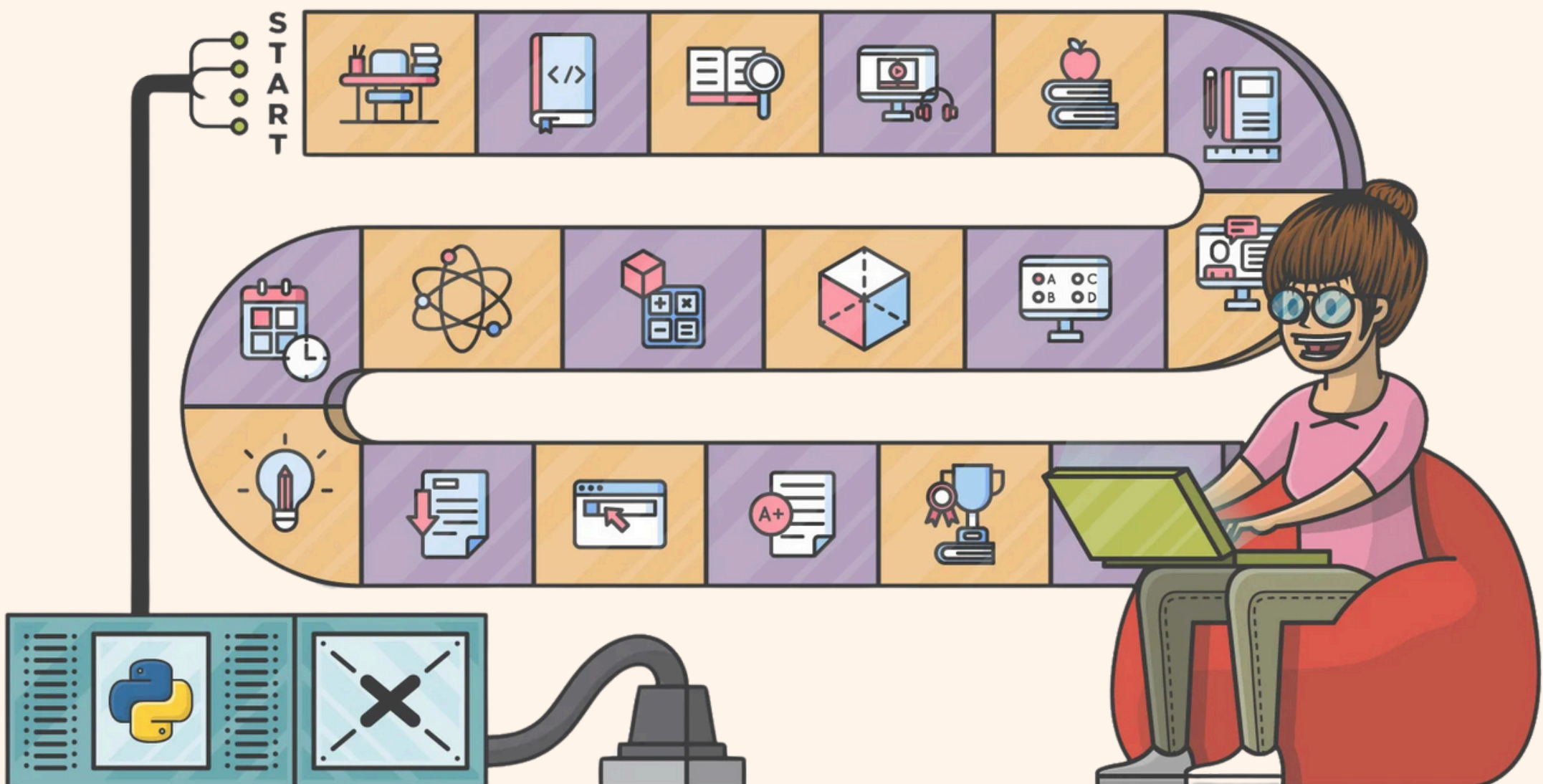


PYTHON CHEATSHEET FOR MACHINE LEARNING



Greg Coquillo
Product Leader





Greg Coquillo
Product Leader

1. LINEAR REGRESSION

Essential Python packages for data analysis and machine learning tasks:

```
import numpy as np          # Numerical operations
import pandas as pd         # Data manipulation
import matplotlib.pyplot as plt # Visualization
import seaborn as sns       # Statistical plots
import sklearn              # Machine learning toolkit
```

Install using:

```
pip install numpy pandas matplotlib seaborn
scikit-learn
```





Greg Coquillo
Product Leader

2. DATA HANDLING (PANDAS)

Load, inspect, and clean data efficiently using Pandas.

```
df = pd.read_csv("data.csv")  
df.head()           # Preview data  
df.info()           # Summary  
df.describe()       # Stats overview  
df.isnull().sum()   # Check missing values
```

Tip:

Use `dropna()` or `fillna()` to handle missing data.





Greg Coquillo
Product Leader

3. PREPROCESSING (SKLEARN)

Prepare your data for model training by scaling and encoding.

```
from sklearn.preprocessing import StandardScaler,  
LabelEncoder  
scaler = StandardScaler().fit_transform(df)  
le = LabelEncoder().fit_transform(df['category'])
```

Tip:

Split data for training and testing using
`train_test_split`.





Greg Coquillo
Product Leader

4. MODEL TRAINING

Train machine learning models to make predictions.

```
from sklearn.linear_model import LinearRegression
model = LinearRegression().fit(X_train, y_train)
pred = model.predict(X_test)
```

Popular models:

Decision Trees, Random Forests, SVM, KNN, Logistic Regression.





Greg Coquillo
Product Leader

5. EVALUATION METRICS

Measure model performance to ensure reliability.

```
from sklearn.metrics import accuracy_score, r2_score
accuracy_score(y_test, pred)
r2_score(y_test, pred)
```

Use:

Precision, Recall, F1-score, or RMSE depending on problem type.





Greg Coquillo
Product Leader

6. VISUALIZATION

Visualize trends and relationships in your data.

```
sns.heatmap(df.corr(), annot=True)  
plt.scatter(y_test, pred)  
plt.show()
```

Popular plots:

Heatmaps, Pairplots, Boxplots, Histograms.





7. FEATURE ENGINEERING

Create or transform features to improve model accuracy.

```
df['new_feature'] = df['col1'] * df['col2']
```

Techniques:

Scaling, Encoding, Dimensionality Reduction (PCA), Feature Selection.



8. CROSS VALIDATION & TUNING

Improve model generalization and optimize hyperparameters.

```
from sklearn.model_selection import GridSearchCV  
grid = GridSearchCV(model, params, cv=5)  
grid.fit(X_train, y_train)
```

Use:

`cross_val_score()` or `RandomizedSearchCV` for faster tuning.





9. DEEP LEARNING (QUICK START)

Build simple neural networks using TensorFlow or Keras.

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense
```

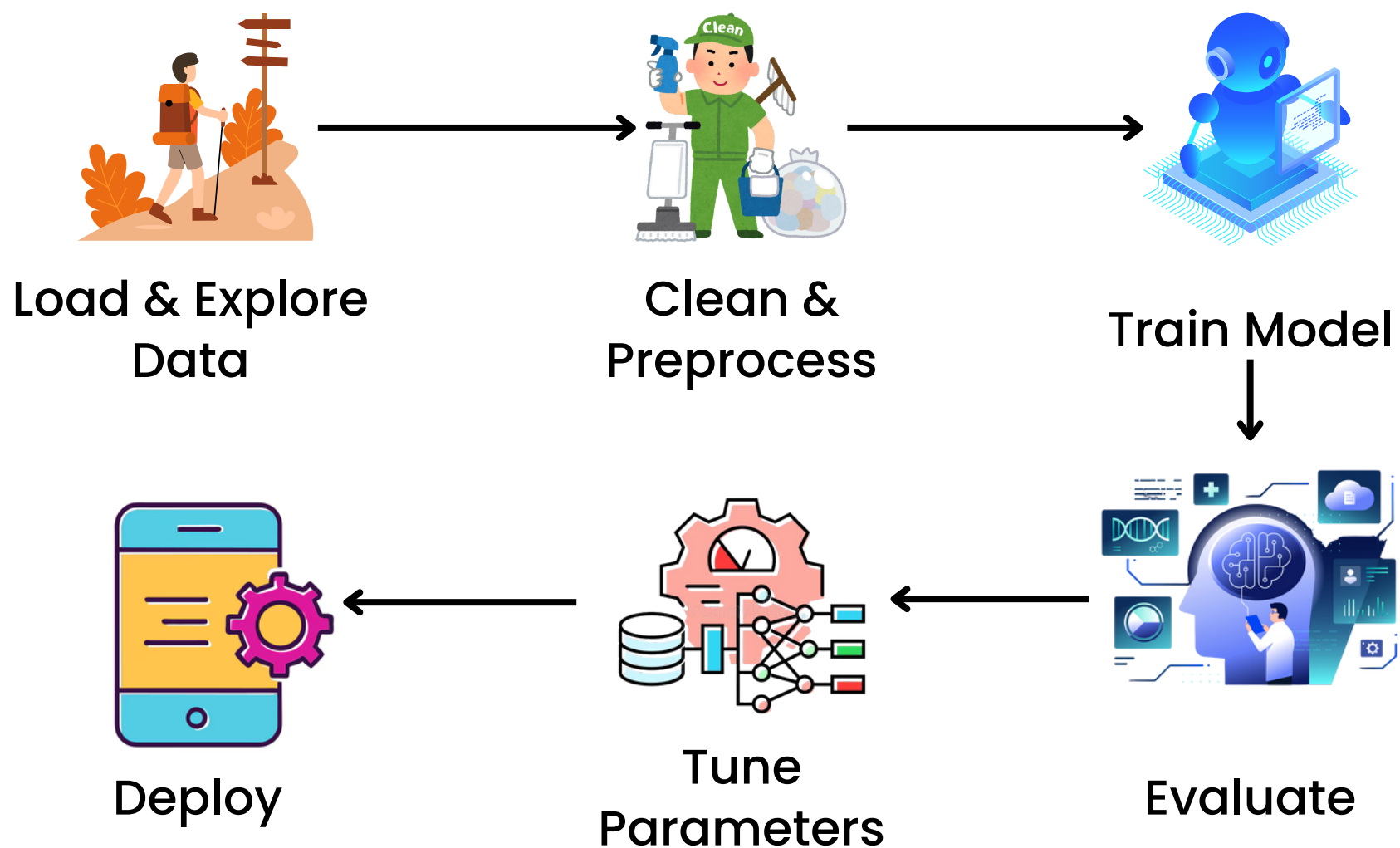
Use:

Use relu for hidden layers and sigmoid/softmax for output layers.



COMMON ML WORKFLOW

A step-by-step guide to every ML project:



Remember:

Data quality matters more than model complexity.





Greg Coquillo

Product Leader at AWS