Cheat Sheet for comprehensive Udacity Data Scientist Nanodegree

Data Collection

Web Scraping

- **Libraries**: BeautifulSoup, Scrapy, Selenium

- **BeautifulSoup**:

- `soup = BeautifulSoup(html_content, 'html.parser')`
- `soup.find('tag_name', {'class': 'class_name'})`
- `soup.find_all('tag_name')`

- **Scrapy**:

- `scrapy startproject project_name`
- `scrapy crawl spider_name`

- **Selenium**:

- `driver = webdriver.Chrome(executable_path='/path/to/chromedriver')`
- `driver.get('https://example.com')`
- `element = driver.find_element_by_xpath('//xpath')`

APIs

- **Requests Library**:

- `response = requests.get('https://api.example.com/data')`
- `response.json()`

- **Authentication**:

- `headers = {'Authorization': 'Bearer YOUR_TOKEN'}`
- `response = requests.get(url, headers=headers)`

Data Wrangling

Pandas

- **DataFrames**:

- `df = pd.DataFrame(data)`
- `df.head()`
- `df.info()`
- `df.describe()`

- **Data Cleaning**:

- `df.dropna()`
- `df.fillna(value)`
- `df.replace(old, new)`

- **Data Transformation**:

- `df['new_column'] = df['column'] * 2`
- `df.apply(lambda x: x + 1)`
- `df.groupby('column').agg({'another_column': 'mean'})`

NumPy

- **Arrays**:

- `arr = np.array([1, 2, 3])`
- `arr.shape`
- `arr.dtype`

- **Operations**:

- `np.mean(arr)`
- `np.sum(arr)`
- `np.max(arr)`

- **Broadcasting**:

- `arr + 1`
- `arr * 2`

Exploratory Data Analysis (EDA)

Visualization

- **Matplotlib**:

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

- **Seaborn**:

- `sns.pairplot(df)`
- `sns.heatmap(df.corr(), annot=True)`
- `sns.boxplot(x='column', y='another_column', data=df)`

Statistical Analysis

- **Descriptive Statistics**:

- `df.mean()`
- `df.median()`
- `df.mode()`

- **Hypothesis Testing**:

- `scipy.stats.ttest_ind(sample1, sample2)`
- `scipy.stats.chi2_contingency(contingency_table)`

Modeling

Supervised Learning

- **Linear Regression**:

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

- **Decision Trees**:

- `from sklearn.tree import DecisionTreeClassifier`
- `model = DecisionTreeClassifier()`
- `model.fit(X_train, y_train)`
- `model.predict(X_test)`

Unsupervised Learning

- **K-Means Clustering**:

- `from sklearn.cluster import KMeans`
- `model = KMeans(n_clusters=3)`
- `model.fit(X)`
- `model.predict(X)`

- **PCA**:

- `from sklearn.decomposition import PCA`
- `pca = PCA(n_components=2)`
- `X_pca = pca.fit_transform(X)`

Evaluation

Metrics

- **Classification**:

- `accuracy_score(y_true, y_pred)`

- `confusion_matrix(y_true, y_pred)`
- `classification_report(y_true, y_pred)`

- **Regression**:

- `mean_squared_error(y_true, y_pred)`
- `r2_score(y_true, y_pred)`

Cross-Validation

- **K-Fold**:

- `from sklearn.model_selection import KFold`
- `kf = KFold(n_splits=5)`
- `for train_index, test_index in kf.split(X):`
- `X_train, X_test = X[train_index], X[test_index]`
- `y_train, y_test = y[train_index], y[test_index]`

Deployment

Flask

- **Basic Setup**:

- `from flask import Flask`
- `app = Flask(__name__)`
- `@app.route('/')`
- `def home():`
- `return "Hello, World!"`

- **Running the App**:

- `if __name__ == '__main__':`
- `app.run(debug=True)`

Docker

- **Dockerfile**:

- `FROM python:3.8-slim`
- `COPY . /app`
- `WORKDIR /app`
- `RUN pip install -r requirements.txt`
- `CMD ["python", "app.py"]`

- **Building and Running**:

- `docker build -t myapp .`
- `docker run -p 5000:5000 myapp`

Tips and Tricks

Jupyter Notebook

- **Magic Commands**:

- `%matplotlib inline`
- `%timeit function()`
- `%%writefile file.py`

- **Shortcuts**:

- `Shift + Enter`: Run cell
- `Ctrl + Enter`: Run cell in place
- `Esc + M`: Markdown mode
- `Esc + Y`: Code mode

Version Control

- **Git Commands**:

- `git init`
- `git add .`
- `git commit -m "message"`
- `git push origin main`
- `git pull origin main`

Debugging

- **Print Statements**:

- `print(variable)`

- **Logging**:

- `import logging`
- `logging.basicConfig(level=logging.DEBUG)`
- `logging.debug('Debug message')`

Resources

Documentation

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

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

- **Matplotlib**: matplotlib.org

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

Community

- **Stack Overflow**: stackoverflow.com
- **Kaggle**: [kaggle.com](https://www.kaggle.com)
- **GitHub**: github.com

Example Workflow

1. **Data Collection**:

- Scrape data from a website using BeautifulSoup.
- Fetch data from an API using the Requests library.

2. **Data Wrangling**:

- Clean the data using Pandas.
- Transform the data as needed.

3. **Exploratory Data Analysis**:

- Visualize data using Matplotlib and Seaborn.
- Perform statistical analysis.

4. **Modeling**:

- Train a Linear Regression model.
- Evaluate the model using cross-validation.

5. **Deployment**:

- Deploy the model using Flask.
- Containerize the application using Docker.

This cheat sheet provides a comprehensive overview of the key concepts, tools, and techniques covered in the Udacity Data Scientist Nanodegree. Use it as a quick reference to navigate through the program efficiently.

By Ahmed Baheeg Khorshid

ver 1.0