

Python Cheat Sheet for Data Science

1. Libraries

```
import numpy as np      # Numerical computations
import pandas as pd     # Data manipulation and analysis
import matplotlib.pyplot as plt # Data visualization
import seaborn as sns   # Advanced data visualization
from sklearn.model_selection import train_test_split # Splitting datasets
from sklearn.metrics import accuracy_score # Model evaluation
from scipy.stats import ttest_ind # Statistical tests
```

2. NumPy

```
np.array([1, 2, 3])    # Create an array
np.mean(array)         # Calculate mean
np.std(array)          # Calculate standard deviation
np.sum(array)          # Sum of elements
np.dot(a, b)           # Dot product of two arrays
np.linspace(0, 1, 10)  # Generate evenly spaced numbers
np.random.rand(3, 3)   # Generate random numbers
```

3. Pandas

```
pd.DataFrame(data)     # Create a DataFrame
df.head()              # Display first 5 rows
df.info()              # Display summary of the DataFrame
df.describe()          # Statistics of numerical columns
df['column'].mean()     # Mean of a column
df.groupby('column').sum() # Group and aggregate
df.isnull().sum()      # Check for missing values
```

4. Matplotlib

```
plt.plot(x, y)         # Line plot
```

```
plt.scatter(x, y)      # Scatter plot
plt.hist(data)         # Histogram
plt.bar(x, height)     # Bar chart
plt.xlabel('X-axis')   # Label for X-axis
plt.ylabel('Y-axis')   # Label for Y-axis
plt.title('Title')     # Title of the plot
```

5. Seaborn

```
sns.scatterplot(x='col1', y='col2', data=df) # Scatter plot
sns.heatmap(data.corr(), annot=True)        # Heatmap with correlations
sns.boxplot(x='col1', y='col2', data=df)    # Boxplot
sns.histplot(data['column'])               # Histogram
sns.pairplot(data)                        # Pairplot of all numerical columns
```

6. Scikit-Learn

```
from sklearn.linear_model import LinearRegression # Import Linear Regression
model = LinearRegression()                       # Create model instance
model.fit(X_train, y_train)                      # Train the model
y_pred = model.predict(X_test)                  # Predict on test data
accuracy_score(y_test, y_pred)                 # Calculate accuracy
train_test_split(X, y, test_size=0.2)          # Split data into train/test
```

7. Data Preprocessing

```
from sklearn.preprocessing import StandardScaler # Import scaler
scaler = StandardScaler()                      # Create scaler instance
X_scaled = scaler.fit_transform(X)             # Scale features
pd.get_dummies(df, columns=['col'])            # One-hot encoding
df.fillna(value, inplace=True)                 # Fill missing values
df.dropna(inplace=True)                       # Drop rows with missing values
```

8. Advanced Topics

```
from sklearn.decomposition import PCA      # Principal Component Analysis
pca = PCA(n_components=2)                  # Create PCA instance
X_pca = pca.fit_transform(X)               # Reduce dimensions
from sklearn.cluster import KMeans         # Import KMeans
kmeans = KMeans(n_clusters=3)              # Create KMeans instance
kmeans.fit(X)                             # Fit KMeans model
```