-class precision/recall — flag any category with F1 below [X%]
5. Handles class imbalance if any category has fewer than [X] training examples
6. Builds a confidence-based routing rule: high-confidence predictions auto-route, low-confidence flag for human review
7. Saves the production pipeline and outputs predictions for all unlabeled tickets with confidence scores

```
Use: Python (scikit-learn, transformers, sentence-transformers) — Provide code and a model performance report.
```

#22 OPERATIONS ANALYTICS

Predictive Maintenance Analysis

> *Act as a reliability engineer and data scientist. I have sensor or operational data from* [X machines / assets] *with columns:* [asset_id, timestamp, sensor readings (temperature, vibration, pressure, etc.), maintenance_logs, failure_flag] .

Build a predictive maintenance model that:

1. Engineers time-windowed features: rolling mean, rolling std, rate of change per sensor over [X hours / days]
2. Labels failure precursor windows: flag data [X hours] before each failure event
3. Trains a classifier to predict failure probability in the next [X hours]
4. Evaluates using precision, recall, and lead time — how early does the model detect failure?
5. Identifies which sensors are the strongest predictors using SHAP
6. Builds a maintenance scheduling rule: trigger alert when probability exceeds [X%]
7. Estimates cost savings from predicted vs. reactive maintenance based on [downtime cost per hour]

```
Use: Python (scikit-learn, tsfresh) — Provide code, a confusion matrix, and a cost-benefit summary.
```

#23 ADVANCED ANALYTICS

Dimensionality Reduction and Visualization

> *Act as a data scientist. I have a high-dimensional dataset with* [X features] *and* [X rows] *representing* [customers / products / documents / sensor readings] *. The data is difficult to interpret or visualize in its raw form.*

Apply dimensionality reduction by:

1. Preprocessing all features: scale numerics, encode categoricals, handle missing values
2. Applying PCA and retaining components that explain at least [80% / 90%] of variance — plot the scree plot and cumulative explained variance
3. Applying t-SNE with perplexity values of [5, 30, 50] and comparing the resulting cluster structures
4. Applying UMAP with [min_dist = 0.1, n_neighbors = 15] as an additional comparison
5. Coloring all 2D projections by [a known label or segment field] to assess whether structure aligns with business categories
6. Identifying any clear clusters or outlier groups visible in the reduced space and describing their characteristics in original feature terms
7. Recommending which reduction method best preserves the structure relevant to [your downstream task] and explaining why

```
Use: Python (scikit-learn, umap-learn, plotly) — Provide code and a side-by-side comparison of all three projections.
```

04 · Forecasting & Time Series

2 PROMPTS

#24 FORECASTING **Time Series Forecasting**

> *Act as a time series analyst. I have* [X months / years] *of* [metric, e.g., weekly sales, daily active users] *data with columns:* [date column, value column, any other relevant columns] *.*

Build a forecasting pipeline that:

1. Decomposes the series into trend, seasonality, and residual components
2. Tests for stationarity (ADF test) and applies differencing if needed
3. Identifies ARIMA parameters using ACF and PACF plots
4. Trains and compares: ARIMA, SARIMA, and Prophet
5. Evaluates models using RMSE, MAE, and MAPE on a held-out test set
6. Generates a [X weeks / months] forecast with 80% and 95% confidence intervals
7. Plots actuals vs. forecast with confidence bands

```
Use: Python — Provide code and flag any seasonality patterns or anomalies found.
```

#25 FINANCE ANALYTICS **Financial Forecasting and Variance Analysis**

> *Act as a financial analyst. I have* [monthly / quarterly] *actuals vs. budget data for* [business unit / product / cost center] *covering* [time period] *. Columns include:* [revenue, COGS, gross margin, opex line items, net income, and period] *.*

Perform a full variance analysis that:

1. Calculates actual vs. budget variance in absolute and percentage terms for each line item
2. Identifies the top 3 drivers of favorable and unfavorable variance
3. Builds a bridge (waterfall) chart showing how each variance contributed to the net income gap
4. Runs a rolling 12-month forecast using actuals trend extrapolation
5. Flags any line items where variance exceeds [X%] threshold for escalation
6. Performs scenario analysis: best case, base case, worst case for next [X quarters]
7. Produces a CFO-ready summary slide narrative with headline numbers and commentary

```
Use: Python or Excel — Provide code or formulas and the narrative template.
```

05 · Customer & Segmentation Analytics

7 PROMPTS

#26 SEGMENTATION

Customer Segmentation Analysis

> *You are a customer analytics expert. I have a dataset of* [X customers] *with the following behavioral and demographic features:* [list features] *.*

Perform a full customer segmentation that:

1. Selects the most relevant features and explains the selection rationale
2. Scales and preprocesses the data appropriately
3. Determines the optimal number of clusters using the elbow method and silhouette score
4. Runs K-Means and DBSCAN and compares results
5. Profiles each segment: size, key characteristics, average metrics
6. Visualizes clusters using PCA or UMAP for 2D representation
7. Recommends a business action for each segment (e.g., upsell, retain, reactivate)

```
Use: Python (scikit-learn, plotly) — Provide both code and a plain-English segment summary.
```

#27 PRODUCT ANALYTICS

Cohort Retention Analysis

> *Act as a product analyst. I have event-level data with columns:* [user_id, event_date, event_type, and any other relevant fields] *. I want to measure user retention by cohort.*

Build a cohort analysis that:

1. Defines cohorts by [first purchase date / signup month / first login]
2. Calculates monthly retention rates for each cohort up to [X months]
3. Generates a cohort retention heatmap (cohorts as rows, months as columns)
4. Identifies which cohorts have the strongest and weakest retention
5. Calculates average Day 1, Day 7, and Day 30 retention across all cohorts
6. Flags any cohort that drops below [X%] retention threshold and suggests possible causes
7. Recommends 2–3 product or lifecycle actions based on the retention pattern

```
Use: Python (pandas, seaborn) — Output the heatmap and a written commentary.
```

#28 PRODUCT ANALYTICS

Funnel Drop-off Analysis

> *You are a conversion optimization analyst. I have funnel data tracking users across* [X steps, e.g., Landing Page > Sign Up > Onboarding > First Purchase] *. The dataset has columns:* [user_id, step_name, timestamp, any segment fields] *.*

Analyze funnel drop-off by:

1. Calculating conversion rate at each step overall
2. Breaking down conversion by [segment: device / channel / region / user type]
3. Identifying the single biggest drop-off point and quantifying the revenue or user impact
4. Detecting if drop-off has worsened or improved over [time period]
5. Running statistical significance tests on segment-level differences
6. Building a Sankey diagram or funnel chart visualization
7. Recommending 3 specific, prioritized experiments to improve the worst-performing step

```
Use: Python or SQL — Provide code and a prioritized action plan.
```

#29 CUSTOMER ANALYTICS

Lifetime Value (LTV) Modeling

> *Act as a customer analytics expert. I need to calculate and model customer lifetime value (LTV) for* [business type, e.g., SaaS, e-commerce, subscription box] *.*

Dataset columns: [customer_id, acquisition_date, orders or payments with amounts, churn_date if applicable, acquisition_channel, plan or product type]

Build a full LTV model that:

1. Calculates historical LTV for each customer (sum of gross margin over lifetime)
2. Segments customers into LTV tiers: top 10%, mid 40%, bottom 50%
3. Builds a predictive LTV model using BG/NBD + Gamma-Gamma (for non-contractual) or regression (for contractual)
4. Compares predicted LTV by acquisition channel and cohort
5. Calculates payback period per channel: LTV vs. CAC
6. Identifies the customer attributes most correlated with high LTV
7. Recommends acquisition and retention strategies for each LTV segment

```
Use: Python (lifetimes library or custom) — Provide code, a LTV distribution chart, and a strategic summary.
```

#30 CUSTOMER ANALYTICS

Customer Journey Mapping with Data

> *Act as a customer experience analyst. I have event-level behavioral data tracking customers across multiple touchpoints with columns:* [customer_id, event_type, channel, timestamp, session_id, device_type, and any outcome flags like purchase or churn] .

Map and analyze the customer journey by:

1. Reconstructing individual customer paths by sequencing events chronologically per customer
2. Identifying the top 10 most common journey sequences from first touch to conversion or churn
3. Calculating average time between each stage transition and flagging stages with the longest delays
4. Building a Sankey diagram visualizing the flow of customers across all major touchpoints
5. Segmenting journeys by outcome: compare paths of converted vs. churned customers to identify divergence points
6. Detecting the single touchpoint whose presence most strongly correlates with a positive outcome using lift analysis
7. Recommending 3 journey optimization interventions with estimated impact on conversion rate or time-to-value

```
Use: Python (pandas, plotly) — Provide code, the Sankey diagram, and a journey insight summary.
```

#31 CUSTOMER ANALYTICS

Demographic Analysis and Segmentation

> *You are a market research analyst. I have demographic survey data with columns:* [age, gender, income_bracket, education_level, location, product_usage_frequency, satisfaction_score, NPS, and others] .

Perform a demographic analysis that:

1. Profiles the overall respondent base (distributions for each demographic variable)
2. Cross-tabulates satisfaction and NPS by each demographic segment
3. Identifies which demographic groups have the highest and lowest satisfaction
4. Tests whether demographic differences in satisfaction are statistically significant
5. Builds a persona for the top 3 most engaged demographic segments
6. Maps demographic segments to product or marketing recommendations
7. Identifies any underrepresented demographic segments and flags potential survey bias

```
Use: Python — Provide code, a segment summary table, and 3 persona write-ups.
```

#32 SEGMENTATION

Cluster Profiling and Business Interpretation

> *Act as a customer analytics expert. I have already run K-Means clustering and produced* [X clusters] *on a dataset with columns:* [list features used] *. Each customer now has a cluster label assigned.*

Profile and interpret the clusters by:

1. Calculating the mean and median of each feature by cluster
2. Identifying the top 3 distinguishing features per cluster using ANOVA F-test or feature importance
3. Writing a plain-English persona for each cluster: name, key traits, likely behaviors, and needs
4. Visualizing clusters using a radar chart overlaid for all segments
5. Sizing each cluster: count, percentage of total, and share of [revenue / orders / usage]
6. Mapping each cluster to the most appropriate product, pricing, or marketing strategy
7. Recommending a measurement plan to track how cluster membership changes over time

```
Use: Python — Provide code and a cluster profile card for each segment.
```

06 · Business & Revenue Analytics

8 PROMPTS

#33 BUSINESS ANALYTICS

Revenue Attribution Analysis

> *Act as a revenue analyst. I need to attribute* [total revenue / MRR / ARR] *across* [channels / products / regions / sales reps] *for* [time period] *. Dataset columns:* [list relevant fields] *.*

Perform attribution analysis that:

1. Calculates total and per-unit revenue by each dimension
2. Identifies the top 20% of drivers contributing 80% of revenue (Pareto analysis)
3. Computes month-over-month and year-over-year growth rates per segment
4. Detects any underperforming segments relative to target or prior period
5. Builds a contribution waterfall chart showing what drove overall revenue change
6. Flags anomalies (sudden spikes or drops) with a possible explanation
7. Provides a one-paragraph executive summary suitable for a board update

```
Use: Python or SQL — Output charts and a narrative summary.
```

#34 BUSINESS ANALYTICS

Multi-dimensional KPI Decomposition

> *Act as a strategic analyst. A key business KPI —* [e.g., revenue per user / gross margin / CAC payback period] *— has changed by* [X%] *between* [period A] *and* [period B]*. I need to fully decompose what drove this change.*

Decompose the KPI by:

1. Breaking it into its mathematical components (e.g., revenue per user = sessions × conversion rate × AOV)
2. Calculating the contribution of each component to the total change using a multiplicative or additive decomposition
3. Further breaking down each component by [dimension: region / channel / product / segment]
4. Identifying which single component and sub-dimension explains the largest share of the change
5. Checking whether the change is broad-based or concentrated in a small subset of records
6. Running a counterfactual: what would the KPI be if only [one component] had changed and everything else stayed flat?
7. Producing a waterfall chart of contributions and a 3-sentence executive summary of the finding

```
Use: Python — Provide code, the waterfall chart, and the executive summary.
```

#35 BUSINESS ANALYTICS

Competitive Benchmarking with Data

> *Act as a competitive intelligence analyst. I have data on* [X competitors] *across metrics like:* [revenue growth, market share, pricing, NPS, product features, employee count, Glassdoor rating, ad spend, etc.]*. Sources include:* [public reports, web scrapes, industry databases, survey data]*.*

Build a competitive benchmarking analysis that:

1. Normalizes all metrics to a common scale for fair comparison
2. Creates a competitive positioning matrix (2×2 or radar chart) using [X vs. Y dimensions]
3. Identifies areas where our company leads, is at parity, or lags behind
4. Scores each competitor on a weighted KPI scorecard
5. Highlights metrics where competitors have improved fastest over the past [X quarters]
6. Identifies one "white space" opportunity no competitor is currently winning
7. Produces a one-page competitive summary with a strategic priority list

```
Use: Python or Excel — Provide code and the formatted output.
```

#36 REVENUE ANALYTICS

Cross-sell and Upsell Opportunity Analysis

> *Act as a revenue analytics specialist. I have purchase history data with columns:* [customer_id, product_id, product_category, purchase_date, revenue, customer_segment, account_manager] .

Identify cross-sell and upsell opportunities by:

1. Calculating product penetration rate by customer segment
2. Identifying which products are most commonly purchased together (use association rules)
3. Finding customers who buy Product A but not Product B, where B has high affinity with A
4. Scoring each customer-product pair for upsell potential using purchase recency, frequency, and fit
5. Ranking the top 50 accounts by estimated expansion revenue opportunity
6. Mapping each opportunity to a recommended outreach action and suggested offer
7. Estimating total addressable expansion revenue if top 20% of opportunities are converted

```
Use: Python — Provide code and a prioritized account expansion list ready for the sales team.
```

#37 PRICING & REVENUE ANALYTICS

Pricing Tier and Packaging Analysis

> *Act as a pricing strategist. I have data on* [X customers] *across* [X pricing tiers or product packages] *including:* [customer_id, tier, mrr or arr, usage metrics, feature adoption flags, support ticket count, nps, churn_flag] .

Analyze pricing tier performance by:

1. Calculating revenue, churn rate, NPS, and support burden by tier
2. Identifying which tier has the highest LTV and lowest churn
3. Detecting customers who are under-tiered (high usage relative to their plan)
4. Detecting customers who are over-tiered (low usage relative to their plan)
5. Analyzing feature adoption: which features differentiate power users from low-usage customers?
6. Running a willingness-to-pay analysis if survey or pricing experiment data is available
7. Recommending a pricing tier restructuring with projected revenue impact

```
Use: Python — Provide code and a tier performance scorecard.
```

#38 PRICING ANALYTICS

Price Sensitivity and Elasticity Analysis

> *Act as a pricing analyst. I have historical data on* [product / service] *sales including:* [price, quantity sold, date, segment or region, and any promotion flags] .

Analyze price sensitivity and elasticity by:

1. Plotting demand curve (price vs. quantity) for each key segment
2. Calculating price elasticity of demand (PED) with confidence intervals
3. Identifying the price point that maximizes revenue vs. volume
4. Segmenting customers by price sensitivity (elastic vs. inelastic buyers)
5. Testing whether promotions have a lasting or temporary demand effect
6. Building a simple price optimization model using regression
7. Recommending 2–3 pricing strategy changes with projected impact

```
Use: Python (statsmodels, scipy) — Provide code, charts, and a pricing recommendation memo.
```

#39 RETAIL & E-COMMERCE ANALYTICS

Market Basket Analysis

> *You are a retail data analyst. I have transaction data with columns:* [transaction_id, customer_id, product_id, product_name, quantity, date] *. I want to find product associations to improve cross-sell and bundling strategies.*

Perform market basket analysis that:

1. Transforms data into a basket format (one row per transaction, columns per product)
2. Runs the Apriori algorithm with minimum support: [X%] and confidence: [X%]
3. Filters and ranks association rules by lift, confidence, and support
4. Identifies the top 10 product pairs or triplets with the strongest lift
5. Visualizes the association network as a graph
6. Translates findings into 3 specific bundling or upsell recommendations
7. Estimates potential revenue uplift if the top recommendation is implemented

```
Use: Python (mlxtend) — Include code and a business-ready summary table.
```

#40 SALES ANALYTICS

Sales Pipeline and Forecast Analysis

> *Act as a sales operations analyst. I have CRM pipeline data with columns:* [deal_id, stage, amount, close_date, owner, product_line, lead_source, created_date, days_in_stage, probability] .

Analyze the pipeline by:

1. Calculating pipeline coverage ratio: total pipeline value vs. quota for [quarter]
2. Identifying deals at risk: past expected close date, stuck in stage for > [X days], or low engagement signals
3. Building a weighted forecast: sum of (amount × probability) vs. a regression-based forecast
4. Analyzing win rate and average deal size by stage, rep, product line, and lead source
5. Detecting pipeline leakage: deals that moved backward in stage or were suddenly lost
6. Forecasting end-of-quarter attainment under three scenarios (commit / upside / downside)
7. Recommending 3 pipeline actions for the sales manager to take this week

```
Use: Python or SQL — Provide code and a deal risk heat map.
```

07 · Marketing & Digital Analytics

4 PROMPTS

#41 MARKETING ANALYTICS

Data-driven Marketing Attribution

> *Act as a marketing data analyst. I need to attribute* [revenue / conversions / sign-ups] *across the following channels:* [email, paid search, organic, social, direct, referral] *. I have user-level touchpoint data with columns:* [user_id, channel, touchpoint_date, conversion_flag, conversion_value] .

Build a multi-touch attribution analysis that:

1. Compares four models: first touch, last touch, linear, and time decay
2. Calculates attributed revenue and conversion count per channel under each model
3. Visualizes how attribution share shifts across models for each channel
4. Identifies which channels are undervalued or overvalued by last-touch attribution
5. Recommends a data-driven attribution model using Shapley values
6. Estimates the incremental ROAS (return on ad spend) per channel
7. Produces a channel investment reallocation recommendation based on findings

```
Use: Python — Provide code and a channel performance scorecard.
```

#42 MARKETING ANALYTICS

Social Media Analytics and Performance Reporting

> *You are a social media analytics expert. I have platform data from* [LinkedIn / Instagram / Twitter / TikTok / YouTube] *with columns:* [post_id, date, format, impressions, reach, engagements, clicks, conversions, follower_count, post_copy] .

Build a performance analysis that:

1. Calculates engagement rate, CTR, and CPE (cost per engagement if paid data available) per post
2. Identifies the top 10 and bottom 10 performing posts by [primary KPI]
3. Analyzes performance by content format (video vs. image vs. carousel vs. text)
4. Tracks follower growth rate and correlates spikes with specific posts or campaigns
5. Detects the best posting day and time for [reach / engagement / conversions]
6. Compares [current period] vs. [prior period] performance across all metrics
7. Recommends a content calendar strategy for next [month / quarter] based on findings

```
Use: Python — Provide code and a monthly performance report template.
```

#43 DIGITAL ANALYTICS

Web Analytics and Conversion Optimization

> *Act as a web analytics expert. I have Google Analytics or similar data covering* [time period] *for* [website / app] . *Available dimensions include:* [sessions, users, bounce rate, pages per session, goal completions, revenue, traffic source, device type, landing page, and others] .

Perform a full web analytics audit that:

1. Identifies top traffic sources by volume and by conversion rate
2. Finds the highest-traffic pages with the lowest conversion rates (biggest opportunity pages)
3. Analyzes mobile vs. desktop performance gap across all key metrics
4. Builds a session quality score using engagement proxies (time on page, pages per session, scroll depth)
5. Detects pages with unusually high exit rates and flags them for UX review
6. Identifies the content or landing pages most correlated with eventual conversion
7. Recommends a prioritized CRO experiment roadmap with estimated impact

```
Use: Python (or GA4 API) — Provide code and a CRO priority matrix.
```

#44 BUSINESS ANALYTICS

Geographic Market Expansion Analysis

> *Act as a market expansion analyst. My company currently operates in* [current markets] *. I am evaluating expanding into* [list of candidate markets / regions] *. I have the following data available:* [market size, population, competitor presence, regulatory complexity score, logistics cost index, historical performance in similar markets] .

Build an expansion prioritization analysis that:

1. Defines and weights scoring criteria: [market size, competition, cost to serve, strategic fit, regulatory risk]
2. Normalizes all criteria to a 0–10 scale and calculates a weighted opportunity score per market
3. Builds a 2×2 prioritization matrix: opportunity score vs. ease of entry
4. Identifies the top 3 markets to enter first and the rationale for each
5. Estimates Year 1 revenue potential for the top market using bottom-up assumptions
6. Flags the biggest risk for each top market and proposes a mitigation
7. Produces a one-page market entry scorecard ready for executive review

```
Use: Python or Excel — Provide the scoring model, the 2x2 chart, and the scorecard.
```

08 · Operations Analytics

4 PROMPTS

#45 OPERATIONS ANALYTICS

Supply Chain Analytics and Demand Forecasting

> *You are a supply chain analyst. I have* [X months] *of historical demand data with columns:* [product_id, product_name, date, units_sold, units_returned, inventory_level, lead_time_days, stockout_flag] .

Perform a supply chain analysis that:

1. Calculates demand variability (CV) per product to classify into fast / slow / erratic movers
2. Builds a demand forecast for each product for the next [X weeks / months] using appropriate model (moving average, exponential smoothing, or ML-based)
3. Calculates safety stock levels using service level target of [X%]
4. Identifies products at risk of stockout in the next [X days] given current inventory
5. Flags slow-moving or dead stock exceeding [X days] of inventory cover
6. Recommends reorder points and order quantities per SKU
7. Estimates the working capital impact of the recommended inventory policy vs. current policy

```
Use: Python — Provide code and a prioritized reorder action list.
```

#46 OPERATIONS ANALYTICS

Inventory Optimization Analysis

> *You are an operations research analyst. I have inventory and sales data with columns:* [product_id, sku, current_stock, average_daily_demand, demand_std_dev, lead_time_days, holding_cost_per_unit, stockout_cost_per_unit, order_cost] .

Optimize my inventory policy by:

1. Calculating EOQ (Economic Order Quantity) for each SKU
2. Computing safety stock at [90% / 95% / 99%] service levels
3. Setting reorder point (ROP) for each SKU
4. Identifying SKUs where current stock is critically below ROP
5. Flagging overstock items (current stock > [X days] of demand cover) and calculating holding cost impact
6. Performing ABC-XYZ classification: value × demand variability
7. Recommending differentiated replenishment policies by ABC-XYZ class with estimated cost savings vs. current policy

```
Use: Python — Provide code and a prioritized reorder action table.
```

#47 HR ANALYTICS

Employee Attrition Analysis

> *You are an HR analytics expert. I have employee data with columns:* [employee_id, department, role, tenure, salary, performance_rating, attrition (yes/no), engagement_score, manager_id, and others] .

Analyze attrition by:

1. Calculating overall and department-level attrition rates
2. Identifying which roles, tenure bands, and pay grades have the highest attrition
3. Running a logistic regression to find the top 5 drivers of attrition
4. Building a survival analysis to estimate expected tenure by employee segment
5. Flagging high-risk current employees based on the model (top 10%)
6. Comparing attrition drivers vs. industry benchmarks where possible
7. Recommending 3 targeted retention interventions with estimated cost vs. impact

```
Use: Python — Provide code, charts, and a summary report for the CHRO.
```

#48 GEOSPATIAL ANALYTICS

Geospatial Data Analysis

> *Act as a geospatial data analyst. I have a dataset with* [X records] *containing location data (latitude, longitude or address) along with:* [other relevant fields like sales, incidents, customers, assets] .

Perform geospatial analysis that:

1. Geocodes addresses to coordinates if needed
2. Plots all records on an interactive map with color-coded markers by [metric or category]
3. Performs clustering to identify geographic hotspots using DBSCAN or KMeans
4. Calculates distance-based metrics (e.g., nearest store, service area coverage)
5. Builds a choropleth map by [region / zip code / city] for [key metric]
6. Identifies underserved or overserved geographic areas
7. Recommends 2–3 location-based strategic actions based on the findings

```
Use: Python (geopandas, folium, plotly) — Provide code and an interactive HTML map output.
```

09 · NLP, Text & Advanced Analytics

4 PROMPTS

#49 NLP & TEXT ANALYTICS

Natural Language Processing on Survey Data

> *You are an NLP analyst. I have* [X responses] *from a customer or employee survey. The open-ended question was: '* [paste question text] *". Responses are in column* [column name] .

Analyze the text data by:

1. Cleaning the text (lowercasing, stopword removal, lemmatization)
2. Running sentiment analysis on each response (positive / negative / neutral) with confidence score
3. Extracting the top 20 keywords and bigrams using TF-IDF
4. Identifying 5–8 themes using topic modeling (LDA or BERTopic)
5. Grouping responses by sentiment × theme matrix
6. Flagging the most critical negative responses for immediate review
7. Generating a one-page executive summary of what respondents care most about

```
Use: Python (NLTK, transformers, gensim) — Provide code and the summary output.
```

#50 NLP & TEXT ANALYTICS

Sentiment Analysis on Product Reviews

> *You are an NLP specialist. I have* [X product reviews] *for* [product name] *from* [platform, e.g., Amazon, G2, App Store] *. The dataset has columns:* [review_text, rating, date, verified_purchase, and any other metadata] .

Analyze reviews by:

1. Classifying sentiment at review level (positive / negative / neutral)
2. Extracting aspect-level sentiment: identify how customers feel about [price, quality, delivery, support, UX, etc.] separately
3. Tracking sentiment trend over [time period] — is sentiment improving or declining?
4. Identifying the top 10 most common complaints and the top 10 most praised features
5. Flagging reviews with high impact (verified + long + negative) for product team review
6. Comparing sentiment distribution across rating levels (do 3-star reviews skew more negative on which aspects?)
7. Generating a product feedback summary memo with prioritized action items for the product team

```
Use: Python (transformers / VADER / spaCy) — Provide code and the final memo.
```

#51 ADVANCED ANALYTICS

Network Analysis for Business Relationships

> *Act as a network analyst. I have relational data representing* [business relationships: customer referrals / supplier connections / employee collaboration / transaction networks] *. The dataset has:* [node_1, node_2, relationship_type, weight or frequency, date] .

Perform a network analysis that:

1. Builds a directed or undirected graph from the data
2. Calculates centrality measures: degree, betweenness, closeness, and PageRank for each node
3. Identifies the top 10 most influential nodes and explains what their centrality means in business terms
4. Detects communities or clusters within the network
5. Finds any critical bridge nodes whose removal would fragment the network
6. Visualizes the network with node size proportional to influence and color by community
7. Recommends 2–3 strategic actions based on the network structure (e.g., key account prioritization, risk nodes, partnership opportunities)

```
Use: Python (networkx, plotly) — Provide code and a strategic summary.
```

#52 ANOMALY DETECTION

Anomaly Detection in Operational Data

> *You are a data scientist specializing in anomaly detection. I have* [time series / transactional / sensor] *data with columns:* [list fields] *. I need to identify unusual patterns that may indicate* [fraud / equipment failure / data pipeline errors / sudden business changes] *.*

Build a detection system that:

1. Applies statistical control charts (mean ± 2 or 3 sigma) as a baseline
2. Implements Isolation Forest and LOF for multivariate anomaly detection
3. Flags anomalies with a severity score (low / medium / high) based on deviation magnitude
4. Plots anomalies in context on a time series chart with highlighted points
5. Groups anomalies by type or likely root cause where possible
6. Recommends an alerting threshold strategy for production use
7. Outputs an anomaly log with timestamp, affected column, deviation value, and severity

```
Use: Python (scikit-learn, pyod) — Provide code and a summary of the most critical anomalies found.
```

10 · Visualization, Reporting & Workflow

8 PROMPTS

#53 VISUALIZATION

Strategic Data Visualization Plan

> *You are a data visualization expert. I need to present findings from a* [type of analysis] *to* [audience: executive team / technical stakeholders / clients] *. My dataset has:* [describe key columns and metrics] *. My main message is:* [the single most important finding] *.*

For each of 5 chart types, tell me: what question it answers; the exact chart type and why; which columns go on which axis; what color encoding or annotation would add insight; and common mistakes to avoid.

Then provide Python code (matplotlib or plotly) for the 2 most impactful charts, with:

- Titles, axis labels, and a one-sentence insight annotation on the chart
- Audience-appropriate color palette and typography settings
- Export at print-quality resolution

```
Use: Python (matplotlib or plotly) — Annotated, publication-ready charts with insight callouts.
```

#54 REPORTING & BI

Dashboard Design and KPI Framework

You are a BI analyst and dashboard designer. I need to build an executive dashboard for [business function: marketing / sales / operations / finance] *tracking performance for* [audience: C-suite / department head / team lead] *. The business goal is:* [describe the decision the dashboard should support] *.*

Available data sources: [list systems, e.g., Salesforce, Google Analytics, ERP]

Key metrics I track today: [list current metrics]

Design the dashboard by:

1. Recommending a KPI hierarchy (north star metric, secondary metrics, diagnostic metrics)
2. Mapping each metric to a specific business question it answers
3. Specifying the best visualization type for each metric and why
4. Designing the layout: which metrics appear on top, how to group related KPIs
5. Defining alert thresholds for each KPI
6. Recommending refresh frequency (real-time / daily / weekly) per metric
7. Providing a mockup description or wireframe in text format ready for Tableau, Power BI, or Looker

```
Output: Include a one-paragraph rationale for the overall design decisions.
```

#55 COMMUNICATION & REPORTING

Data Storytelling and Presentation Builder

Act as a data storytelling expert. I have the following analysis results to present: [paste your key findings, charts, or tables] *. My audience is:* [e.g., CEO, VP of Marketing, board of directors] *. Their key question is:* [e.g., "Should we expand into the EMEA market?"] *. Decision to be made:* [describe the decision] *.*

Build a data-driven narrative that:

1. Opens with a single headline insight that directly answers the audience's key question
2. Structures the story as: situation → complication → resolution (SCR framework)
3. Selects the 3 most compelling data points and writes a plain-English sentence for each
4. Recommends which charts to include and where in the narrative flow
5. Writes transitions between sections that maintain analytical logic
6. Anticipates the top 3 questions the audience will ask and provides pre-drafted answers with supporting data
7. Ends with a clear call to action tied to the business decision

```
Output: Full narrative script and a slide-by-slide outline.
```

#56 REPORTING & BI

Real-time Dashboard with Streaming Data Simulation

> *Act as a BI engineer. I need to build a real-time monitoring dashboard for* [use case: operations center / e-commerce live sales / server health / call center] *. The data updates every* [X seconds / minutes] *.*

Build a real-time dashboard that:

1. Simulates or connects to a streaming data source producing [describe metrics: orders per minute, error rate, active sessions, etc.]
2. Displays a live updating line chart for the primary metric over a rolling [X minute] window
3. Shows current-value KPI cards with delta vs. [X minutes ago] and color-coded status (green / amber / red)
4. Triggers a visible alert when any metric crosses a [user-defined threshold]
5. Logs all threshold breaches to a running alert table with timestamp and severity
6. Includes a toggle to pause / resume the live feed for investigation
7. Is deployable as a local web app requiring no external database

```
Use: Python (Dash / Streamlit / Panel) with threading or async data refresh — Provide full runnable code and a deployment guide.
```

#57 DIAGNOSTICS

Root Cause Analysis with Data

> *Act as an analytical problem-solver. A key metric (* [metric name, e.g., conversion rate / NPS / order volume] *) dropped by* [X%] *between* [date A] *and* [date B] *. I need to identify the root cause.*

Available data: [describe datasets available — e.g., web analytics, transaction logs, CRM data, support tickets]

Perform a data-driven root cause analysis by:

1. Confirming the metric drop is real (not a tracking error or data pipeline issue)
2. Breaking the metric down by all available dimensions (segment, region, channel, product)
3. Isolating which dimension(s) account for most of the drop using a contribution analysis
4. Analyzing whether the issue started suddenly or gradually
5. Cross-referencing with events log (releases, campaigns, external events) for that period
6. Forming and testing the top 3 hypotheses with statistical evidence
7. Recommending a corrective action for the most likely root cause with an implementation timeline

```
Output: Structured RCA report with supporting charts and evidence.
```

#58 WORKFLOW AUTOMATION

End-to-end Analytics Workflow Builder

> *Act as a senior data engineer and analyst. I need to build an automated end-to-end analytics workflow for* [use case, e.g., weekly sales reporting / monthly churn analysis / real-time fraud scoring] *.*

The workflow should:

1. Extract data from [source: database / API / CSV] on [schedule: daily / weekly / on trigger]
2. Run data validation checks before processing
3. Transform and aggregate data according to [business logic: describe the KPIs and groupings needed]
4. Apply [analytical model or rule, e.g., scoring model, statistical summary, forecasting function]
5. Output results to [destination: dashboard, email report, Slack alert, database table]
6. Log errors and send an alert if any step fails
7. Be fully reproducible and parameterizable so it can be run for any time period on demand

```
Use: Python (pandas, SQLAlchemy, schedule or Airflow DAG) — Provide full modular code with a README and deployment checklist.
```

#59 WORKFLOW AUTOMATION

Automated Reporting Pipeline with Email Delivery

> *Act as a data engineer. I need an automated reporting pipeline that pulls data, generates a formatted report, and emails it to* [stakeholders] *every* [daily / weekly / monthly] *on* [schedule] *.*

Build the pipeline that:

1. Connects to [data source: PostgreSQL / BigQuery / CSV / API] and runs the required query or extraction
2. Calculates the KPIs: [list metrics with formulas]
3. Generates a formatted report in [PDF / Excel / HTML email] with charts, summary tables, and period-over-period comparisons
4. Highlights metrics that are above or below target with color coding
5. Writes a plain-English narrative summary section that auto-populates based on the data values
6. Sends the report via email using [SMTP / SendGrid / AWS SES] to a configurable recipient list
7. Logs each run with status (success / failure), record count, and timestamp — alerts on failure via [Slack / email]

```
Use: Python (pandas, matplotlib, smtplib or sendgrid, schedule or cron) — Provide full modular code, a config file template, and a deployment checklist.
```

#60 SQL & DATABASES

SQL Query Optimization for Large Datasets

> *Act as a senior database engineer. I have a SQL query that is running slowly on a table with* [X million rows] *.*
> *Here is the current query:* [paste query] *.*

Analyze and optimize it by:

1. Identifying all performance bottlenecks (full table scans, missing indexes, unnecessary joins)
2. Rewriting the query with optimizations explained line by line
3. Recommending indexes to create and explaining the trade-offs
4. Estimating the expected performance improvement
5. Suggesting query execution plan analysis steps (EXPLAIN / EXPLAIN ANALYZE)

Database: [PostgreSQL / MySQL / BigQuery / Snowflake]

```
Output: Include before and after versions of the query with full line-by-line explanation.
```