### Data Science Project To-Do List

#### 1. \*\*Project Setup\*\*

- [ ] Create a new project directory.

- [ ] Initialize a Git repository in the project directory.

- [ ] Set up a virtual environment for the project.

- [ ] Install necessary libraries (e.g., pandas, numpy, matplotlib, seaborn, scikit-learn).

#### 2. \*\*Data Loading\*\*

- [ ] Load the dataset from the CSV file.

- [ ] Display the first few rows of the dataset to understand its structure.

- [ ] Check for missing values and data types.

#### 3. \*\*Data Cleaning\*\*

- [ ] Handle missing values (e.g., imputation, removal).

- [ ] Convert columns to appropriate data types (e.g., dates to datetime).

- [ ] Remove or handle outliers if necessary.

- [ ] Add any derived columns (e.g., extracting the month from the date).

#### 4. \*\*Exploratory Data Analysis (EDA)\*\*

- [ ] Generate summary statistics for numerical columns.

- [ ] Plot histograms for numerical columns to understand their distributions.

- [ ] Plot box plots to identify outliers.

- [ ] Calculate and visualize the correlation matrix.

- [ ] Analyze time series trends (e.g., monthly sales trends).

- [ ] Plot relationships between features (e.g., scatter plots, pair plots).

#### 5. \*\*Feature Engineering\*\*

- [ ] Create new features that might be useful for predictive modeling (e.g., age of the house).

- [ ] Encode categorical variables if necessary.

- [ ] Scale numerical features if necessary.

#### 6. \*\*Model Building\*\*

- [ ] Split the data into training and test sets.

- [ ] Choose a baseline model (e.g., linear regression).

- [ ] Train the baseline model and evaluate its performance.

- [ ] Experiment with different models (e.g., decision trees, random forests, gradient boosting).

- [ ] Tune hyperparameters using cross-validation.

- [ ] Evaluate model performance using appropriate metrics (e.g., RMSE, R²).

#### 7. \*\*Model Evaluation\*\*

- [ ] Compare the performance of different models.

- [ ] Analyze residuals to check for patterns.

- [ ] Plot feature importance for tree-based models.

- [ ] Select the best model based on performance and interpretability.

#### 8. \*\*Model Deployment (Optional)\*\*

- [ ] Create a simple web app to interact with the model (e.g., using Flask or Streamlit).

- [ ] Deploy the app to a cloud service (e.g., Heroku).

#### 9. \*\*Documentation and Reporting\*\*

- [ ] Document the project setup and dependencies (e.g., in a README file).

- [ ] Provide detailed comments in the code.

- [ ] Create a Jupyter notebook or report summarizing your findings and model performance.

- [ ] Visualize key results and insights with plots and charts.

#### 10. \*\*Version Control and Sharing\*\*

- [ ] Make regular commits to the Git repository with meaningful commit messages.

- [ ] Push the project to a GitHub repository.

- [ ] Share your project on LinkedIn or a personal portfolio site.

### Additional Considerations

- [ ] Regularly review and refine your code for readability and efficiency.

- [ ] Seek feedback from peers or mentors to improve your analysis and modeling.

- [ ] Stay updated with best practices and new techniques in data science.