Cheat Sheet for comprehensive edX Data Science MicroMasters by UC San Diego

Data Science Fundamentals

Key Concepts

- **Data Types**: Numeric (int, float), Categorical (nominal, ordinal), Text, Date/Time
- **Data Structures**: Arrays, DataFrames, Series
- **Data Cleaning**: Handling missing values, outliers, duplicates
- **Exploratory Data Analysis (EDA)**: Summary statistics, visualizations, correlation analysis

Python Libraries

- **Pandas**: Data manipulation and analysis

- `import pandas as pd`
- `df = pd.read_csv('file.csv')`
- `df.head()`, `df.describe()`, `df.info()`

- **NumPy**: Numerical operations

- `import numpy as np`
- `array = np.array([1, 2, 3])`
- `np.mean(array)`, `np.std(array)`

- **Matplotlib/Seaborn**: Data visualization

- `import matplotlib.pyplot as plt`
- `plt.plot(x, y)`, `plt.scatter(x, y)`
- `import seaborn as sns`
- `sns.heatmap(data.corr())`

Machine Learning

Supervised Learning

- **Regression**: Predicting continuous values

- Linear Regression: `from sklearn.linear_model import LinearRegression`
- Decision Trees: `from sklearn.tree import DecisionTreeRegressor`

- **Classification**: Predicting categorical values

- Logistic Regression: `from sklearn.linear_model import LogisticRegression`
- Random Forest: `from sklearn.ensemble import RandomForestClassifier`

Unsupervised Learning

- **Clustering**: Grouping similar data points

- K-Means: `from sklearn.cluster import KMeans`
- Hierarchical Clustering: `from scipy.cluster.hierarchy import linkage, dendrogram`

- **Dimensionality Reduction**: Reducing the number of features

- PCA: `from sklearn.decomposition import PCA`
- t-SNE: `from sklearn.manifold import TSNE`

Model Evaluation

- **Metrics**: Accuracy, Precision, Recall, F1-Score, ROC-AUC

- `from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score`

- **Cross-Validation**: `from sklearn.model_selection import cross_val_score`

- **Confusion Matrix**: `from sklearn.metrics import confusion_matrix`

Data Visualization

Matplotlib

- **Basic Plots**: Line, Scatter, Bar, Histogram

- `plt.plot(x, y)`, `plt.scatter(x, y)`, `plt.bar(x, y)`, `plt.hist(data)`

- **Customization**: Labels, Titles, Legends

- `plt.xlabel('X-axis')`, `plt.ylabel('Y-axis')`, `plt.title('Title')`, `plt.legend()`

Seaborn

- **Advanced Plots**: Heatmaps, Pairplots, Boxplots

- `sns.heatmap(data.corr())`, `sns.pairplot(data)`, `sns.boxplot(x='feature', y='target', data=data)`

- **Styling**: `sns.set_style('darkgrid')`, `sns.set_palette('colorblind')`

Data Wrangling

Pandas Operations

- **Filtering**: `df[df['column'] > value]`

- **Grouping**: `df.groupby('column').mean()`
- **Merging**: `pd.merge(df1, df2, on='key')`
- **Pivoting**: `df.pivot_table(values='value', index='index', columns='columns')`

Handling Missing Data

- **Dropping**: `df.dropna()`
- **Filling**: `df.fillna(method='ffill')`, `df.fillna(value=mean_value)`

Statistical Analysis

Descriptive Statistics

- **Mean**: `df['column'].mean()`
- **Median**: `df['column'].median()`
- **Mode**: `df['column'].mode()`
- **Standard Deviation**: `df['column'].std()`

Inferential Statistics

- **Hypothesis Testing**: t-test, chi-square test
  - `from scipy.stats import ttest_ind`, `from scipy.stats import chi2_contingency`
- **Confidence Intervals**: `import statsmodels.stats.api as sms`
  - `sms.DescrStatsW(data).tconfint_mean()`

Big Data and Scalability

Hadoop and Spark

- **Hadoop**: Distributed storage and processing
  - HDFS: Hadoop Distributed File System
  - MapReduce: Data processing framework
- **Spark**: In-memory distributed computing
  - `from pyspark import SparkContext`
  - `sc = SparkContext("local", "AppName")`
  - `data = sc.textFile("file.txt")`

Scaling Techniques

- **Batch Processing**: Processing data in chunks

- **Parallel Processing**: Utilizing multiple cores/machines

- **Cloud Computing**: AWS, Google Cloud, Azure

Ethics and Best Practices

Data Privacy

- **Anonymization**: Removing personally identifiable information

- **Consent**: Ensuring data subjects agree to data usage

Bias and Fairness

- **Bias Detection**: Identifying and mitigating biases in data and models

- **Fairness Metrics**: Equal opportunity, disparate impact

Version Control

- **Git**: Tracking changes in code

- `git init`, `git add .`, `git commit -m "message"`
- `git push`, `git pull`, `git clone`

Additional Resources

Documentation

- **Pandas**: pandas.pydata.org

- **Scikit-Learn**: scikit-learn.org

- **Matplotlib**: matplotlib.org

- **Seaborn**: seaborn.pydata.org

Online Courses

- **edX**: [edx.org](https://www.edx.org)

- **Coursera**: [coursera.org](https://www.coursera.org)

Communities

- **Stack Overflow**: stackoverflow.com

- **Kaggle**: [kaggle.com](https://www.kaggle.com)

Tips and Tricks

Efficient Coding

- **Vectorization**: Use NumPy for faster operations

- **Caching**: Store intermediate results to avoid recomputation
- **Parallelization**: Use libraries like Dask for parallel computing

Debugging

- **Print Statements**: Use `print()` for quick checks
- **Logging**: Use `logging` module for detailed logs
- **Debugger**: Use `pdb` or IDE debuggers for step-by-step debugging

Performance Optimization

- **Profiling**: Use `cProfile` to identify bottlenecks
- **Memory Management**: Use `gc` module for garbage collection
- **Efficient Algorithms**: Choose algorithms with lower time complexity

This cheat sheet provides a comprehensive overview of the essential concepts, tools, and techniques covered in the edX Data Science MicroMasters by UC San Diego. Use it as a quick reference to navigate through the course material effectively.

By Ahmed Baheeg Khorshid

ver 1.0