Data Science Final Year Project Ideas PDF

A data science final year project should run end to end: collect and clean real data, store it in a database, analyse it, build a model where it helps, and present results in a dashboard that answers a business or social question. E-commerce delivery delays, taxi demand, market basket analysis and crop yield are strong choices.

Key takeaways

- Final-year data science projects are judged on the full pipeline, not only the model.
- Use SQL for storage and Python for analysis and modelling.
- End with a dashboard (Power BI, Tableau or Streamlit) and three clear findings.
- Public data from Kaggle, UCI, NYC TLC or data.gov.in is enough.
- The PDF gives plans and steps, not finished source code.

Who this PDF is for

This PDF is for final-year B.Tech, BCA, MCA and M.Sc students who want an end-to-end data science project. Unlike a single model notebook, each idea here has a data pipeline, analysis, a model and a dashboard, which gives a team of 3 or 4 a clear split of work. Each plan names public data and the questions your analysis should answer.

How to choose your project

- **Start from a question**, such as "Why are deliveries late?", not from an algorithm.
- **Check data size.** A few lakh rows is enough; tens of millions may need Spark.
- **Plan the dashboard users**: manager, farmer, city planner.
- **Split work by stage**: pipeline, analysis, model, dashboard.

Standard level projects

1. E-commerce Delivery Delay Analysis and Prediction

- **Problem:** Find why orders arrive late and predict late orders.
- **Dataset:** Kaggle "Brazilian E-Commerce Public Dataset by Olist".
- **Tech stack:** Python, PostgreSQL, pandas, LightGBM, Power BI or Streamlit.
- **Main modules:** Table joins in SQL, delay features, model, seller and region dashboard.
- **Steps:** Load tables into PostgreSQL, join orders and sellers, engineer distance and time features, predict delay, chart hotspots.

- **Show in the viva:** The three biggest delay drivers.
- **Extensions:** Link delays to review scores.

2. Market Basket Analysis for a Grocery App

- **Problem:** Find products often bought together for offers.
- **Dataset:** Kaggle "Instacart Market Basket Analysis".
- **Tech stack:** Python, pandas, mlxtend, SQL, Tableau.
- **Main modules:** Data sampling, association rules, reorder prediction, offer dashboard.
- **Steps:** Sample orders, run Apriori or FP-Growth, filter rules by lift, build a reorder model.
- **Show in the viva:** Support, confidence and lift in plain words.
- **Extensions:** Recommend bundles per aisle.

3. Customer Lifetime Value and Segmentation

- **Problem:** Estimate future value of customers and group them.
- **Dataset:** UCI "Online Retail II".
- **Tech stack:** Python, pandas, lifetimes library, scikit-learn, Power BI.
- **Main modules:** Cleaning returns, RFM, clustering, CLV model, segment dashboard.
- **Steps:** Remove cancellations, compute RFM, cluster, fit BG/NBD and Gamma-Gamma models, rank customers.
- **Show in the viva:** How CLV changes the marketing budget per segment.
- **Extensions:** Add a churn flag.

4. Indian Air Quality Insights Dashboard

- **Problem:** Show pollution trends and seasonal patterns by city.
- **Dataset:** Kaggle "Air Quality Data in India (2015 - 2020)".
- **Tech stack:** Python, pandas, SQL, Plotly Dash.
- **Main modules:** Gap filling, AQI trend analysis, festival and season comparisons, forecast panel.
- **Steps:** Clean data, compare months and cities, test differences, add a short-term forecast.
- **Show in the viva:** One finding backed by a statistical test.
- **Extensions:** Add weather data.

Advanced level projects

5. City Taxi Demand Forecasting

- **Problem:** Predict taxi pickups by zone and hour.
- **Dataset:** NYC Taxi and Limousine Commission (TLC) trip record data.
- **Tech stack:** Python, DuckDB or PySpark, LightGBM, Streamlit with maps.
- **Main modules:** Big file ingestion, zone-hour aggregation, weather features, forecasting, map dashboard.
- **Steps:** Load a few months with DuckDB, aggregate by zone and hour, forecast with time-based validation.
- **Show in the viva:** How you handled data that does not fit in pandas.
- **Extensions:** Add surge-pricing simulation.

6. Retail Sales Forecast and Promotion Impact

- **Problem:** Forecast store sales and measure promo effect.
- **Dataset:** Kaggle "Rossmann Store Sales".
- **Tech stack:** Python, pandas, Prophet, LightGBM, Power BI.
- **Main modules:** Feature pipeline, forecasting, promo uplift analysis, store dashboard.
- **Steps:** Build time features, forecast, compare promo and non-promo days with matched stores.
- **Show in the viva:** Correlation versus causation in promo analysis.
- **Extensions:** Optimise promo calendar.

7. Crop Production Analysis for Indian States

- **Problem:** Find trends in crop production and area by state.
- **Dataset:** Crop production data published on data.gov.in (check the latest file).
- **Tech stack:** Python, pandas, SQL, Tableau.
- **Main modules:** Data cleaning across years, yield calculation, state comparisons, simple forecast.
- **Steps:** Standardise crop and state names, compute yield, chart trends, forecast for major crops.
- **Show in the viva:** How you cleaned inconsistent names across years.
- **Extensions:** Add rainfall data from data.gov.in.

8. Hospital Readmission Risk

- **Problem:** Predict which diabetic patients may return within 30 days.
- **Dataset:** UCI "Diabetes 130-US Hospitals for Years 1999-2008".
- **Tech stack:** Python, scikit-learn, XGBoost, SHAP, Streamlit.
- **Main modules:** Cleaning codes, feature groups, model, explanation, care-team dashboard.
- **Steps:** Group diagnosis codes, handle missing values, train XGBoost, explain with SHAP.
- **Show in the viva:** Ethical use of patient risk scores.
- **Extensions:** Add cost-weighted evaluation.

Research-oriented projects

9. A/B Test Analysis Framework

- **Problem:** Build a tool to analyse online experiments correctly.
- **Dataset:** Public A/B testing datasets on Kaggle, or simulated data you generate.
- **Tech stack:** Python, SciPy, statsmodels, Streamlit.
- **Main modules:** Sample size calculator, significance tests, confidence intervals, peeking simulation.
- **Steps:** Implement tests, simulate repeated peeking, show how false positives rise.
- **Show in the viva:** The peeking simulation chart.
- **Extensions:** Add Bayesian A/B testing.

10. Data Quality Monitoring Pipeline

- **Problem:** Detect bad or drifting data before it reaches dashboards.
- **Dataset:** Any dataset above, with errors you inject.

- **Tech stack:** Python, Great Expectations or pandera, Airflow or cron.
- **Main modules:** Rules, drift checks, alert report, pipeline schedule.
- **Steps:** Write rules, inject errors, run checks on a schedule, report failures.
- **Show in the viva:** Which injected errors were caught.
- **Extensions:** Add a data catalogue page.

11. Explainable Credit Risk Scorecard

- **Problem:** Build an interpretable credit score and compare with a black-box model.
- **Dataset:** UCI "Default of Credit Card Clients" dataset.
- **Tech stack:** Python, scikit-learn, optbinning or WoE methods, XGBoost, SHAP.
- **Main modules:** Binning and WoE, logistic scorecard, XGBoost, comparison, fairness check.
- **Steps:** Build the scorecard, train XGBoost, compare AUC and explainability.
- **Show in the viva:** Why banks often prefer scorecards.
- **Extensions:** Add reject-reason codes.

12. Public Transport Ridership Study

- **Problem:** Analyse metro or bus ridership patterns.
- **Dataset:** Open ridership data from a city transport portal, or data.gov.in where available.
- **Tech stack:** Python, pandas, GeoPandas, Kepler.gl or Plotly.
- **Main modules:** Data ingestion, station-level patterns, peak-hour analysis, map.
- **Steps:** Clean ridership files, find peak hours and busiest stations, map flows.
- **Show in the viva:** One planning suggestion supported by the data.
- **Extensions:** Forecast ridership for new stations.

Report structure

A final-year report is marked on clarity as much as on code. A structure most universities accept:

1. **Abstract** (150 to 250 words): problem, method, main result.
2. **Introduction:** background, problem statement, objectives and scope.
3. **Literature review:** 8 to 15 papers or existing systems, and the gap your project fills.
4. **Requirements and design:** functional and non-functional requirements, architecture diagram, the data pipeline diagram (source to database to analysis to dashboard) and the database schema.
5. **Implementation:** modules, tools, key algorithms and screenshots.
6. **Testing and results:** key findings with charts, model metrics against a baseline, and dashboard screenshots.
7. **Conclusion and future scope.**
8. **References** in the format your department asks for (IEEE style is common).

Follow your own university's template if it gives one; it overrides this list.

Common mistakes to avoid

- Building a model before understanding the data.

- A dashboard full of charts but no clear findings.
- Ignoring data licences and not citing sources.
- Claiming cause and effect from correlation alone.

Frequently asked questions

What makes a data science project final-year level?

An end-to-end pipeline: real data, cleaning, a database, analysis, a model where useful, and a dashboard with clear findings.

Which tools should I use?

Python (pandas, scikit-learn), SQL, and a dashboard tool such as Power BI, Tableau or Streamlit. Use DuckDB or Spark only if the data is too large for pandas.

Is a dashboard compulsory?

Not always, but it makes results easy to see in the viva. Check with your guide.

Where can I find Indian datasets?

data.gov.in publishes many government datasets. Kaggle also hosts Indian data such as air quality records. Always check the licence.

Does this PDF include full source code?

No. It gives the problem, data, stack, modules and steps for each project. You write the code yourself, which is what examiners test in the viva.

Is this PDF free?

Yes. Fill in the short form on this page and the PDF downloads right away.

Sources and further reading

- Kaggle Datasets
- UCI Machine Learning Repository
- NYC TLC trip record data
- Open Government Data Platform India

About this PDF

Prepared by the Uncodemy training team as practice project ideas. Datasets and tools named are public; check each dataset's licence before use. We update it every January. Corrections: info@uncodemy.com.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550