DATA SCIENCE PROJECT GUIDE

Note:
Please remember this is a guide to develop / build an ideal project. You can choose to do the best at your capacity. Focus is to show how a full stack data science project looks like and analyze your understanding against that. Every team (solo/group) will have approx. ***10-12 min*** *to present their work. Therefore kindly, prepare presentation slides /video accordingly*

What You Will Do

You will complete an end-to-end data science project in three connected phases:
1. EXTRACT – Build or prepare a dataset.
2. LEARN – Train and evaluate machine learning models.
3. PREDICT – Deliver predictions through a Jupyter notebook **OR** Streamlit app. short pitch.
All three missions connect:
Part 1 gives you the data, Part 2 builds the model, and Part 3 produces predictions.

EXAMPLE THEMES

Choose ONE theme. State your choice clearly in your README. You can TURN THIS INTO classification or regression problem. BELOW GIVES EXAMPLE OF HOW YOU CAN DO IT AS CLASSIFICATION PROBLEM.

1. MUSIC
Detect artists/songs with hit potential despite low visibility.
Build dataset → classify "Hit or Not" → share predictions.

2. LAYOFFS
Classify companies as High vs Low Layoff Risk and highlight anomalies.

3. ENVIRONMENT
Classify regions as At-Risk vs Stable and optionally map hotspots.

LEARNING OUTCOMES

• Convert a real-world question into a structured data science question.
• Collect data, clean it, and perform feature engineering.
• Train baseline classification models (Decision Tree, Naive Bayes, KNN) and evaluate using accuracy, F1-score, confusion matrix, ROC (optional), etc.
• Communicate results via Jupyter Notebook or Streamlit + a short pitch deck.

TIERED FORMAT (Choose your level)

You can complete the assignment entirely using Core (Required) tasks.
You may complete Bonus tasks for additional credit.

PART 1 - OPERATION EXTRACT (Create Your Dataset) - 25 MARKS

CORE REQUIREMENTS:
• Use an existing public CSV (Kaggle or official portals). YOU MSUT BE ABLE TO EXPLAIN YOUR DATASET.
• Perform data processing

• Produce at least 2 visualizations (histogram, scatterplot, boxplot, correlation heatmap, etc.)

BONUS (Up to +10 Points):
• Build dataset from 2 or more APIs/web sources
• If you create your own dataset:
 – 100 < rows < 200
 – 10 < columns < 20
• Engineer at least 2 features

PART 2 — OPERATION LEARN (Model Training & Evaluation) - 25 MARKS

CORE REQUIREMENTS:
• Define a binary target (Hit=1, High Risk=1, At-Risk=1).
• Train at least the following models as examples given below
 – Decision Tree
 – Random Forest
• Report your metrices:
 – Accuracy
 – Confusion matrix
 – Explanation of model errors (false positives & false negatives)

BONUS (Up to +10 Points):
• Add XGBoost (example given, not mandatory unless you choose) (+3)
• Perform hyperparameter tuning (+4)
• Add F1-score and optionally ROC curve (+3)

PART 3 - OPERATION PREDICT (Final Output) - 25 MARKS

CORE REQUIREMENTS:
• Prepare a Jupyter Notebook demo that:
 – Loads a sample CSV
 – Runs predictions using your best model
 – Prints predictions
 – Includes summary + key observations
• Submit a one-page summary containing:
 Problem → Data → Model → Results → Limitations → Next Steps

BONUS (Up to +10 Points):
• Build a Streamlit application with:
 – CSV upload
 – Interactive predictions
• Theme-based bonus:
 – Music
 – Layoffs
 – Environment

PART 4 – Presentation and Communication – 15 marks
• Either you are creating PPT (5-10 slide)/(10-12 min) video, make sure you include these:
 1. Problem

2. Solution
3. Results
4. Users /usefulness
5. Future Work

On presentation day you can demo any one or the 2(PPT/video) + pitch you work

SAMPLE DATA SOURCES

MUSIC:

• APIs: Spotify Web API (audio features, popularity), YouTube API (views/likes), Last.fm API (scrobbles), Genius API (lyrics metadata), MusicBrainz.
• Public CSVs: "Spotify Tracks," "Top Spotify Songs 2023," "YouTube Trending."
• Feature ideas: danceability, tempo, valence, visibility, Cipher Risk Score (high quality but low visibility).

LAYOFFS:

• APIs/Feeds: Layoffs.fyi, Crunchbase, SEC EDGAR filings, Reddit sentiment, GDELT/News API.
• Public CSVs: Layoffs.fyi Kaggle dump, "Tech Layoffs 2022–2025."
• Feature ideas: funding-to-layoff ratio, review sentiment, burn-rate proxy, hiring freeze signal.

ENVIRONMENT:

• APIs/Portals: NASA FIRMS, USGS earthquakes/land cover, OpenWeatherMap, iNaturalist API, NOAA climate datasets, NPS API.
• Public CSVs: NASA FIRMS fire data, "Global Wildfires," USGS earthquake datasets.
• Feature ideas: fire density, biodiversity change %, weather anomaly index, drought index; optional map layers.

WHAT TO SUBMIT (GitHub Repository)

Your repository must include:

/Data
→ CSV(s) or script/notebook used for data collection & cleaning

/Notebooks
→ EDA notebook
→ Modeling notebook (TEST,TRAIN, EVALUATE)
→ Prediction demo notebook

/App (Bonus +5)
→ app.py (Streamlit interface)

/Model
→ Saved best model

/Slides
→ 5-10 slide PPT PDF or 10-12 min demo video

README.md
→ Clear instructions on how to run your project (environment, steps)

/summary
→ One-page observation summary (PDF)

RUBRIC (High-Level)

- Problem Framing — 10%
- Data Quality + Feature Engineering — 25%
- Modeling & Evaluation Correctness — 30%
- Communication (Notebook clarity, visuals) — 20%
- Reproducibility (README, code hygiene) — 15%
- Bonus — Up to +30 points

WORKED EXAMPLE (ALL DETAILS INCLUDED)

Theme: MUSIC
Goal: Predict "Hit" (=1) vs "Not Hit" using audio features & visibility. Identify "candidate" songs - high quality but low visibility.

SAMPLE EXPECTION FROM YOUR PROJECT

PART 1 — EXTRACT (CREATE DATA SET)

Data Sources:

- Kaggle: "Top Spotify Songs 2023"
- Bonus: Spotify API + YouTube API

Cleaning & Merging:

- Keep: artist, track, tempo, danceability, valence, loudness, energy, followers/views
- Drop duplicates
- Coerce numeric types
- Handle missing values
- Perform feature engineering

Feature Engineering

EDA(Examples):

- Histogram of quality_index
- Scatter: quality_index vs views
- Highlight high-quality/low-visibility songs

PART 2 - LEARN (TEST DATA SET ON DIFFERENT MODELS AND SELECT BEST ONE)

Define target:

- Hit = 1 if Spotify popularity ≥70 (or top 25th percentile)

Train models:

- Decision Tree
- Naive Bayes
- Bonus: KNN (with standardized features)

Evaluate:

- Accuracy
- Confusion matrix
- false positives vs false negatives
- Bonus: F1-score + ROC curve + tuning

Save model

PART 3 — PREDICT

Notebook Demo:

- Load sample CSV

• best_model.predict(df)
• Show Hit/Not Hit predictions
• Plot class count bar chart

Streamlit App (Bonus):

• File uploader
• Predictions table
• Cipher Risk Score ranking
• Feature importance chart

PART4 – Presentation and Pitch Deck (Bonus +5):

1. Problem (algorithmic suppression)
2. Solution (your ML pipeline)
3. Results (metrics + insights)
4. Users/usefullness (A&R, indie labels)
5. Future Improvements (more sources, fairness)

QUICK CHECKLIST

What I have to submit in canvas?

Git Repo Link + Working App (only if you choose to make a working app on streamlit) having following items:

• Picked one theme
• Built clean dataset
• Engineered features
• Create ≥2 visuals (using python libraries)
• Defined target
• Trained required model
• Evaluated properly
• Create. ipynb File Notebook only
• Summary page done
• Streamlit app (bonus)
• Model saved

.