Absolutely, Atif! Here's a comprehensive cheat sheet for Machine Learning that covers the full pipeline—from foundational concepts to deployment. It’s structured for clarity and depth, ideal for both quick reference and deeper study.

**🧠 Machine Learning Cheat Sheet**

**📚 1. Core Concepts**

* **Types of ML**
  + *Supervised Learning*: Labeled data (e.g., regression, classification)
  + *Unsupervised Learning*: Unlabeled data (e.g., clustering, dimensionality reduction)
  + *Semi-supervised*: Mix of labeled and unlabeled
  + *Reinforcement Learning*: Agent learns via rewards/punishments
* **Key Terminology**
  + *Feature*: Input variable
  + *Label*: Output variable (target)
  + *Model*: Mathematical representation of a process
  + *Training*: Learning from data
  + *Inference*: Making predictions
  + *Overfitting*: Model too closely fits training data
  + *Underfitting*: Model too simple to capture patterns

**🛠️ 2. Data Preprocessing**

* **Cleaning**
  + Handle missing values (mean, median, drop)
  + Remove duplicates
  + Fix inconsistent formats
* **Encoding**
  + Label Encoding
  + One-Hot Encoding
* **Scaling**
  + StandardScaler (mean=0, std=1)
  + MinMaxScaler (0 to 1)
* **Feature Engineering**
  + Polynomial features
  + Interaction terms
  + Domain-specific transformations
* **Dimensionality Reduction**
  + PCA (Principal Component Analysis)
  + t-SNE (for visualization)

**📊 3. Exploratory Data Analysis (EDA)**

* **Visualization Tools**
  + matplotlib, seaborn, plotly
* **Common Plots**
  + Histograms, Boxplots, Pairplots
  + Correlation heatmaps
* **Statistical Summary**
  + Mean, median, std, skewness, kurtosis

**🤖 4. Model Selection**

| **Task** | **Algorithms** |
| --- | --- |
| Regression | Linear Regression, SVR, XGBoost |
| Classification | Logistic Regression, Random Forest, SVM, KNN, XGBoost |
| Clustering | K-Means, DBSCAN, Hierarchical |
| Dim. Reduction | PCA, LDA |

**🧪 5. Model Evaluation**

* **Metrics**
  + *Regression*: MAE, MSE, RMSE, R²
  + *Classification*: Accuracy, Precision, Recall, F1, ROC-AUC
* **Validation Techniques**
  + Train/Test Split
  + K-Fold Cross Validation
  + Stratified Sampling

**🧮 6. Advanced Techniques**

* **Ensemble Methods**
  + Bagging (Random Forest)
  + Boosting (AdaBoost, Gradient Boosting, XGBoost, LightGBM)
  + Stacking
* **Hyperparameter Tuning**
  + Grid Search
  + Random Search
  + Bayesian Optimization
  + Tools: scikit-learn, Optuna, Ray Tune
* **Feature Selection**
  + Recursive Feature Elimination (RFE)
  + Lasso Regularization
  + Tree-based importance

**🧬 7. Deep Learning (DL)**

* **Frameworks**
  + TensorFlow, Keras, PyTorch
* **Components**
  + Neurons, Layers, Activation Functions
  + Loss Functions (MSE, Cross-Entropy)
  + Optimizers (SGD, Adam)
* **Architectures**
  + CNNs (images)
  + RNNs, LSTMs (sequences)
  + Transformers (NLP)

**🚀 8. Deployment**

* **Serialization**
  + pickle, joblib, ONNX
* **Serving**
  + REST APIs: FastAPI, Flask
  + Model Hosting: AWS SageMaker, Azure ML, GCP AI Platform
* **Monitoring**
  + Drift detection
  + Performance tracking
  + Retraining pipelines

**🧰 9. Tools & Libraries**

| **Category** | **Tools/Libraries** |
| --- | --- |
| Data Handling | pandas, numpy, cuDF |
| Modeling | scikit-learn, xgboost, lightgbm |
| Deep Learning | tensorflow, keras, pytorch |
| Visualization | matplotlib, seaborn, plotly |
| Deployment | flask, fastapi, mlflow |

**📈 10. Workflow Summary**

1. Define problem

2. Collect and clean data

3. Explore and visualize

4. Engineer features

5. Select model

6. Train and validate

7. Tune hyperparameters

8. Evaluate performance

9. Deploy and monitor

Would you like this cheat sheet tailored for Power BI integration, GPU-accelerated workflows, or procurement analytics? I can adapt it to your specific use case.