Data Science Cheatsheet

A comprehensive cheat sheet covering essential concepts, tools, and techniques in Data Science. It provides a quick reference for machine learning algorithms, data manipulation, statistical methods, and more.

Fundamentals

Key Concepts

Supervised Learning	Learning from labeled data to predict outcomes. Examples: Regression, Classification.
Unsupervised Learning	Discovering patterns in unlabeled data. Examples: Clustering, Dimensionality Reduction.
Reinforcement Learning	Training an agent to make decisions in an environment to maximize a reward.
Bias-Variance Tradeoff	Balancing model complexity to minimize both bias (underfitting) and variance (overfitting).
Cross-Validation	Evaluating model performance on multiple subsets of the data to ensure generalization.
Feature Engineering	Creating new features or transforming existing ones to improve model accuracy.

Common Algorithms

- **Linear Regression:** Predicts a continuous outcome using a linear equation.
- **Logistic Regression:** Predicts a binary outcome using a logistic function.
- **Decision Trees:** Partitions data into subsets based on feature values to make predictions.
- **Random Forest:** An ensemble of decision trees that averages predictions to improve accuracy.
- **Support Vector Machines (SVM):** Finds the optimal hyperplane to separate data into classes.
- **K-Nearest Neighbors (KNN):** Classifies data based on the majority class among its k nearest neighbors.
- **K-Means Clustering:** Partitions data into k clusters based on distance to cluster centroids.

Python for Data Science

Data Manipulation with Pandas

Creating a DataFrame

```
import pandas as pd
data = {'col1': [1, 2], 'col2': [3, 4]}
df = pd.DataFrame(data)
```

Selecting Columns

```
df['col1']
df[['col1', 'col2']]
```

Filtering Rows

```
df[df['col1'] > 1]
```

Grouping and Aggregation

```
df.groupby('col1').mean()
```

Handling Missing Data

```
df.dropna()
df.fillna(0)
```

Data Visualization with Matplotlib and Seaborn

Basic Plotting with Matplotlib

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.show()
```

Scatter Plot with Seaborn

```
import seaborn as sns
sns.scatterplot(x='col1', y='col2', data=df)
plt.show()
```

Histogram with Seaborn

```
sns.histplot(df['col1'])
plt.show()
```

Box Plot with Seaborn

```
sns.boxplot(x='col1', y='col2', data=df)
plt.show()
```

Scikit-learn for Machine Learning

Training a Model

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

Making Predictions

```
y_pred = model.predict(X_test)
```

Model Evaluation

```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test, y_pred)
print(mse)
```

Data Preprocessing

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Train-Test Split

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Statistical Methods

Descriptive Statistics

Mean	Average value of a dataset. Formula: $ \frac{\sum_{i=1}^{n} x_i}{n} $
Median	Middle value of a sorted dataset.
Mode	Most frequent value in a dataset.
Standard Deviation	Measure of the spread of data around the mean. Formula: $ \sqrt{\frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n}} $
Variance	Square of the standard deviation. Formula: $ \frac{\sum_{i=1}^{n} (x_i - \mu)^2}{n} $

Inferential Statistics

Hypothesis Testing	A method for testing a claim or hypothesis about a population parameter.
P-value	Probability of obtaining results as extreme as the observed results, assuming the null hypothesis is true.
Confidence Interval	Range of values likely to contain the true population parameter with a certain level of confidence.
T-test	Used to compare the means of two groups.
ANOVA	Used to compare the means of more than two groups.

Model Evaluation and Tuning

Evaluation Metrics

Accuracy	Fraction of correctly classified instances. Formula: $ \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} $
Precision	Fraction of true positives among predicted positives. Formula: $ \frac{\text{True Positives}}{\text{True Positives + False Positives}} $
Recall	Fraction of true positives among actual positives. Formula: $ \frac{\text{True Positives}}{\text{True Positives + False Negatives}} $
F1-Score	Harmonic mean of precision and recall. Formula: $ 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision + Recall}} $
AUC-ROC	Area under the Receiver Operating Characteristic curve, measures the ability of a classifier to distinguish between classes.
Mean Squared Error (MSE)	Average squared difference between predicted and actual values. Formula: $ \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 $
R-squared	Proportion of variance in the dependent variable that can be predicted from the independent variables. Formula: $ 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}i)^2}{\sum{i=1}^{n} (y_i - \bar{y})^2} $

Hyperparameter Tuning

Grid Search: Exhaustively search a specified subset of the hyperparameters of a learning algorithm.

Randomized Search: Sample a given number of candidates from a hyperparameter search space.

Bayesian Optimization: Uses Bayesian inference to find the hyperparameters that optimize a given metric.

Cross-Validation: Evaluate model performance on multiple subsets of the data to ensure generalization.