Problem 2

```
from svm_classifier import SVM
```

SVM modul

```
class SVM(alpha=0.001, lambda_param=0.01, iters=1000)
```

Parameters

`alpha`: The learning rate

`lambda_param`: The regulization parameter

`iters`: The max number of iterations for the gradient descent

Attributes:

`w`: Weights assigned to the features

Methods:

`fit(X,y)`: Fit the SVM model according to the given training data.

`compute_cost(X, y)`: Calculate the cost function for the given data

`predict(X)`: Perform classification on samples in X.

The cost function

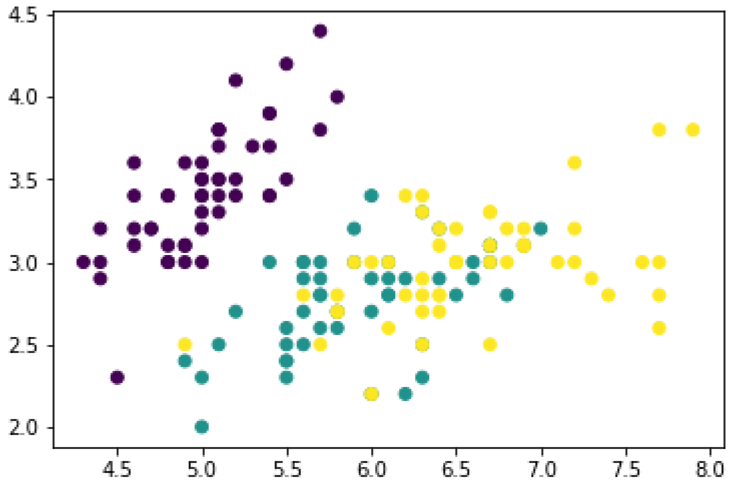
$$J = \lambda \|w\|^2 + \frac{1}{n} \sum_{i=1}^{n} max(0, 1 - y_i(w \cdot x_i - b))$$

Loading the iris data

```
from sklearn import datasets
import matplotlib.pyplot as plt

iris = datasets.load_iris()
X = iris.data[:, :2]
y = iris.target
plt.scatter(X[:,0],X[:,1],c = y)
```

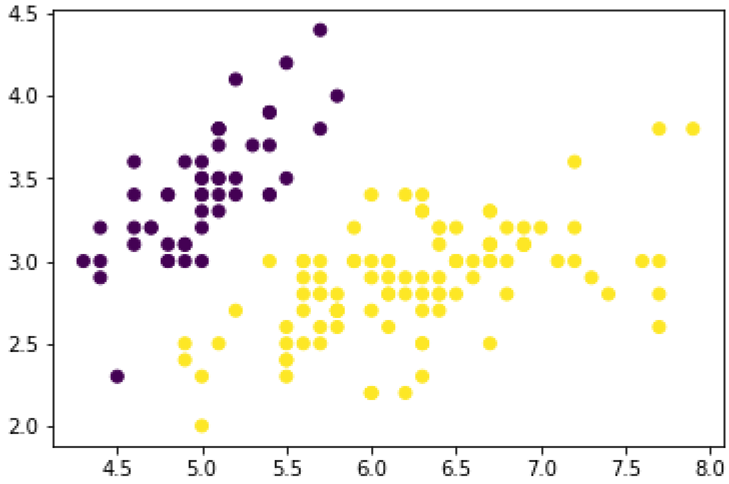
```
<matplotlib.collections.PathCollection at 0x2285feba1a0>
```



The iris data has 3 categories so we need to convert it to a binary classification problem for category 0

```
import numpy as np
y = np.where(y == 0, -1, 1)
plt.scatter(X[:,0],X[:,1],c = y)
```

```
<matplotlib.collections.PathCollection at 0x2285ff29ba0>
```



Splitting the data Creating the SVM modul

```
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size=0.20,
random_state=1)
svm = SVM()
```

```
svm.fit(X_train,Y_train)
from sklearn.metrics import accuracy_score
print("The classification accuracy:")
print(accuracy_score(Y_test,svm.predict(X_test)))
```

```
The classification accuracy:
1.0
```