UnCodemy

Data Science Cheat Sheet PDF

A data science cheat sheet gives fast reminders: common pandas calls to load and reshape data, core statistics formulas, the scikit-learn fit and predict pattern, evaluation metrics and a table for choosing a model. It helps revision but should be used after you understand the concepts.

Key takeaways

- Keep one sheet for each of pandas, statistics, modelling and metrics.
- The scikit-learn pattern is fit on training data, then predict and score on test data.
- Match the metric to the cost of errors, not to habit.
- Cheat sheets support memory; they do not replace understanding.
- Check library documentation for current function names.

What is data science cheat sheet?

A data science cheat sheet is a compact reference that lists frequently used commands, formulas and decision rules so a learner can recall them quickly while coding or revising.

Which pandas commands should I remember?

```
df = pd.read_csv("file.csv")        load a table
df.head(); df.info(); df.describe() first look
df.isna().sum()                     missing values per column
df.dropna(); df.fillna(value)       handle missing values
df[df["age"] > 30]                  filter rows
df.groupby("city")["sales"].sum()   aggregate
df.merge(other, on="id", how="left")  join tables
df.sort_values("sales", ascending=False)
pd.get_dummies(df["city"])          one-hot encoding
```

Which statistics formulas matter?

Idea	Meaning
Mean	Sum of values divided by count
Variance	Average squared distance from the mean
Standard deviation	Square root of variance

Idea	Meaning
Z-score	(value minus mean) divided by standard deviation
Correlation	Strength and direction of a linear link, from -1 to 1
Standard error	Standard deviation divided by square root of n
Remember that a p-value is the probability of seeing a result at least as extreme if the null hypothesis were true, not the probability that the hypothesis is true.	

What is the scikit-learn pattern?

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
model = SomeModel()
model.fit(X_train, y_train)
pred = model.predict(X_test)
score = metric(y_test, pred)
```

Wrap scaling and encoding in a **Pipeline** so the same steps apply to training and test data and leakage is avoided.

How do I choose a model quickly?

Situation	Try first
Predict a number, linear pattern	Linear regression
Predict a number, complex pattern	Random forest or gradient boosting
Predict yes or no, need explanation	Logistic regression
Mixed features, interactions	Decision tree or forest
No labels, find groups	K-means
Too many features	PCA

Which metrics fit which problem?

For balanced classes accuracy is a start; for rare events prefer **precision, recall and F1**. When missing a positive case is costly, favour recall; when false alarms are costly, favour precision. For regression use **MAE** for average error size and **RMSE** when large errors matter more. For ranking problems consider ROC-AUC.

What checks prevent common mistakes?

Check the target is not in the features, split before scaling, set a random seed, compare to a baseline, look at residuals or the confusion matrix and test on data from a later period if the data has time order.

How should I use a cheat sheet well?

Write your own version from memory first, then compare it with a reference and fix gaps. Retyping commands builds recall better than reading them.

How do I read a confusion matrix quickly?

A confusion matrix for two classes has four cells. True positives are cases correctly flagged, false positives are normal cases wrongly flagged, false negatives are real cases that were missed and true negatives are normal cases correctly left alone. Precision is true positives divided by all flagged cases, and recall is true positives divided by all real cases. The F1 score combines the two. When you present results, always state which cell matters most to the business, because a fraud model and a spam filter pay very different prices for each error.

What are the quick rules for cleaning and preparing data?

Check types and missing values first, then remove duplicates, fix labels, handle outliers with a reason, encode categories, scale numeric columns when the algorithm uses distances and split the data before any fitting step. Keep every transformation in code so that it can be repeated on new data. Record assumptions in comments, such as why a value was capped. These small habits turn a notebook into reliable work, and they are easy to recall when written on a single sheet beside your keyboard.

When should I move beyond the cheat sheet?

Once you can write each snippet from memory, begin solving small problems without looking at it. If you still need to check a command, mark it and practise only that item for a few days. The aim is to rely on understanding, so that the cheat sheet becomes a rarely used safety net instead of a daily crutch, and so that you can explain every line to an interviewer.

What else belongs on a one-page revision sheet?

Add the steps of your standard workflow, a list of three libraries per task, the metrics you use most and two warnings from your own past errors. A short list of reminders such as "split first, scale second" and "check the baseline" often prevents the mistakes that cost the most time. Update the sheet after every project with one new lesson learned.

Frequently asked questions

What goes on a data science cheat sheet?

Include common pandas commands, key statistics formulas, the scikit-learn fit and predict pattern, evaluation metrics and a table for choosing models.

Can a cheat sheet replace learning?

No. It only jogs memory for concepts you already understand, so use it for quick revision and while coding, not as your first source.

Which metric should I use for imbalanced data?

Use precision, recall and F1 rather than accuracy alone, because accuracy can look high when a model ignores the rare class.

Why use a Pipeline in scikit-learn?

A Pipeline applies the same preprocessing to training and test data, which reduces leakage and keeps the workflow repeatable and tidy.

How often should I update my cheat sheet?

Review it whenever you learn something new or when library versions change, and check official documentation for current function names.

Sources and further reading

- Python documentation
- pandas documentation
- scikit-learn user guide

About this PDF

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to info@uncodemy.com and the trainer team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550