Here's a curated list of the **top 30 Python libraries for Data Science productivity**, along with explanations and code examples:

---

1. NumPy

- **Purpose**: Numerical computing; provides support for large, multi-dimensional arrays and matrices.
- **Example**:
- `import numpy as np`
- `arr = np.array([1, 2, 3])`
- `print(arr.mean())`

---

2. Pandas

- **Purpose**: Data manipulation and analysis.
- **Example**:
- `import pandas as pd`
- `data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}`
- `df = pd.DataFrame(data)`
- `print(df.head())`

---

3. Matplotlib

- **Purpose**: Data visualization; create static, animated, and interactive plots.
- **Example**:
- `import matplotlib.pyplot as plt`
- `plt.plot([1, 2, 3], [4, 5, 6])`
- `plt.show()`

---

4. Seaborn

- **Purpose**: Statistical data visualization; built on top of Matplotlib.
- **Example**:
- `import seaborn as sns`
- `sns.boxplot(x=[1, 2, 3, 4, 5])`

---

5. SciPy

- **Purpose**: Advanced scientific computations like optimization, integration, and more.
- **Example**:

```
from scipy.optimize import minimize
f = lambda x: (x - 3)**2
result = minimize(f, x0=0)
print(result.x)
```

---

6. Scikit-learn

- **Purpose**: Machine learning; includes algorithms for classification, regression, and clustering.
- **Example**:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit([[1], [2], [3]], [1, 2, 3])
print(model.coef_)
```

---

7. TensorFlow

- **Purpose**: Deep learning and neural networks; developed by Google.
- **Example**:

```
import tensorflow as tf
print(tf.constant("Hello TensorFlow!"))
```

---

8. PyTorch

- **Purpose**: Deep learning framework; developed by Facebook.
- **Example**:

```
import torch
x = torch.tensor([1.0, 2.0, 3.0])
print(x.mean())
```

---

9. Keras

- **Purpose**: High-level deep learning API running on TensorFlow.
- **Example**:

```
from keras.models import Sequential
from keras.layers import Dense
model = Sequential([Dense(10, activation='relu')])
```

---

10. Statsmodels

- **Purpose**: Statistical modeling and hypothesis testing.
- **Example**:

- `import statsmodels.api as sm`
- `data = sm.datasets.get_rdataset("mtcars").data`
- `print(data.head())`

---

11. Plotly

- **Purpose**: Interactive, web-based visualizations.
- **Example**:
- `import plotly.express as px`
- `fig = px.scatter(x=[1, 2, 3], y=[4, 5, 6])`
- `fig.show()`

---

12. Bokeh

- **Purpose**: Interactive visualizations for web applications.
- **Example**:
- `from bokeh.plotting import figure, show`
- `p = figure()`
- `p.line([1, 2, 3], [4, 5, 6])`
- `show(p)`

---

13. Dask

- **Purpose**: Parallel computing for larger-than-memory datasets.
- **Example**:
- `import dask.dataframe as dd`
- `df = dd.read_csv('large_dataset.csv')`
- `print(df.head())`

---

14. NLTK

- **Purpose**: Natural Language Processing.
- **Example**:
- `import nltk`
- `nltk.download('punkt')`
- `print(nltk.word_tokenize("Hello, world!"))`

---

15. SpaCy

- **Purpose**: NLP library optimized for production.
- **Example**:
- `import spacy`

- `nlp = spacy.load("en_core_web_sm")`
- `doc = nlp("Hello, world!")`
- `print([token.text for token in doc])`

---

16. Gensim

- **Purpose**: Topic modeling and document similarity.
- **Example**:
- `from gensim.models import Word2Vec`
- `model = Word2Vec([["hello", "world"]])`

---

17. LightGBM

- **Purpose**: Fast gradient boosting for machine learning.
- **Example**:
- `import lightgbm as lgb`
- `print(lgb.__version__)`

---

18. XGBoost

- **Purpose**: Gradient boosting framework for speed and performance.
- **Example**:
- `import xgboost as xgb`
- `print(xgb.__version__)`

---

19. CatBoost

- **Purpose**: Gradient boosting optimized for categorical features.
- **Example**:
- `from catboost import CatBoostClassifier`
- `model = CatBoostClassifier(iterations=100)`

---

20. PyCaret

- **Purpose**: Low-code machine learning library.
- **Example**:
- `from pycaret.datasets import get_data`
- `print(get_data("iris"))`

---

21. OpenCV

- **Purpose**: Image processing and computer vision.
- **Example**:
- `import cv2`
- `img = cv2.imread('image.jpg')`
- `cv2.imshow('Image', img)`

---

22. PIL (Pillow)

- **Purpose**: Image processing.
- **Example**:
- `from PIL import Image`
- `img = Image.open('image.jpg')`
- `img.show()`

---

23. Shap

- **Purpose**: Model explainability.
- **Example**:
- `import shap`
- `print(shap.__version__)`

---

24. LIME

- **Purpose**: Explain machine learning models.
- **Example**:
- `from lime.lime_tabular import LimeTabularExplainer`
- `print("LIME loaded")`

---

25. Altair

- **Purpose**: Declarative statistical visualization.
- **Example**:
- `import altair as alt`
- `alt.Chart({'x': [1, 2, 3], 'y': [4, 5, 6]})`

---

26. PyOD

- **Purpose**: Outlier detection.
- **Example**:
- `from pyod.models.knn import KNN`
- `model = KNN()`

---

27. NetworkX

- **Purpose**: Network and graph analysis.
- **Example**:
- `import networkx as nx`
- `G = nx.Graph()`
- `G.add_edge(1, 2)`
- `print(G.nodes)`

---

28. TensorFlow Probability

- **Purpose**: Probabilistic reasoning and statistical analysis.
- **Example**:
- `import tensorflow_probability as tfp`
- `print(tfp.__version__)`

---

29. Auto-sklearn

- **Purpose**: Automated machine learning (AutoML).
- **Example**:
- `import autosklearn.classification`
- `print("Auto-sklearn loaded")`

---

30. Theano

- **Purpose**: Mathematical computations for deep learning.
- **Example**:
- `import theano`
- `print(theano.__version__)`
