Cheat Sheet for comprehensive CIW Data Analyst

Data Collection and Preparation

Data Sources

- **Structured Data**: Databases, CSV files, Excel spreadsheets
- **Unstructured Data**: Text files, PDFs, social media posts
- **Semi-Structured Data**: JSON, XML, NoSQL databases

Data Cleaning

- **Missing Values**:
  - **Remove Rows**: `df.dropna()`
  - **Fill with Mean/Median**: `df.fillna(df.mean())`
  - **Forward/Backward Fill**: `df.ffill()`, `df.bfill()`
- **Duplicates**:
  - **Identify**: `df.duplicated()`
  - **Remove**: `df.drop_duplicates()`
- **Outliers**:
  - **Detect**: Z-score, IQR
  - **Handle**: Remove, Cap, Transform

Data Transformation

- **Normalization**:
  - **Min-Max Scaling**: `(X - X.min()) / (X.max() - X.min())`
  - **Z-Score**: `(X - X.mean()) / X.std()`
- **Encoding**:
  - **One-Hot Encoding**: `pd.get_dummies()`
  - **Label Encoding**: `LabelEncoder()`
- **Binning**:

- **Equal Width**: `pd.cut()`
- **Equal Frequency**: `pd.qcut()`

Data Analysis

Descriptive Statistics

- **Central Tendency**:
  - **Mean**: `df.mean()`
  - **Median**: `df.median()`
  - **Mode**: `df.mode()`
- **Dispersion**:
  - **Range**: `df.max() - df.min()`
  - **Variance**: `df.var()`
  - **Standard Deviation**: `df.std()`
- **Correlation**:
  - **Pearson**: `df.corr(method='pearson')`
  - **Spearman**: `df.corr(method='spearman')`

Inferential Statistics

- **Hypothesis Testing**:
  - **T-Test**: `scipy.stats.ttest_ind()`
  - **Chi-Square**: `scipy.stats.chi2_contingency()`
  - **ANOVA**: `scipy.stats.f_oneway()`
- **Confidence Intervals**:
  - **Mean CI**: `scipy.stats.t.interval()`
  - **Proportion CI**: `statsmodels.stats.proportion.proportion_confint()`

Data Visualization

Basic Plots

- **Line Plot**: `plt.plot(x, y)`

- **Bar Plot**: `plt.bar(x, y)`
- **Histogram**: `plt.hist(data, bins=n)`
- **Scatter Plot**: `plt.scatter(x, y)`

Advanced Plots

- **Box Plot**: `plt.boxplot(data)`
- **Heatmap**: `sns.heatmap(data)`
- **Pair Plot**: `sns.pairplot(df)`
- **Violin Plot**: `sns.violinplot(x, y)`

Data Modeling

Supervised Learning

- **Regression**:
  - **Linear Regression**: `LinearRegression()`
  - **Polynomial Regression**: `PolynomialFeatures()`
- **Classification**:
  - **Logistic Regression**: `LogisticRegression()`
  - **Decision Trees**: `DecisionTreeClassifier()`
  - **Random Forest**: `RandomForestClassifier()`

Unsupervised Learning

- **Clustering**:
  - **K-Means**: `KMeans()`
  - **Hierarchical**: `AgglomerativeClustering()`
- **Dimensionality Reduction**:
  - **PCA**: `PCA()`
  - **t-SNE**: `TSNE()`

Model Evaluation

Metrics

- **Regression**:
  - **R-Squared**: `r2_score()`
  - **Mean Squared Error**: `mean_squared_error()`
- **Classification**:
  - **Accuracy**: `accuracy_score()`
  - **Precision/Recall**: `precision_score()`, `recall_score()`
  - **F1-Score**: `f1_score()`
- **Clustering**:
  - **Silhouette Score**: `silhouette_score()`
  - **Davies-Bouldin Index**: `davies_bouldin_score()`

Cross-Validation

- **K-Fold**: `KFold()`
- **Stratified K-Fold**: `StratifiedKFold()`
- **Leave-One-Out**: `LeaveOneOut()`

Tools and Libraries

Python Libraries

- **Pandas**: Data manipulation and analysis
  - **DataFrame**: `pd.DataFrame()`
  - **Read CSV**: `pd.read_csv()`
- **NumPy**: Numerical operations
  - **Array**: `np.array()`
  - **Random**: `np.random.rand()`
- **Matplotlib**: Plotting
  - **Figure**: `plt.figure()`

  - **Subplots**: `plt.subplots()`
- **Seaborn**: Statistical data visualization
  - **Heatmap**: `sns.heatmap()`
  - **Pairplot**: `sns.pairplot()`
- **Scikit-Learn**: Machine learning
  - **Model**: `model.fit()`, `model.predict()`
  - **Metrics**: `accuracy_score()`, `mean_squared_error()`

SQL Basics

- **SELECT**: `SELECT column FROM table`
- **WHERE**: `SELECT * FROM table WHERE condition`
- **JOIN**: `SELECT * FROM table1 JOIN table2 ON table1.id = table2.id`
- **GROUP BY**: `SELECT column, COUNT(*) FROM table GROUP BY column`
- **ORDER BY**: `SELECT * FROM table ORDER BY column ASC/DESC`

Tips and Tricks

Data Exploration

- **Quick Summary**: `df.describe()`
- **Unique Values**: `df.nunique()`
- **Value Counts**: `df.value_counts()`

Performance Optimization

- **Vectorization**: Use NumPy for array operations
- **Parallel Processing**: `joblib.Parallel()`
- **Caching**: `functools.lru_cache()`

Debugging

- **Print Statements**: Use `print()` for quick checks
- **Logging**: `import logging` for detailed logs
- **Profiling**: `cProfile` for performance analysis

Examples

Data Cleaning Example

```
import pandas as pd

# Load data
df = pd.read_csv('data.csv')

# Handle missing values
df.fillna(df.mean(), inplace=True)

# Remove duplicates
df.drop_duplicates(inplace=True)
```

Visualization Example

```
import matplotlib.pyplot as plt
import seaborn as sns

# Scatter plot
plt.scatter(df['X'], df['Y'])
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Scatter Plot')
plt.show()

# Heatmap
sns.heatmap(df.corr(), annot=True)
plt.show()
```

Model Training Example

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2)

# Train model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)

# Evaluate
```

```
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')
```

This cheat sheet provides a comprehensive overview of essential concepts, tools, and techniques for a CIW Data Analyst. Each section is designed to be concise yet thorough, ensuring that users can quickly reference and apply the information as needed.

By Ahmed Baheeg Khorshid

ver 1.0