PANDAS

Category	Function/Method	Description (Simple)	Example
Reading Data	pd.read_csv()	Read CSV file	pd.read_csv('file.csv')
	pd.read_excel()	Read Excel file	pd.read_excel('file.xlsx')
	pd.read_json()	Read JSON file	pd.read_json('file.json')
	pd.read_html()	Read HTML tables	pd.read_html('file.html')
	pd.read_sql()	Read SQL query	pd.read_sql('SELECT * FROM table', conn)
Writing Data	df.to_csv()	Write to CSV	df.to_csv('file.csv')
	df.to_excel()	Write to Excel	df.to_excel('file.xlsx')
	df.to_json()	Write to JSON	df.to_json('file.json')
Viewing Data	df.head()	First rows	df.head(5)
	df.tail()	Last rows	df.tail(5)
	df.sample()	Random sample	df.sample(3)
Info & Summary	df.info()	Data summary	df.info()
	df.describe()	Statistics summary	df.describe()
	df.shape	Rows and columns	df.shape
	df.columns	Column names	df.columns
	df.index	Index info	df.index
	df.dtypes	Data types	df.dtypes
	df.memory_usage()	Memory usage	df.memory_usage()
Selecting Data	df['col']	Select column	df['Name']
	df[['col1','col2']]	Select multiple columns	df[['Name','Age']]
	df.loc[]	Select by label	df.loc[0:5, 'Name']
	df.iloc[]	Select by index	df.iloc[0:5, 1:3]
	df.at[]	Single value by label	df.at[0, 'Name']
	df.iat[]	Single value by index	df.iat[0,1]
Filtering Data	df[df['col'] > value]	Filter rows	df[df['Age']>30]
Missing Data	df.isnull()	Check missing	df.isnull()
	df.isnull().sum()	Count missing	df.isnull().sum()
	df.notnull()	Not missing	df.notnull()
	df.dropna()	Drop missing rows	df.dropna()
	df.fillna()	Fill missing values	df.fillna(0)
Duplicates	df.duplicated()	Check duplicates	df.duplicated()
	df.drop_duplicates()	Remove duplicates	df.drop_duplicates()
Sorting	df.sort_values()	Sort by column	df.sort_values('Age')
	df.sort_index()	Sort by index	df.sort_index()
Grouping & Aggregation	df.groupby()	Group data	df.groupby('Dept').mean()
	df.agg()	Multiple aggregates	df.agg({'Age':'mean','Salary':'sum'})
	df.pivot_table()	Pivot table	df.pivot_table(values='Sales', index='Region', columns='Month')
Merging & Joining	pd.concat()	Combine dfs	pd.concat([df1,df2])
	pd.merge()	Merge dfs	pd.merge(df1, df2, on='ID')
	df.join()	Join on index	df1.join(df2)
Encoding & Mapping	pd.get_dummies()	One-hot encode	pd.get_dummies(df['Gender'])
	df.map()	Map values	df['Grade'].map({'A':4,'B':3})
	df.apply()	Apply function	df['Score'].apply(np.sqrt)
	df.applymap()	Apply function to all	df.applymap(str)
Replacing Values	df.replace()	Replace values	df.replace('old','new')
Indexing	df.set_index()	Set index	df.set_index('ID')
	df.reset_index()	Reset index	df.reset_index()
Insertion & Deletion	df.insert()	Insert column	df.insert(2, 'New', value)
	df.drop()	Drop column/row	df.drop('Col', axis=1)
	df.pop()	Remove & return column	df.pop('Col')
Unique & Counts	df['col'].unique()	Unique values	df['City'].unique()
	df['col'].nunique()	Count unique	df['City'].nunique()
	df['col'].value_counts()	Value counts	df['City'].value_counts()

NUMPY

Category	Function/Method	Description (Simple)	Example
Creation	np.array()	Create array	np.array([1,2,3])
	np.zeros()	Array of zeros	np.zeros((2,3))
	np.ones()	Array of ones	np.ones((3,3))
	np.full()	Filled array	np.full((2,2),7)
	np.eye()	Identity matrix	np.eye(3)
	np.arange()	Range array	np.arange(0,10,2)
	np.linspace()	Evenly spaced values	np.linspace(0,1,5)
	np.random.rand()	Random [0,1)	np.random.rand(2,3)
	np.random.randn()	Random normal	np.random.randn(3,3)

	np.random.randint()	Random integers	np.random.randint(1,10,5)
Properties	arr.shape	Shape of array	arr.shape
	arr.size	Number of elements	arr.size
	arr.ndim	Dimensions	arr.ndim
	arr.dtype	Data type	arr.dtype
Reshaping	arr.reshape()	Reshape array	arr.reshape(3,2)
	np.expand_dims()	Add dimension	np.expand_dims(arr,0)
	np.squeeze()	Remove dimensions	np.squeeze(arr)
Operations	np.mean()	Mean	np.mean(arr)
	np.median()	Median	np.median(arr)
	np.std()	Std deviation	np.std(arr)
	np.var()	Variance	np.var(arr)
	np.sum()	Sum	np.sum(arr)
	np.min()	Minimum	np.min(arr)
	np.max()	Maximum	np.max(arr)
	np.argmin()	Index of min	np.argmin(arr)
	np.argmax()	Index of max	np.argmax(arr)
Math	np.dot()	Dot product	np.dot(a,b)
	np.matmul()	Matrix multiplication	np.matmul(a,b)
	np.transpose()	Transpose	np.transpose(arr)
Manipulation	np.concatenate()	Concatenate arrays	np.concatenate([a,b])
	np.vstack()	Stack vertically	np.vstack((a,b))
	np.hstack()	Stack horizontally	np.hstack((a,b))
	np.split()	Split array	np.split(arr,2)
Other	np.unique()	Unique elements	np.unique(arr)
	np.sort()	Sort array	np.sort(arr)
	arr.astype()	Change type	arr.astype(float)

sklearn

Category	Function/Method	Description (Simple)	Example
Data Split	train_test_split	Split data for train & test	X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2)
Preprocessing	StandardScaler()	Standardize features	scaler=StandardScaler(); X_scaled=scaler.fit_transform(X)
	MinMaxScaler()	Scale between min-max	scaler=MinMaxScaler(); X_scaled=scaler.fit_transform(X)
	LabelEncoder()	Encode labels	le=LabelEncoder(); y=le.fit_transform(y)
	OneHotEncoder()	One-hot encode	ohe=OneHotEncoder(); X_enc=ohe.fit_transform(X)
	PolynomialFeatures()	Generate polynomial features	poly=PolynomialFeatures(2); X_poly=poly.fit_transform(X)
Linear Models	LinearRegression()	Linear regression	model=LinearRegression(); model.fit(X,y)
	LogisticRegression()	Logistic regression	model=LogisticRegression(); model.fit(X,y)
Tree Models	DecisionTreeClassifier()	Decision tree (classification)	model=DecisionTreeClassifier(); model.fit(X,y)
	DecisionTreeRegressor()	Decision tree (regression)	model=DecisionTreeRegressor(); model.fit(X,y)
	RandomForestClassifier()	Random forest (classification)	model=RandomForestClassifier(); model.fit(X,y)
	RandomForestRegressor()	Random forest (regression)	model=RandomForestRegressor(); model.fit(X,y)
SVM	SVC()	Support Vector Classifier	model=SVC(); model.fit(X,y)
	SVR()	Support Vector Regressor	model=SVR(); model.fit(X,y)
Clustering	KMeans()	K-means clustering	kmeans=KMeans(n_clusters=3); kmeans.fit(X)
Model Evaluation	accuracy_score()	Accuracy	accuracy_score(y_test,y_pred)
	confusion_matrix()	Confusion matrix	confusion_matrix(y_test,y_pred)
	classification_report()	Precision, recall, f1	classification_report(y_test,y_pred)
	mean_squared_error()	MSE	mean_squared_error(y_test,y_pred)
	r2_score()	R2 score	r2_score(y_test,y_pred)
Cross Validation	cross_val_score()	Cross-validation score	cross_val_score(model,X,y,cv=5)
Pipelines	Pipeline()	Create pipeline	pipe=Pipeline([('scaler',StandardScaler()),('model',LogisticRegression())]); pipe.fit(X,y)

TensorFlow

Category	Function/Method	Description (Simple)	Example
Basics	tf.constant()	Constant tensor	tf.constant([1,2,3])
	tf.Variable()	Variable tensor	tf.Variable([1,2,3])
	tf.matmul()	Matrix multiplication	tf.matmul(a,b)
	tf.reduce_mean()	Mean value	tf.reduce_mean(tensor)
	tf.reduce_sum()	Sum value	tf.reduce_sum(tensor)
Model Building (Keras)	tf.keras.models.Sequential()	Sequential model	model=tf.keras.models.Sequential()
	tf.keras.layers.Dense()	Dense (fully connected) layer	model.add(tf.keras.layers.Dense(64,activation='relu'))
	tf.keras.layers.Conv2D()	Convolution layer	model.add(tf.keras.layers.Conv2D(32,(3,3)))
	tf.keras.layers.MaxPooling2D()	Max pooling layer	model.add(tf.keras.layers.MaxPooling2D(2,2))
	tf.keras.layers.Flatten()	Flatten layer	model.add(tf.keras.layers.Flatten())
	model.compile()	Compile model	model.compile(optimizer='adam',loss='mse')
	model.fit()	Train model	model.fit(X_train,y_train,epochs=10)
	model.evaluate()	Evaluate model	model.evaluate(X_test,y_test)
	model.predict()	Make predictions	model.predict(X_test)

PyTorch

Category	Function/Method	Description (Simple)	Example
Basics	torch.tensor()	Create tensor	torch.tensor([1,2,3])
	torch.zeros()	Zeros tensor	torch.zeros(2,3)
	torch.ones()	Ones tensor	torch.ones(3,3)
	torch.rand()	Random tensor	torch.rand(2,2)
	torch.matmul()	Matrix multiplication	torch.matmul(a,b)
	torch.mean()	Mean value	torch.mean(tensor.float())
	torch.sum()	Sum value	torch.sum(tensor)
Model Building (nn)	torch.nn.Linear()	Fully connected layer	layer = torch.nn.Linear(10,5)
	torch.nn.ReLU()	ReLU activation	act = torch.nn.ReLU()
	torch.nn.Sequential()	Sequential model	model = torch.nn.Sequential(layer1,act,layer2)
Optimizer & Loss	torch.optim.SGD()	Stochastic Gradient Descent	optim = torch.optim.SGD(model.parameters(),lr=0.01)
	torch.optim.Adam()	Adam optimizer	optim = torch.optim.Adam(model.parameters(),lr=0.001)
	torch.nn.CrossEntropyLoss()	Cross entropy loss	loss_fn = torch.nn.CrossEntropyLoss()
Training	loss.backward()	Backpropagation	loss.backward()
	optimizer.step()	Update weights	optim.step()
	optimizer.zero_grad()	Reset gradients	optim.zero_grad()

Matplotlib

Function/Method	Description (Simple)	Example
plt.plot()	Line plot	plt.plot(x,y)
plt.scatter()	Scatter plot	plt.scatter(x,y)
plt.bar()	Bar plot	plt.bar(x,y)
plt.hist()	Histogram	plt.hist(data)
plt.boxplot()	Box plot	plt.boxplot(data)
plt.xlabel()	X-axis label	plt.xlabel('X Label')
plt.ylabel()	Y-axis label	plt.ylabel('Y Label')
plt.title()	Plot title	plt.title('My Plot')
plt.legend()	Show legend	plt.legend()
plt.show()	Show plot	plt.show()

Seaborn

Function/Method	Description (Simple)	Example
sns.lineplot()	Line plot	sns.lineplot(x,y)
sns.scatterplot()	Scatter plot	sns.scatterplot(x,y)
sns.barplot()	Bar plot with stats	sns.barplot(x,y)
sns.histplot()	Histogram	sns.histplot(data)
sns.boxplot()	Box plot	sns.boxplot(x,y)
sns.heatmap()	Heatmap matrix	sns.heatmap(data)
sns.pairplot()	Pairwise plots	sns.pairplot(df)
sns.countplot()	Count plot	sns.countplot(x='col',data=df)