Python Data Science Cheat Sheet

1. Setup and Libraries

- **Install Libraries**

```
bash

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

- **Import Libraries**

```
python

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, accuracy_score
```

2. Data Manipulation with Pandas

- **Read Data**

```
python

df = pd.read_csv('data.csv')
```

- **Explore Data**

```
python

df.head()
df.info()
df.describe()
```

- **Data Cleaning**

```
python

df.dropna(inplace=True)
df.fillna(value, inplace=True)
df.drop(columns=['col1', 'col2'], inplace=True)
```

- **Data Selection**

```
python

df['column_name']
df[['col1', 'col2']]
df.iloc[0]  # Select row by index
df.loc[df['column'] > 0]  # Conditional selection
```

- **Data Aggregation**

```
python

df.groupby('column').mean()
df.groupby('column').agg({'col1': 'mean', 'col2': 'sum'})
df.pivot_table(values='value', index='index', columns='column',
aggfunc='mean')
```

3. Data Analysis and Visualization

- **Basic Plots**

```
python

df['column'].hist()
df.plot(kind='bar')
df.plot(kind='scatter', x='col1', y='col2')
```

- **Matplotlib**

```
python

plt.figure(figsize=(10, 6))
plt.plot(df['column'])
plt.xlabel('X-axis label')
plt.ylabel('Y-axis label')
plt.title('Title of the plot')
plt.show()
```

- **Seaborn**

```
python

sns.histplot(df['column'])
sns.boxplot(x='column', data=df)
sns.scatterplot(x='col1', y='col2', data=df)
sns.heatmap(df.corr(), annot=True)
```

4. Machine Learning with Scikit-Learn

- **Split Data**

```
python

X = df.drop('target', axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

- **Preprocessing**

```
python

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

- **Model Training**

```
python

model = LinearRegression()
model.fit(X_train_scaled, y_train)
```

- **Model Prediction**

```
python

predictions = model.predict(X_test_scaled)
```

- **Model Evaluation**

```
python

mse = mean_squared_error(y_test, predictions)
accuracy = accuracy_score(y_test, predictions)
print(f'Mean Squared Error: {mse}')
print(f'Accuracy: {accuracy}')
```

- **Common Models**

```
python

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC

# Example: Random Forest
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
rf_predictions = rf.predict(X_test)
```

5. Advanced Topics

- **Pipelines**

```
python

from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.svm import SVC

pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('pca', PCA(n_components=2)),
    ('svm', SVC())
])

pipeline.fit(X_train, y_train)
pipeline_predictions = pipeline.predict(X_test)
```

- **Cross-Validation**

```
python

from sklearn.model_selection import cross_val_score

scores = cross_val_score(model, X, y, cv=5)
print(f'Cross-validation scores: {scores}')
```

- **Hyperparameter Tuning**

```
python

from sklearn.model_selection import GridSearchCV

param_grid = {'n_estimators': [100, 200], 'max_depth': [10, 20]}
grid_search = GridSearchCV(RandomForestClassifier(), param_grid, cv=5)
grid_search.fit(X_train, y_train)
print(f'Best parameters: {grid_search.best_params_}')
```

6. Saving and Loading Models

- **Saving a Model**

```
python

import joblib

joblib.dump(model, 'model.pkl')
```

- **Loading a Model**

```
python

loaded_model = joblib.load('model.pkl')
```

This cheat sheet provides a concise reference for common tasks in Python data science using popular libraries like NumPy, Pandas, Matplotlib, Seaborn, and Scikit-Learn.