```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from mpl_toolkits import mplot3d
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
```

```python
class Perceptron:
    def __init__(self, learning_rate = 0.1):
        self.learning_rate = learning_rate
        self.b = 0.0
        self.w = None
        self.misclassified_samples = []

    def fit(self, x: np.array, y: np.array, n_iter = 10):
        self._b = 0.0
        self._w = np.zeros(x.shape[1])
        self.misclassified_samples = []
        for _ in range(n_iter):
            errors = 0
            for xi, yi in zip(x, y):
                update = self.learning_rate * (yi - self.predict(xi))
                self._b += update
                self._w += update * xi
                errors += int(update != 0.0)
            self.misclassified_samples.append(errors)

    def f(self, x: np.array) -> float:
        return np.dot(x, self._w) + self._b

    def predict(self, x: np.array) -> int:
        return np.where(self.f(x) >= 0, 1, -1)

url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data'
df = pd.read_csv(url, header=None)
print(df.head())
y = df.iloc[:, 4].values
x = df.iloc[:, 0:3].values
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.set_title('Iris data set')
ax.set_xlabel("Sepal length in width (cm)")
ax.set_ylabel("Sepal width in width (cm)")
ax.set_zlabel("Petal length in width (cm)")
ax.scatter(x[:50,0], x[:50,1], x[:50,2], color='red',marker='o', s=4, label="Iris Setosa")
ax.scatter(x[50:100,0], x[50:100,1], x[50:100,2], color='blue',marker='^', s=4, label="Iris Versicolour")
ax.scatter(x[100:150,0], x[100:150,1], x[100:150,2], color='green',marker='x', s=4, label="Iris Virginica")
plt.legend(loc='upper left')
plt.show()
x = x[0:100, 0:2]
y = y[0:100]
plt.figure(figsize=(4,4))
plt.scatter(x[:50, 0], x[:50, 1], color='red', marker='o', label='Setosa')
plt.scatter(x[50:100, 0], x[50:100, 1], color='blue', marker='x',label='Versicolour')
plt.xlabel("Sepal length")
plt.ylabel("Petal length")
plt.legend(loc='upper left')
plt.show()
y = np.where(y == 'Iris-setosa', 1, -1)
x[:, 0] = (x[:, 0] - x[:, 0].mean()) / x[:, 0].std()
x[:, 1] = (x[:, 1] - x[:, 1].mean()) / x[:, 1].std()
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25,random_state=0)
classifier = Perceptron(learning_rate=0.01)
classifier.fit(x_train, y_train)
print("accuracy", accuracy_score(classifier.predict(x_test), y_test)*100)
plt.figure(figsize=(4,4))
plt.plot(range(1, len(classifier.misclassified_samples) + 1),classifier.misclassified_samples, marker='o')
plt.xlabel('Epoch')
plt.ylabel('Errors')
plt.show()
```
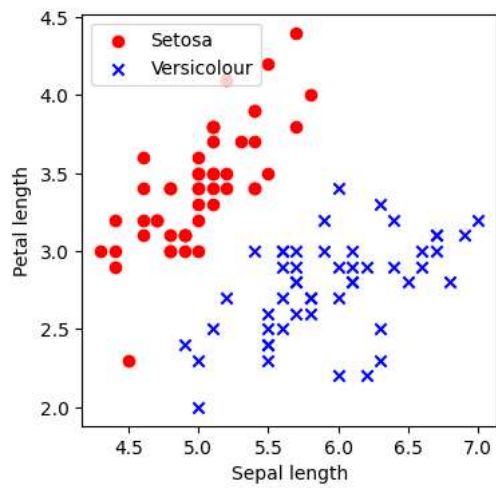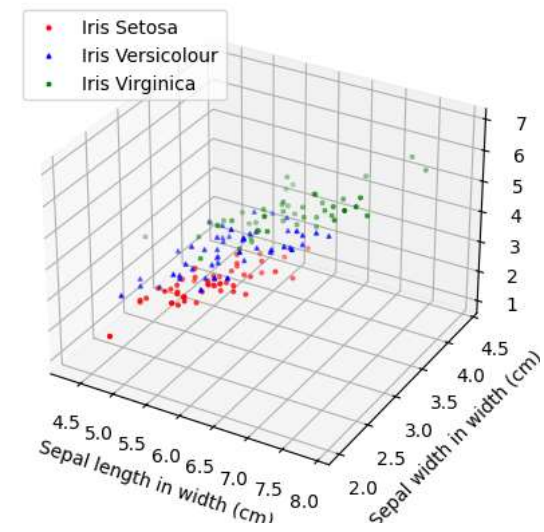
```
     0    1    2    3             4
0  5.1  3.5  1.4  0.2  Iris-setosa
1  4.9  3.0  1.4  0.2  Iris-setosa
2  4.7  3.2  1.3  0.2  Iris-setosa
3  4.6  3.1  1.5  0.2  Iris-setosa
4  5.0  3.6  1.4  0.2  Iris-setosa
```

Iris data set


```
accuracy 100.0
```