Iris Flower Classification

```
import pandas as pd

from sklearn.datasets import load_iris

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score

iris = load_iris()

X = iris.data

y = iris.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

models = [LogisticRegression(), KNeighborsClassifier(), SVC(kernel='linear'), SVC(kernel='rbf'), ]

for model in models:
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test, y_pred)
    print(f"Accuracy of {model.__class__.__name__}: {accuracy:.2f}")
```

```
Accuracy of LogisticRegression: 1.00
Accuracy of KNeighborsClassifier: 1.00
Accuracy of SVC: 1.00
Accuracy of SVC: 1.00
/usr/local/lib/python3.10/dist-packages/sklearn/linear_model/_logistic.py:458: ConvergenceWarning: lbfgs failed to converge (status=
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```
best_model = models[2]

new_data = [[5.1, 3.5, 1.4, 0.2]]

prediction = best_model.predict(new_data)

print("Predicted species:", iris.target_names[prediction[0]])
```

```
Predicted species: setosa
```