**Python for Data Science Cheat Sheet**

### Introduction

SciKit-Learn: "sklearn" is a machine learning library for the Python programming language. Simple and efficient tool for data mining, Data analysis and Machine Learning.

Importing Convention - import sklearn

### Data Loading

- Using Numpy:
  ```python
  >>> import numpy as np
  >>> a = np.array([[1,2,3,4],[7,8,9,10]], dtype=int)
  >>> data = np.loadtxt('file_name.csv', delimiter=',')
  ```

- Using Pandas:
  ```python
  >>> import pandas as pd
  >>> df = pd.read_csv('file_name.csv', header=0)
  ```

### Train-Test Data

```python
>>> from sklearn.model_selection import train_test_split
>>> X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

### Data Preparation

- Standardization:
  ```python
  >>> from sklearn.preprocessing import StandardScaler
  >>> get_names = df.columns
  >>> X = preprocessing.StandardScaler().fit_transform(df)
  >>> standardized_X = preprocessing.normalize(X)
  ```

- Normalization:
  ```python
  >>> from sklearn.preprocessing import Normalizer
  >>> X = preprocessing.normalize(X)
  >>> X = pd.read_csv('file_name.csv')
  >>> X = X.values
  >>> normalized_X = preprocessing.normalize(X)
  ```

### Preprocessing

**Working On Model**

#### Model Choosing

**Supervised Learning Estimator:**
- **Linear Regression:** `from sklearn.linear_model import LinearRegression`
- **Naive Bayes:** `from sklearn.naive_bayes import GaussianNB`
- **KNN:** `from sklearn import neighbors`
- **Support Vector Machine:** `from sklearn.svm import SVC`

**Unsupervised Learning Estimator:**
- **Principal Component Analysis (PCA):** `from sklearn.decomposition import PCA`
- **KMeans:** `from sklearn.cluster import KMeans`

**Post-Processing**

**Evaluation Performance**

**Classification:**
1. Confusion Matrix:
   ```python
   >>> from sklearn.metrics import confusion_matrix
   >>> confusion_matrix(y_test, y_pred)
   ```
2. Accuracy Score:
   ```python
   >>> from sklearn.metrics import accuracy_score
   >>> accuracy_score(y_test, y_pred)
   ```

**Regression:**
1. Mean Absolute Error:
   ```python
   >>> from sklearn.metrics import mean_absolute_error
   ```
2. Mean Squared Error:
   ```python
   >>> from sklearn.metrics import mean_squared_error
   ```
3. R² Score:
   ```python
   >>> from sklearn.metrics import r2_score
   ```

**Clustering:**
```python
>>> from sklearn import cluster
>>> kmeans = cluster.KMeans(n_clusters=5)
>>> kmeans.fit(X_train)
>>> kmeans.labels_.shape
```