```
!pip install mlflow
```

```
Collecting mlflow
  Downloading mlflow-3.3.1-py3-none-any.whl.metadata (30 kB)
Collecting mlflow-skinny==3.3.1 (from mlflow)
  Downloading mlflow_skinny-3.3.1-py3-none-any.whl.metadata (31 kB)
Collecting mlflow-tracing==3.3.1 (from mlflow)
  Downloading mlflow_tracing-3.3.1-py3-none-any.whl.metadata (19 kB)
Requirement already satisfied: Flask<4 in /usr/local/lib/python3.12/dist-packages (from mlflow) (3.1.1)
Collecting alembic!=1.10.0,<2 (from mlflow)
  Downloading alembic-1.16.4-py3-none-any.whl.metadata (7.3 kB)
Requirement already satisfied: cryptography<46,>=43.0.0 in /usr/local/lib/python3.12/dist-packages (from mlflow) (43.0.3)
Collecting docker<8,>=4.0.0 (from mlflow)
  Downloading docker-7.1.0-py3-none-any.whl.metadata (3.8 kB)
Collecting graphene<4 (from mlflow)
  Downloading graphene-3.4.3-py2.py3-none-any.whl.metadata (6.9 kB)
Collecting gunicorn<24 (from mlflow)
  Downloading gunicorn-23.0.0-py3-none-any.whl.metadata (4.4 kB)
Requirement already satisfied: matplotlib<4 in /usr/local/lib/python3.12/dist-packages (from mlflow) (3.10.0)
Requirement already satisfied: numpy<3 in /usr/local/lib/python3.12/dist-packages (from mlflow) (2.0.2)
Requirement already satisfied: pandas<3 in /usr/local/lib/python3.12/dist-packages (from mlflow) (2.2.2)
Requirement already satisfied: pyarrow<22,>=4.0.0 in /usr/local/lib/python3.12/dist-packages (from mlflow) (18.1.0)
Requirement already satisfied: scikit-learn<2 in /usr/local/lib/python3.12/dist-packages (from mlflow) (1.6.1)
Requirement already satisfied: scipy<2 in /usr/local/lib/python3.12/dist-packages (from mlflow) (1.16.1)
Requirement already satisfied: sqlalchemy<3,>=1.4.0 in /usr/local/lib/python3.12/dist-packages (from mlflow) (2.0.43)
Requirement already satisfied: cachetools<7,>=5.0.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->ml
Requirement already satisfied: click<9,>=7.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlflow) (
Requirement already satisfied: cloudpickle<4 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlflow) (
Collecting databricks-sdk<1,>=0.20.0 (from mlflow-skinny==3.3.1->mlflow)
  Downloading databricks_sdk-0.64.0-py3-none-any.whl.metadata (39 kB)
Requirement already satisfied: fastapi<1 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlflow) (0.11
Requirement already satisfied: gitpython<4,>=3.1.9 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlf
Requirement already satisfied: importlib_metadata!=4.7.0,<9,>=3.7.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-s
Requirement already satisfied: opentelemetry-api<3,>=1.9.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.
Requirement already satisfied: opentelemetry-sdk<3,>=1.9.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.
Requirement already satisfied: packaging<26 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlflow) (2
Requirement already satisfied: protobuf<7,>=3.12.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlf
Requirement already satisfied: pydantic<3,>=1.10.8 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlf
Requirement already satisfied: pyyaml<7,>=5.1 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlflow)
Requirement already satisfied: requests<3,>=2.17.3 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlf
Requirement already satisfied: sqlparse<1,>=0.4.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlfl
Requirement already satisfied: typing-extensions<5,>=4.0.0 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.
Requirement already satisfied: uvicorn<1 in /usr/local/lib/python3.12/dist-packages (from mlflow-skinny==3.3.1->mlflow) (0.35
Requirement already satisfied: Mako in /usr/lib/python3/dist-packages (from alembic!=1.10.0,<2->mlflow) (1.1.3)
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.12/dist-packages (from cryptography<46,>=43.0.0->mlflow)
Requirement already satisfied: urllib3>=1.26.0 in /usr/local/lib/python3.12/dist-packages (from docker<8,>=4.0.0->mlflow) (2.
Requirement already satisfied: blinker>=1.9.0 in /usr/local/lib/python3.12/dist-packages (from Flask<4->mlflow) (1.9.0)
Requirement already satisfied: itsdangerous>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from Flask<4->mlflow) (2.2.0)
Requirement already satisfied: jinja2>=3.1.2 in /usr/local/lib/python3.12/dist-packages (from Flask<4->mlflow) (3.1.6)
Requirement already satisfied: markupsafe>=2.1.1 in /usr/local/lib/python3.12/dist-packages (from Flask<4->mlflow) (3.0.2)
Requirement already satisfied: werkzeug>=3.1.0 in /usr/local/lib/python3.12/dist-packages (from Flask<4->mlflow) (3.1.3)
Collecting graphql-core<3.3,>=3.1 (from graphene<4->mlflow)
  Downloading graphql_core-3.2.6-py3-none-any.whl.metadata (11 kB)
Collecting graphql-relay<3.3,>=3.1 (from graphene<4->mlflow)
  Downloading graphql_relay-3.2.0-py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: python-dateutil<3,>=2.7.0 in /usr/local/lib/python3.12/dist-packages (from graphene<4->mlflow)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib<4->mlflow) (1.3.3
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib<4->mlflow) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib<4->mlflow) (4.59
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib<4->mlflow) (1.4
```

```python
import numpy as np
import pandas as pd
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report
import mlflow
import mlflow.sklearn

data = load_breast_cancer()
X = data.data
y = data.target

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

def evaluate_knn(k):
    with mlflow.start_run(run_name=f"KNN_k={k}"):
        model = KNeighborsClassifier(n_neighbors=k)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        acc = accuracy_score(y_test, y_pred)
        mlflow.log_param("n_neighbors", k)
        mlflow.log_metric("accuracy", acc)
        mlflow.sklearn.log_model(model, "knn_model")
        return k, acc, classification_report(y_test, y_pred, output_dict=True)

results = []
for k in range(1, 11):
    k_val, acc, report = evaluate_knn(k)
    results.append((k_val, acc))

df_results = pd.DataFrame(results, columns=["K", "Accuracy"])
print(df_results)
```

```
2025/08/25 09:06:03 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:07 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:07 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:11 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:11 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:16 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:16 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:20 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:20 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:24 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:24 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:28 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:28 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:33 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:33 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:36 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:36 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:40 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:40 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:49 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
    K  Accuracy
0   1  0.959064
1   2  0.959064
2   3  0.953216
3   4  0.953216
4   5  0.959064
5   6  0.959064
6   7  0.964912
7   8  0.959064
8   9  0.964912
9  10  0.964912
```

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, classification_report
import mlflow
import mlflow.sklearn

data = load_breast_cancer()
X = data.data
y = data.target

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, random_state=42, stratify=y
)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

def evaluate_knn(k):
    with mlflow.start_run(run_name=f"KNN_k={k}"):
        model = KNeighborsClassifier(n_neighbors=k)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        acc = accuracy_score(y_test, y_pred)
        mlflow.log_param("n_neighbors", k)
        mlflow.log_metric("accuracy", acc)
        mlflow.sklearn.log_model(model, "knn_model")
        return k, acc, classification_report(y_test, y_pred, output_dict=True)

results = []
```

```
for k in range(1, 11):
    k_val, acc, report = evaluate_knn(k)
    results.append((k_val, acc))

df_results = pd.DataFrame(results, columns=["K", "Accuracy"])
print(df_results)
```

```
2025/08/25 09:06:49 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:53 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:53 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:06:57 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:06:57 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:01 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:07:01 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:05 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:07:05 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:09 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:07:09 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:13 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:07:13 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:17 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:07:17 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:21 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:07:21 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:28 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
2025/08/25 09:07:28 WARNING mlflow.models.model: `artifact_path` is deprecated. Please use `name` instead.
2025/08/25 09:07:32 WARNING mlflow.models.model: Model logged without a signature and input example. Please set `input_example`
    K  Accuracy
0   1  0.959064
1   2  0.959064
2   3  0.953216
3   4  0.953216
4   5  0.959064
5   6  0.959064
6   7  0.964912
7   8  0.959064
8   9  0.964912
9  10  0.964912
```

```
input_example = pd.DataFrame(X_train[:5], columns=data.feature_names).astype(np.float64)
input_example_no_names = input_example.values  # numpy array without feature names

def evaluate_knn(k, weight_type):
    with mlflow.start_run(run_name=f"KNN_k={k}_{weight_type}"):
        model = KNeighborsClassifier(n_neighbors=k, weights=weight_type)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        acc = accuracy_score(y_test, y_pred)
        mlflow.log_param("n_neighbors", k)
        mlflow.log_param("weights", weight_type)
        mlflow.log_metric("accuracy", acc)
        mlflow.sklearn.log_model(model, name="knn_model", input_example=input_example_no_names)
        return acc
```

```
import numpy as np
import pandas as pd
from sklearn.datasets import load_breast_cancer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
import mlflow
import mlflow.sklearn
import matplotlib.pyplot as plt

data = load_breast_cancer()
X, y = data.data, data.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

input_example = pd.DataFrame(X_train[:5], columns=data.feature_names).astype(np.float64)
input_example_no_names = input_example.values

def evaluate_knn(k, weight_type):
    with mlflow.start_run(run_name=f"KNN_k={k}_{weight_type}"):
        model = KNeighborsClassifier(n_neighbors=k, weights=weight_type)
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        acc = accuracy_score(y_test, y_pred)
        mlflow.log_param("n_neighbors", k)
        mlflow.log_param("weights", weight_type)
        mlflow.log_metric("accuracy", acc)
        mlflow.sklearn.log_model(model, name="knn_model", input_example=input_example_no_names)
        return acc

k_values = range(1, 21)
uniform_acc = []
weighted_acc = []

for k in k_values:
    uniform_acc.append(evaluate_knn(k, "uniform"))
    weighted_acc.append(evaluate_knn(k, "distance"))

plt.figure(figsize=(10,6))
plt.plot(k_values, uniform_acc, label="Uniform Weights", marker='o')
plt.plot(k_values, weighted_acc, label="Distance Weights", marker='s')
plt.xlabel("Number of Neighbors (k)")
plt.ylabel("Accuracy")
plt.title("KNN Accuracy Comparison: Uniform vs Weighted")
plt.legend()
plt.grid(True)
plt.show()
```