This plan (checklist) can guide Data scientists through their Machine Learning projects.

Frame the Problem and Look at the Big Picture

1. Define the business objectives.
2. Identify Use cases.
3. Identify the current solutions/workarounds (if any).
4. Identify possible ML solutions (supervised/unsupervised, online/offline, etc.)
5. Identify KPI (aligned with the business objective).
6. Identify the minimum performance needed to reach the business objective.
7. Search for comparable problems (to reuse experience or tools).
8. Assign resources (find available expertise).
9. Identify how to solve the problem manually.
10. List assumptions and risks.
11. Verify assumptions if possible.

Data Collection

Note: automate as much as possible so we can easily update data when needed.

1. List the data we need and how much we need.
2. Find and document where we can get that data.
3. Validate how much space it will take.
4. Check legal obligations and get access authorization if necessary.
5. Validate all access authorizations are working.
6. Create workspaces (with enough storage space).
7. Get the data.
8. Convert the data to a format we can easily manipulate (without changing the data itself).
9. Ensure sensitive information is deleted or protected (e.g., anonymized).
10. Validate the size and type of data (time series, sample, geographical, etc.).
11. Sample a test set and store it aside (to validate the model).

Data analysis (explore the data)

Note: SME are required to get insights for these steps.

1. Create a subset copy of the data for exploration.
2. Create a Jupyter notebook to keep a record of your data exploration.
3. Study each attribute and its characteristics: name, type, % of missing values, noisiness and type of noise, usefulness for the task, type of distribution (Gaussian, uniform, logarithmic, etc.)
4. For supervised learning tasks, identify the target attributes.
5. Analyze correlations between attributes.
6. Analyze how to solve the problem manually (if possible).
7. Identify extra data that would be useful.
8. Document what we have learned.

Data Preparation

Notes: We should always work on copies of the data (keep the original dataset intact) and write functions for all data transformations

1. Data cleaning: remove duplicates, correct errors, deal with missing values, normalization, data type conversions, etc.
2. Feature engineering, where appropriate:
   - Decompose features (e.g., categorical, date/time, etc.)
   - Enhance features (e.g., log(x), sqrt(x), x square, etc.)
   - Aggregate features into promising new features.
   - Standardize or normalize features.
3. Split into training and evaluation sets

Build and train the model

Notes: If the data is huge, we should sample smaller training sets so we can train many different models in a reasonable time. Automate these steps as much as possible.

1. Shortlist promising models.
2. Build the models.
3. Train different models.
4. Validate models:
   - Measure and compare their performance.
   - Analyze the most significant variables for each algorithm.
   - Analyze the types of errors the models make.
   - Perform a quick round of feature selection and engineering.
5. Perform few more iterations of the previous steps.
6. Shortlist the top three to five most promising models (prefer models that make different types of errors).

Hyperparameter Tuning and Training at scale

Notes: Use as much data as possible for this step, especially as you move toward the end of fine-tuning. Automate as much as possible.

1. Fine-tune the hyperparameters using cross-validation. Treat the data transformation choices as hyperparameters.
2. Combine the best models to produce better performance than running them individually.
3. Make prediction: Once we are confident about the final model, measure its performance on the test set to estimate the generalization error.
4. Train the final model at scale.

Deployment

1. Get the solution ready for production (plug into production data inputs, write unit tests, etc.).
2. Write monitoring code to check system's live performance at regular intervals and trigger alerts when it drops and beware of slow degradation (models tend to "rot" as data evolves).
3. Monitor inputs' quality.
4. Retrain models on a regular basis on fresh data (automate as much as possible).

Present the Solution

1. Complete the documentation of the model.
2. Create a presentation and make sure to highlight the big picture first.
3. Explain how the solution achieves the business objective.
4. Present interesting points noticed along the way (what worked and what did not, assumptions, system's limitations, etc.).
5. Ensure key findings are communicated through visualizations or easy-to-remember statements.