Mini Projects for AI & ML Students PDF

AI and ML mini projects should take one to four weeks, use a small public dataset and teach one core idea, such as regression, classification, clustering or a basic neural network. Good examples are Iris or Titanic classification, house price regression, a spam filter, digit recognition, customer segmentation and a simple chatbot. Finish each one with a short write-up and a notebook on GitHub.

Key takeaways

- A mini project teaches one concept well. Do not try to build a full product.
- Each idea lists the dataset, model, steps, viva points and a rough cost.
- Most mini projects cost nothing using Google Colab or Kaggle notebooks. Any costs are approximate and vary by city and vendor.
- Always split data into train and test sets and report one clear metric.
- Two or three well-explained mini projects prepare you for a strong final year project.

How to choose an AI and ML mini project

- **Match it to your syllabus.** If this semester covers classification, build a classifier.
- **Use a small dataset.** Small data trains fast on a laptop or free notebook.
- **Learn the full workflow.** Load, clean, explore, model, evaluate, explain.
- **Add a tiny demo** with Streamlit or Gradio if time allows.

Easy mini projects (1 to 2 weeks)

1. Iris flower classification

- **Data:** Iris dataset (built into scikit-learn).
- **Models:** k-nearest neighbours, decision tree.
- **Steps:** explore with plots, split, train, compare accuracy, show confusion matrix.
- **Viva points:** choosing k, overfitting in trees.
- **Approximate cost:** Rs 0 (varies by city and vendor).

2. Titanic survival prediction

- **Data:** Kaggle Titanic dataset.
- **Models:** logistic regression, random forest.
- **Steps:** handle missing ages, encode categories, train, submit to Kaggle.
- **Viva points:** feature engineering, why accuracy is not the only metric.
- **Approximate cost:** Rs 0 (varies by city and vendor).

3. House price regression

- **Data:** a public housing dataset such as California housing.
- **Models:** linear regression, ridge, random forest regressor.
- **Viva points:** MAE, RMSE, R squared, multicollinearity.
- **Approximate cost:** Rs 0 (varies by city and vendor).

4. SMS spam filter

- **Data:** public SMS spam collection.
- **Models:** bag of words or TF-IDF with naive Bayes.
- **Viva points:** Bayes theorem, stop words, precision versus recall.
- **Approximate cost:** Rs 0 (varies by city and vendor).

5. Handwritten digit recognition

- **Data:** MNIST.
- **Models:** a simple neural network, then a small CNN.
- **Viva points:** activation functions, softmax, convolution and pooling.
- **Approximate cost:** Rs 0 (varies by city and vendor).

Medium mini projects (2 to 3 weeks)

6. Customer segmentation with k-means

- **Data:** public mall customers or retail datasets.
- **Steps:** scale features, elbow method, silhouette score, describe each segment.
- **Viva points:** why scaling matters, choosing k, limits of k-means.
- **Approximate cost:** Rs 0 (varies by city and vendor).

7. Movie review sentiment analysis

- **Data:** IMDb reviews dataset.
- **Models:** TF-IDF with logistic regression, then a pretrained transformer for comparison.
- **Viva points:** tokenisation, embeddings, handling sarcasm.
- **Approximate cost:** Rs 0 (varies by city and vendor).

8. Face detection and blur tool

- **Tools:** OpenCV Haar cascades or a pretrained face detector.
- **Steps:** detect faces in webcam or images, blur them, save output.
- **Viva points:** sliding windows, why privacy tools matter.
- **Approximate cost:** Rs 0 with a laptop webcam (varies by city and vendor).

9. Simple FAQ chatbot

- **Tools:** Python, TF-IDF similarity or sentence embeddings.
- **Steps:** build a FAQ list, match user questions to the closest answer, fallback message.
- **Viva points:** cosine similarity, thresholds, why a fallback is needed.
- **Approximate cost:** Rs 0 (varies by city and vendor).

10. Stock price trend visualiser with a baseline forecast

- **Data:** public historical price data.
- **Models:** moving average, simple linear trend.
- **Viva points:** why past prices are a weak predictor, time-based splits.
- **Approximate cost:** Rs 0 (varies by city and vendor).

11. Heart disease risk classifier

- **Data:** UCI heart disease dataset.
- **Models:** logistic regression, SVM.
- **Viva points:** sensitivity, ROC curve, why this is a learning exercise and not medical advice.
- **Approximate cost:** Rs 0 (varies by city and vendor).

Stretch mini projects (3 to 4 weeks)

12. Image classifier with transfer learning and a web demo

- **Data:** a small image set you collect, such as 3 to 5 classes of objects.
- **Models:** MobileNet fine-tuned in Keras or PyTorch.
- **Steps:** collect and label images, train, deploy with Gradio.
- **Viva points:** freezing layers, augmentation, small-data limits.
- **Approximate cost:** Rs 0 (varies by city and vendor).

13. Text summariser using a pretrained model

- **Tools:** Hugging Face transformers.
- **Steps:** summarise news articles, compare lengths, judge quality on a few samples.
- **Viva points:** extractive versus abstractive summaries, ROUGE score.
- **Approximate cost:** Rs 0 (varies by city and vendor).

14. Gesture-controlled media player

- **Tools:** OpenCV, MediaPipe hand landmarks.
- **Steps:** detect hand, map gestures to play, pause and volume.
- **Viva points:** landmark detection, frame rate, lighting issues.
- **Approximate cost:** Rs 0 (varies by city and vendor).

15. Mini RAG over one PDF

- **Tools:** an embedding model, FAISS, an open model or LLM API, Streamlit.
- **Steps:** split one PDF into chunks, embed, retrieve top chunks, generate an answer with page references.
- **Viva points:** embeddings, retrieval, why answers should cite sources.
- **Approximate cost:** Rs 0 to 300 for API calls (varies by city and vendor).

16. Reinforcement learning agent for a simple game

- **Tools:** Gymnasium CartPole or FrozenLake, Python.
- **Models:** Q-learning, then a small DQN.
- **Viva points:** reward, exploration versus exploitation, discount factor.

- **Approximate cost:** Rs 0 (varies by city and vendor).

A simple 3-week plan

- **Week 1:** pick the dataset, read its description and licence, and explore it with summary statistics and plots.
- **Week 2:** build a baseline, then one better model. Keep a small table of settings and scores.
- **Week 3:** look at errors, write a short explanation of what the model learned, add a README and, if time allows, a Streamlit or Gradio demo.

Write the notebook as if a classmate will read it: short markdown notes before each step, clear plot titles and a final summary cell. This habit makes your later final year report and job interviews much easier.

Mini project report structure

Keep it to 15 to 30 pages:

1. Title page and certificate
2. Introduction and objective
3. Dataset: source, size, features
4. Method: preprocessing and model
5. Results: metric table and one or two plots
6. Conclusion and limitations
7. References

Use the project report title page generator for the first page.

Common mistakes to avoid

- Training and testing on the same data.
- Reporting only accuracy on imbalanced data.
- Copying a Kaggle notebook without understanding each cell.
- Skipping data exploration before modelling.
- No README explaining how to run the notebook.

Building on your mini projects

Each mini project can become part of a bigger final year project. A sentiment classifier can grow into a review analysis app with a dashboard, and an image classifier can move onto a small board for edge AI. Keep every notebook clean, with a short README that states the dataset, the metric and what you would try next. Recruiters and examiners both read these first.

Frequently asked questions

What are good mini projects for AI and ML students?

Iris or Titanic classification, house price regression, a spam filter, digit recognition, customer segmentation and a simple FAQ chatbot are good starting points.

How long does an AI and ML mini project take?

Usually one to four weeks. Easy projects on built-in datasets can be done in a week.

Do I need to pay for GPU or software?

Usually not. Google Colab and Kaggle notebooks are free with limits, and the main libraries are open source. Any costs are approximate and vary by city and vendor.

Which language should I use?

Python is the standard choice, with pandas, scikit-learn and PyTorch or TensorFlow.

How is a mini project different from a final year project?

A mini project focuses on one concept with a small dataset. A final year project solves a real problem end to end with baselines, error analysis and a deployed demo.

Is this PDF free?

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

Sources and further reading

- scikit-learn user guide
- Kaggle Learn
- TensorFlow tutorials
- Hugging Face documentation
- Gymnasium documentation

About this PDF

Prepared by the Uncodemy trainer team as a practical mini project guide. Free tier limits change, so treat costs as approximate. Corrections: info@uncodemy.com.

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