Lab-5

Logistics Regression

```
import numpy as np

class LogisticsRegression:
    def __init__(self, learning_rate=0.01,
                 n=1000):
        self.learning_rate = learning_rate
        self.num_ite = num_ite
        self.weights = None
        self.bias = None

def sigmoid(self, z):
    return 1/(1+np.exp(-z))

def fit(self, x, y)
    n_samples, n_features = x.shape
    self.weights = np.zeros(n_features)
    self.bias = 0

for _ in range(self.n):
    linear_model = np.dot(x, self.weights
                          + self.bias)

    y_predicted = self.sigmoid(linear_
                               model)

dw = (1/n_samples) * np.dot(X.T,
                            (y_predicted-y))

db = (1/n_samples) * np.sum(
                    y_predicted - y))
```

```
self.weights -= self.learning_rate * dw
self.bias = self.learning_rate * db

def predict(self, X):
linear_model = np.dot(x, self.weights
y_p = self.sigmoid(linear_model)
y_p_cls = [1 if i > 0.5 else 0
                for i in y_p]

return np.array(y_p_cls)

if __name__ == "__main__":

    x = np.array([1, 2], [2, 3],
        [3, 4], [5, 6])

    y = np.array([0, 0, 1, 1, 1])

    model = LogisticRegression()
    model.fit(x, y)

    y_pred = model.predict(x)

    print("Predictions:", y_pred)
```

output: [0 1 1 1 1]