# Prompt Engineering Guide for Data Scientists & ML Engineers

This comprehensive playbook is designed as a progressive, end-to-end resource—like a "Prompt Engineering Book"— tailored for data scientists and ML engineers. It is structured in chapters and levels (Beginner → Intermediate → Advanced → Expert) and delivers a practical blend of ready-to-use prompts, pedagogical rationale, real-world scenarios, and expert variations, so you can copy, learn, and adapt each prompt for your actual work.

## Table of Contents

# 1. Introduction to Prompt Engineering

## What Is Prompt Engineering?

Prompt engineering is the systematic craft of designing precise instructions for large language models (LLMs) to achieve domain-aligned, high-quality results in coding, analysis, modeling, and more. In modern data workflows, prompt engineering is as critical as coding or statistical acumen.[1][2][3][4][5][6]

**Core Principles:**

- **Role specification** (e.g., "act as a senior data scientist")
- **Context and constraints** ("here is a summary table, perform cleaning…")

- **Format and style** (“return the output as Python code with stepwise comments”)
- **Progressive refinement and example-driven prompting** (zero-shot, few-shot, chain-of-thought)
- **Task decomposition and prompt chaining**
- **Output validation** (“explain your logic”).[7][2][8][9][6][10]

## 2. Data Cleaning & Preprocessing Prompts

### Beginner

**Prompt:**

```
You are a data cleaning assistant. I will give you a pandas DataFrame summary with
missing values, duplicates, and inconsistent formats. Suggest the best Python code to
clean it while explaining each step.
```

- **Purpose:** Automates basic cleaning for newcomers.
- **Why it works:** Role assignment, context inclusion, and explicit instruction.
- **Use Case:** Cleaning messy open datasets.
- **Expert Variation:**
  “Suggest multiple data-cleaning strategies, explain trade-offs, and generate Python code optimized for speed with pandas and polars.”

### Intermediate

**Prompt:**

```
Given the following dataset description and sample rows, identify possible data issues
(missing values, outliers, invalid types). Recommend specific preprocessing steps for
each, and provide the pandas or pyjanitor code, annotated line by line.
```

- **Purpose:** Structured troubleshooting for common data issues.
- **Why it works:** Task decomposition; demand for line-by-line explanation.

- **Use Case:** Initial data audit in a new project.
- **Expert Variation:**
  "For each identified issue, analyze at least two alternative code approaches (e.g., imputation methods) and evaluate their pros/cons for this dataset."

## 3. Exploratory Data Analysis (EDA) Prompts

### Beginner

**Prompt:**

```
Given a DataFrame schema and column types, suggest a list of the most useful univariate
and bivariate summary statistics and visualizations to explore this dataset. Provide
code snippets using pandas, seaborn, or matplotlib.
```

- **Purpose:** Jump-starts EDA for new data practitioners.
- **Why it works:** Concrete schema can be directly mapped to visual and statistical tasks.
- **Use Case:** Starting a Kaggle challenge with unfamiliar data.
- **Expert Variation:**
  "Given sample data and business goals, devise a tailored EDA workflow, justify analysis decisions, and produce a Jupyter-ready pipeline using ydata-profiling or Sweetviz. Highlight key actionable insights."

### Intermediate

**Prompt:**

```
I have a sales_dataset with columns: date, product_id, region, units_sold, and price.
Create an EDA checklist with step-by-step Python code to examine seasonality, outliers,
and sales trends. Include code for at least three plots and interpret the results.
```

- **Purpose:** Teaches not just code but interpretation, guiding best practices.
- **Why it works:** Explicit context + demand for interpretation pushes output quality.

- **Use Case:** Preparing a dataset for forecasting models.
- **Expert Variation:**
  "For time-series, automate STL decomposition and annotate detected change-points. Generate an EDA report that flags anomalies and suggests domain-specific follow-ups."

## 4. Feature Engineering Prompts

### Beginner

**Prompt:**

```
I am building a regression model to predict house prices. Columns: square_feet,
bedrooms, bathrooms, year_built, neighborhood. Suggest 5 derived features and write
pandas code for each.
```

- **Purpose:** Moves from theory to practical feature creation.
- **Why it works:** Direct and outcome-focused.[11][12][13]
- **Use Case:** Improving baseline models in hackathons.
- **Expert Variation:**
  "Given past model stats and SHAP importances, propose novel features that maximize mutual information with the target variable. Output your logic in JSON format and explain potential trade-offs."

### Advanced

**Prompt:**

```
You are an NLP feature engineer. For a collection of user reviews, generate code to
extract sentiment, top keywords, and language complexity features. Output the pipeline
as a Scikit-learn Transformer class.
```

- **Purpose:** Automates sophisticated, reproducible feature pipelines.
- **Why it works:** Role and class-based output structures, driving engineering best practices.[13]

- **Use Case:** Productionizing NLP workflows.
- **Expert Variation:**
  "For multimodal data, suggest cross-modal features (e.g., text/image fusion). Provide pseudocode and cite relevant benchmarking literature."

## 5. Model Building & Debugging Prompts

### Beginner

**Prompt:**

```
Given my sklearn RandomForestClassifier with high training but low validation accuracy,
list 3 reasons for overfitting and provide code suggestions to fix each.
```

- **Purpose:** Guides troubleshooting for a classic ML problem.
- **Why it works:** Encourages chain-of-thought (reasoning + solution).
- **Use Case:** Early-phase project debugging.
- **Expert Variation:**
  "Provide a systematic debugging checklist that spans feature selection, data leaks, hyperparameter tuning, and novel techniques (e.g., elastic net, stacking). Output code templates and highlight when each technique is most effective."

### Advanced

**Prompt:**

```
My model shows unexpected prediction drift between two time periods. Analyze potential
causes (concept drift, feature distribution changes), suggest statistical tests for
drift detection, and generate code for addressing it in production pipelines.
```

- **Purpose:** Brings in advanced concepts like monitoring and mitigation.[14][15][16]
- **Why it works:** Encourages holistic, reproducible engineering solutions.
- **Use Case:** Maintaining deployed ML systems.

- **Expert Variation:**
  "Automate model monitoring dashboards in Python, recommend deployment architecture for drift alerts, and explain how to integrate human-in-the-loop interventions."

## 6. Research & Paper Summarization Prompts

### Beginner

**Prompt:**

```
Summarize the following ML research paper in 5 bullet points. Include the main contribution, methods, results, and any noted limitations. Explain how one might adapt the approach in scikit-learn or PyTorch.
```

- **Purpose:** Translating academic research into practice.
- **Why it works:** Structured summary aligns with common literature review needs.[17][18]
- **Use Case:** Bridging the gap between papers and prototyping.
- **Expert Variation:**
  "Summarize the same paper (a) as a PhD-level technical note, (b) as a Kaggle cheat sheet, (c) for C-suite non-technical stakeholders. Provide PyTorch or TensorFlow pseudocode for reproduction."

### Advanced

**Prompt:**

```
Given this set of new research abstracts, cluster papers by research area, summarize main trends, and extract emerging open problems. Suggest which may be ripe for industry adoption in the next year.
```

- **Purpose:** Research landscape mapping and technical trend scouting.
- **Why it works:** Combines summarization, clustering, and practical insight.
- **Use Case:** Technical due diligence for startups or product teams.

- **Expert Variation:**
  "Generate a citation map with cross-domain links, predict trend inflections, and recommend actionable next steps for in-house R&D."

## 7. Communication & Reporting Prompts

### Beginner

**Prompt:**

```
Summarize the model results for executives: state key metrics (accuracy, AUC), list 3
most important features in plain English, and briefly describe business impact. Avoid
technical jargon.
```

- **Purpose:** Bridges the gap between technical and business audiences.[19][1]
- **Why it works:** Explicit guidance for tone, complexity, and content.
- **Use Case:** Model handover meetings, stakeholder updates.
- **Expert Variation:**
  "Tailor model performance summaries in tabular format for (a) executives, (b) ML engineers, (c) compliance officers. Highlight critical trade-offs and suggest next actions for each audience."

## 8. Advanced Prompt Engineering Techniques for ML

### Chaining & Decomposition

**Prompt:**

```
For this full ML workflow (data cleaning → feature engineering → modeling → reporting),
break the task into modular steps. For each, design a separate prompt chain and document
inputs/outputs. Output your prompt chain as a configuration file for orchestration
platforms (e.g., LangChain, Prefect).
```

- **Purpose:** Moves from single-task to full workflow orchestration.[9][20][21]
- **Why it works:** Mimics engineering best practices—modular, testable, auditable chains.

- **Use Case:** Production, repeatable MLOps architectures.
- **Expert Variation:**
  "Include prompt self-evaluation (Reflexion), test-driven prompting, and multi-agent designs where outputs of one prompt become context for the next."

## Meta-Prompting & Self-Refinement

**Prompt:**

```
Act as a prompt engineer for LLMs. Evaluate this prompt for ambiguity, context, and
structure. Suggest three refinements and explain the reasoning behind each.
```

- **Purpose:** Teaches critical self-review—a must as prompt chains grow in complexity.[22][21][10]
- **Why it works:** Iterative, test-driven engineering enhances reliability at scale.
- **Use Case:** Team prompt library maintenance; version control.
- **Expert Variation:**
  "Develop a prompt evaluation rubric with objective quality metrics (clarity, specificity, consistency), design A/B tests, and automate prompt selection with an LLM-based prompt optimizer."

## Specialty: Multimodal Prompts

**Prompt:**

```
Given this time-series chart and related text dataset, propose a multimodal analysis
workflow (detect anomalies, extract events), provide Python code, and specify how to
structure the prompt for an LLM with vision capabilities (e.g., GPT-4o).
```

- **Purpose:** Enables prompt engineering with multimodal models common in advanced ML.[23][24]
- **Why it works:** Incorporates both textual and visual context, requiring richer LLM instructions.
- **Use Case:** Automated report creation with mixed input data.

- **Expert Variation:**
  "Chain multimodal prompts to surface cross-domain insights (e.g., connect video features to time-series sensor stats in predictive maintenance). Propose schema for storing intermediate outputs."

## 9. Appendices: Frameworks, Tools, and Further Reading

- **Frameworks:** RACE, TRACE, RECAP, Agile Prompt Engineering, ReAct, Reflexion, Chain-of-Thought (CoT), Self-Consistency, Retrieval-Augmented Generation (RAG), Automatic Prompt Engineering (APE).[25][4][5][10][22]
- **Tools:** LangChain, PromptLayer, Mirascope, torchtune Prompt Templates, Python LangChain Prompt Templates.[26][20][21]
- **Best Practices:**
  - Be clear, specific, and action-oriented.[8][6][9]
  - Specify format and constraints (output type, length, style).[2][5][6]
  - Show by example (few-shot), and incrementally build prompt complexity.[27][6][1][9]
  - Use role/goal framing and chain-of-thought reasoning.
  - Test and iterate—expect prompt evolution as models and requirements change.[10][1]

## Conclusion

Prompt engineering is the "new feature engineering"—the core lever for data scientists and ML engineers in the LLM era. This guide empowers you to automate, optimize, and scale your workflows, from data cleaning to communication, and from research to production MLOps, using the latest LLM advances. Each prompt template can be plugged directly into your favorite LLM interface, modified for your context, or chained as part of a custom AI workflow.[11]

*I'm on here if you want to follow:*
GitHub - https://github.com/AdilShamim8

LinkedIn - https://www.linkedin.com/in/adilshamim8

Twitter(x) - https://x.com/adil_shamim8

1. https://dev.to/zerozulu/prompt-engineering-techniques-every-data-scientist-should-know-2025-guide-25gg
2. https://developers.google.com/machine-learning/resources/prompt-eng
3. https://www.dataquest.io/blog/introduction-to-prompt-engineering-for-data-professionals/
4. https://platform.openai.com/docs/guides/prompt-engineering
5. https://www.promptingguide.ai/guides/optimizing-prompts
6. https://mirascope.com/blog/prompt-engineering-best-practices
7. https://www.semanticscholar.org/paper/bd1a119141713d83ef901171dc20541433b36b1e
8. https://www.geeksforgeeks.org/prompt-engineering-best-practices/
9. https://www.promptingguide.ai/techniques
10. https://learnprompting.org/courses/advanced-prompt-engineering
11. https://www.linkedin.com/pulse/prompt-engineering-new-feature-debasish-deb-gwszf
12. https://www.datapro.news/p/5-ai-prompts-boost-productivity-data-engineers
13. https://towardsdatascience.com/advanced-prompt-engineering-for-data-science-projects/
14. https://www.ijfmr.com/research-paper.php?id=52779
15. https://semiwiki.com/artificial-intelligence/359520-a-quick-tour-through-prompt-engineering-as-it-might-apply-to-debug/
16. https://www.codestringers.com/insights/prompt-debugging/
17. https://blog.promptlayer.com/best-prompts-for-text-summarization-guide-to-ai-summaries/
18. https://www.linkedin.com/pulse/llm-prompt-research-paper-analysis-simplifying-academic-hani-simo-rp24f
19. https://towardsdatascience.com/become-a-better-data-scientist-with-these-prompt-engineering-hacks/
20. https://python.langchain.com/docs/concepts/prompt_templates/
21. https://mirascope.com/blog/advanced-prompt-engineering
22. https://www.mercity.ai/blog-post/advanced-prompt-engineering-techniques
23. https://www.semanticscholar.org/paper/cbad78cbedb24f2fe5c3315e376ef32667687980

24. https://arxiv.org/abs/2405.10689

25. https://www.linkedin.com/pulse/prompt-engineering-frameworks-comprehensive-guide-aman-saxena-3e3lc

26. https://docs.pytorch.org/torchtune/0.6/basics/prompt_templates.html

27. https://www.coursera.org/articles/prompt-engineering-examples

28. https://index.ieomsociety.org/index.cfm/article/view/ID/26678

29. https://onlinelibrary.wiley.com/doi/10.1002/sdr.70008

30. https://arxiv.org/pdf/2402.14837.pdf

31. http://arxiv.org/pdf/2410.08696.pdf

32. https://arxiv.org/pdf/2311.08364.pdf

33. https://arxiv.org/pdf/2407.12994.pdf

34. https://www.ijisrt.com/prompt-engineering-methodology

35. https://jds-online.org/doi/10.6339/25-JDS1184

36. https://www.semanticscholar.org/paper/9b0e385ecc4b1665b25712c513cc219156a7c65c

37. http://peer.asee.org/23003

38. https://www.ewadirect.com/proceedings/ace/article/view/17687

39. https://www.mdpi.com/2079-9292/13/15/2961

40. https://www.ijsr.net/getabstract.php?paperid=SR220610115023

41. https://arxiv.org/abs/2402.05129

42. https://www.ssrn.com/abstract=4984967

43. http://arxiv.org/pdf/2406.06608.pdf

44. https://arxiv.org/pdf/2412.05127.pdf

45. http://arxiv.org/pdf/2401.14423.pdf

46. https://arxiv.org/pdf/2402.07927.pdf

47. https://arxiv.org/pdf/2401.08189.pdf

48. http://arxiv.org/pdf/2502.11560.pdf

49. http://arxiv.org/pdf/2405.18369.pdf

50. https://arxiv.org/html/2411.06099v1

51. https://arxiv.org/pdf/2302.11382.pdf

52. https://arxiv.org/html/2409.08775v1

53. https://www.k2view.com/blog/prompt-engineering-techniques/

54. https://www.tredence.com/blog/prompt-engineering-best-practices-for-structured-ai-outputs

55. https://www.hostinger.com/in/tutorials/ai-prompt-engineering

56. https://cloud.google.com/discover/what-is-prompt-engineering

57. https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/prompt-engineering

58. https://aws.amazon.com/what-is/prompt-engineering/

59. https://ieeexplore.ieee.org/document/10791375/

60. https://pubs.acs.org/doi/10.1021/acscentsci.4c01935

61. https://www.cambridge.org/core/product/identifier/S1351324923000438/type/journal_article

62. https://link.springer.com/10.1007/s41019-023-00235-6

63. https://www.ssrn.com/abstract=4429658

64. https://ieeexplore.ieee.org/document/10466952/

65. https://www.semanticscholar.org/paper/26b986905dd5ec01b05eb16ea5c9fcb3a6ab7564

66. http://ijraset.com/fileserve.php?FID=19829

67. https://dl.acm.org/doi/10.1145/3663649.3664368

68. http://ieeexplore.ieee.org/document/6051796/

69. https://arxiv.org/pdf/2402.16932.pdf

70. http://arxiv.org/pdf/2410.00880.pdf

71. http://arxiv.org/pdf/2210.01848v2.pdf

72. http://arxiv.org/pdf/2409.16635.pdf

73. https://arxiv.org/pdf/2307.12980.pdf

74. https://arxiv.org/pdf/2308.03854.pdf

75. https://arxiv.org/pdf/2502.16965.pdf

76. https://www.packtpub.com/en-us/learning/how-to-tutorials/chatgpt-for-exploratory-data-analysis-eda

77. https://cloud.google.com/vertex-ai/generative-ai/docs/learn/prompts/prompt-templates

78. https://www.coursera.org/articles/exploratory-data-analysis

79. https://clickup.com/templates/ai-prompts/machine-learning

80. https://facultyecommons.com/ai-prompt-templates-and-examples-for-course-content/

81. https://arxiv.org/abs/2503.19742

82. https://dl.acm.org/doi/10.1145/3676536.3676801

83. https://learning-analytics.info/index.php/JLA/article/view/8575

84. https://dl.acm.org/doi/10.1145/3674399.3674482

85. https://arxiv.org/abs/2507.05296

86. https://ieeexplore.ieee.org/document/11030993/

87. https://ojs.bbwpublisher.com/index.php/JCER/article/view/10508

88. https://www.semanticscholar.org/paper/7bc59bb5c19c98df31f2b29aeb3622946dacc3f0

89. https://arxiv.org/pdf/2503.05070.pdf

90. http://arxiv.org/pdf/2309.06135.pdf

91. https://arxiv.org/pdf/2406.12334.pdf

92. http://arxiv.org/pdf/2309.09128v3.pdf

93. http://arxiv.org/pdf/2412.09722.pdf

94. https://arxiv.org/html/2410.01242

95. http://arxiv.org/pdf/2307.06865.pdf

96. https://arxiv.org/pdf/2304.02195.pdf

97. https://arxiv.org/pdf/2311.05661.pdf

98. https://blog.promptlayer.com/best-prompts-for-asking-a-summary-a-guide-to-effective-ai-summarization/

99. https://www.v7labs.com/blog/prompt-engineering-guide

100. https://jisem-journal.com/index.php/journal/article/view/9035

101. https://ijsrem.com/download/expanding-horizons-in-prompt-engineering-techniques-frameworks-and-challenges/

102. https://www.mdpi.com/2079-9292/14/11/2281

103. https://arxiv.org/abs/2412.08593

104. https://ieeexplore.ieee.org/document/10541229/

105. https://www.ijfmr.com/research-paper.php?id=33981