Cheat Sheet for comprehensive Stanford University Data Science Certificate

Data Science Fundamentals

Data Types

- **Numerical**: Continuous (e.g., height) and Discrete (e.g., number of children)
- **Categorical**: Nominal (e.g., colors) and Ordinal (e.g., education levels)
- **Text**: Free-form text data
- **Time-Series**: Data indexed by time

Data Structures

- **Arrays**: Homogeneous collections (e.g., NumPy arrays)
- **DataFrames**: Heterogeneous tabular data (e.g., Pandas DataFrame)
- **Series**: One-dimensional labeled array (e.g., Pandas Series)

Data Collection and Cleaning

Data Collection

- **APIs**: Use libraries like `requests` to fetch data
- **Web Scraping**: Use `BeautifulSoup` and `Scrapy`
- **Databases**: Use `SQLAlchemy` or `psycopg2` for PostgreSQL

Data Cleaning

- **Handling Missing Values**:
  - Drop rows/columns: `df.dropna()`
  - Fill with mean/median: `df.fillna(df.mean())`
- **Removing Duplicates**: `df.drop_duplicates()`
- **Data Transformation**:
  - Normalization: `(df - df.min()) / (df.max() - df.min())`
  - Standardization: `(df - df.mean()) / df.std()`

Exploratory Data Analysis (EDA)

Summary Statistics

- **Mean**: `df.mean()`
- **Median**: `df.median()`
- **Mode**: `df.mode()`
- **Variance**: `df.var()`
- **Standard Deviation**: `df.std()`

Visualization

- **Histograms**: `df.hist()`
- **Box Plots**: `df.boxplot()`
- **Scatter Plots**: `df.plot.scatter(x='col1', y='col2')`
- **Pair Plots**: `sns.pairplot(df)`

Statistical Methods

Hypothesis Testing

- **T-Test**: `scipy.stats.ttest_ind(a, b)`
- **Chi-Square Test**: `scipy.stats.chi2_contingency(observed)`
- **ANOVA**: `scipy.stats.f_oneway(a, b, c)`

Probability Distributions

- **Normal Distribution**: `scipy.stats.norm.pdf(x)`
- **Binomial Distribution**: `scipy.stats.binom.pmf(k, n, p)`
- **Poisson Distribution**: `scipy.stats.poisson.pmf(k, mu)`

Machine Learning

Supervised Learning

- **Linear Regression**: `LinearRegression()`
- **Logistic Regression**: `LogisticRegression()`
- **Decision Trees**: `DecisionTreeClassifier()`
- **Random Forests**: `RandomForestClassifier()`

- **Support Vector Machines**: `SVC()`

Unsupervised Learning

- **K-Means Clustering**: `KMeans(n_clusters=k)`
- **Hierarchical Clustering**: `AgglomerativeClustering()`
- **Principal Component Analysis (PCA)**: `PCA(n_components=n)`

Model Evaluation

- **Cross-Validation**: `cross_val_score(model, X, y, cv=k)`
- **Confusion Matrix**: `confusion_matrix(y_true, y_pred)`
- **ROC Curve**: `roc_curve(y_true, y_pred)`

Big Data and Scalability

Distributed Computing

- **Apache Spark**: Use `pyspark` for large-scale data processing
- **Hadoop**: Use `HDFS` for distributed storage

Cloud Computing

- **AWS**: Use `S3` for storage, `EMR` for big data processing
- **Google Cloud**: Use `BigQuery` for data warehousing
- **Azure**: Use `Azure Databricks` for Spark-based analytics

Data Visualization

Libraries

- **Matplotlib**: Basic plotting library
- **Seaborn**: Statistical data visualization
- **Plotly**: Interactive plots

Best Practices

- **Clarity**: Ensure labels, titles, and legends are clear
- **Aesthetics**: Use color schemes that are easy on the eyes
- **Interactivity**: Use interactive plots for deeper insights

Ethics and Fairness in Data Science

Bias and Fairness

- **Data Bias**: Ensure data collection methods are unbiased
- **Algorithmic Fairness**: Use fairness metrics like `Equal Opportunity` and `Demographic Parity`

Privacy

- **Data Anonymization**: Use techniques like `k-anonymity` and `differential privacy`
- **Consent**: Ensure data subjects have given informed consent

Tools and Libraries

Python Libraries

- **Pandas**: Data manipulation and analysis
- **NumPy**: Numerical computing
- **Scikit-Learn**: Machine learning
- **TensorFlow/Keras**: Deep learning

R Libraries

- **dplyr**: Data manipulation
- **ggplot2**: Data visualization
- **caret**: Machine learning

Practical Tips

Version Control

- **Git**: Use `git init`, `git add`, `git commit`, `git push`
- **GitHub**: Host repositories and collaborate

Documentation

- **Jupyter Notebooks**: Use for interactive coding and documentation
- **Sphinx**: Generate documentation from docstrings

Collaboration

- **Slack/Teams**: For real-time communication
- **Jira/Trello**: For project management

Example Workflow

1. **Data Collection**: Fetch data from API
2. **Data Cleaning**: Handle missing values and duplicates
3. **EDA**: Generate summary statistics and visualizations
4. **Modeling**: Train a machine learning model
5. **Evaluation**: Use cross-validation and confusion matrix
6. **Deployment**: Deploy model using cloud services

Conclusion

This cheat sheet provides a comprehensive overview of the essential concepts, tools, and techniques required for the Stanford University Data Science Certificate. Use it as a quick reference to navigate through the complexities of data science.

By Ahmed Baheeg Khorshid

ver 1.0