**5 high-value use cases with prompt templates** for a **data analyst transitioning to data science**:

### 1. Statistical Analysis & Hypothesis Testing

* 📊 Prompt template:  
  "Given this dataset: [insert dataset], perform statistical analysis. Test hypotheses, calculate correlations, p-values, confidence intervals, and summarize findings in plain language."
* 🔧 Use case: Learning inferential statistics and drawing conclusions from data.

### 2. Machine Learning Modeling

* 🤖 Prompt template:  
  "Build a predictive model for [target variable] using this dataset. Compare at least 3 algorithms (e.g., linear regression, random forest, XGBoost). Evaluate using metrics [accuracy, RMSE, etc.] and explain which model performs best and why."
* 🔧 Use case: Practicing end-to-end ML workflows and model evaluation.

### 3. Feature Engineering & Selection

* 🧩 Prompt template:  
  "Suggest and create 5–10 new features from this dataset for a predictive model. Explain why each feature might improve performance, and show Python code for implementation."
* 🔧 Use case: Understanding how to enhance models with better data representation.

### 4. Data Visualization & Storytelling

* 📈 Prompt template:  
  "Generate visualizations for this dataset that reveal key trends, correlations, and anomalies. Summarize insights in plain language for business stakeholders."
* 🔧 Use case: Developing the ability to communicate results effectively.

### 5. Model Interpretation & Explainability

* 🔍 Prompt template:  
  "Explain the predictions of a trained model on this dataset. Show feature importance, SHAP or LIME plots, and provide a plain-language summary of why the model makes these predictions."
* 🔧 Use case: Learning how to interpret complex models and explain them to non-technical audiences.

I can **also create a single master learning prompt with a structured output template** for data science, similar to what we did for data engineering and analytics, so you can use it for any dataset or task. Do you want me to do that?

##### You said:

yes

###### ChatGPT said:

Here’s your **master learning prompt for data science**, structured like the analytics and data engineering templates:

⚡ **Master Data Science Learning Prompt**  
\*"Act as my data science mentor. I’m a data analyst transitioning into data science. Given this dataset or task: [insert dataset or description], guide me through the following and present results in this exact structure:

1. **Statistical Analysis & Hypothesis Testing** → Perform key statistical tests, correlations, p-values, and confidence intervals. Provide Python code and summarize findings in plain language.
2. **Machine Learning Modeling** → Build 2–3 predictive models for the target variable. Compare performance metrics (accuracy, RMSE, etc.) and explain which model is best. Provide Python code.
3. **Feature Engineering & Selection** → Suggest and implement 5–10 new features, with explanations of why they improve the model. Provide Python code.
4. **Data Visualization & Storytelling** → Create visualizations to reveal trends, correlations, and anomalies. Summarize insights for business stakeholders. Provide Python code.
5. **Model Interpretation & Explainability** → Show feature importance, SHAP/LIME plots, and explain predictions in plain language."\*

**Output must follow this template:**

* 📊 Statistics → code block + plain-language summary
* 🤖 Modeling → code block + performance comparison + explanation
* 🧩 Features → code block + rationale
* 📈 Visualization → code block + key insights
* 🔍 Interpretation → code block + explanation of model predictions

R**eady-to-use structured output template for data science**, designed to pair with the master prompt so AI always produces consistent sections with code and explanations:

## 📊 1. Statistical Analysis & Hypothesis Testing

# Python code for statistical analysis

# Example: correlation, t-tests, ANOVA, confidence intervals

* 🔹 Key findings: [plain-language summary of results]

## 🤖 2. Machine Learning Modeling

# Python code to build predictive models

# Example: Linear Regression, Random Forest, XGBoost

# Include model training, evaluation, and metrics

* 🔹 Model comparison: [accuracy, RMSE, F1, etc.]
* 🔹 Best model explanation: [why this model performs best]

## 🧩 3. Feature Engineering & Selection

# Python code to create and select new features

* 🔹 Suggested features: [feature name → rationale]
* 🔹 Notes on feature importance: [optional insights]

## 📈 4. Data Visualization & Storytelling

# Python code for visualizations

# Example: matplotlib/seaborn plots, correlation heatmaps

* 🔹 Key insights: [5 concise bullet points for business stakeholders]

## 🔍 5. Model Interpretation & Explainability

# Python code for feature importance, SHAP, or LIME

* 🔹 Explanation of model predictions: [plain-language summary]
* 🔹 Key drivers/features affecting predictions

⚡ **Master Data Science Learning Prompt**  
\*"Act as my data science mentor. I’m a data analyst transitioning into data science. Given this dataset or task: [insert dataset or description], guide me through the following and present results in this exact structure:

1. **Statistical Analysis & Hypothesis Testing** → Perform key statistical tests, correlations, p-values, and confidence intervals. Provide Python code and summarize findings in plain language.
2. **Machine Learning Modeling** → Build 2–3 predictive models for the target variable. Compare performance metrics (accuracy, RMSE, etc.) and explain which model is best. Provide Python code.
3. **Feature Engineering & Selection** → Suggest and implement 5–10 new features, with explanations of why they improve the model. Provide Python code.
4. **Data Visualization & Storytelling** → Create visualizations to reveal trends, correlations, and anomalies. Summarize insights for business stakeholders. Provide Python code.
5. **Model Interpretation & Explainability** → Show feature importance, SHAP/LIME plots, and explain predictions in plain language."\*

**Output must follow this template:**

* 📊 Statistics → code block + plain-language summary
* 🤖 Modeling → code block + performance comparison + explanation
* 🧩 Features → code block + rationale
* 📈 Visualization → code block + key insights
* 🔍 Interpretation → code block + explanation of model predictions