KUMAR GAURAV

20122065

Support vector machine

In [1]:

```
import pandas as pd
import numpy as np
```

In [7]:

```
df=pd.read_csv(r'C:\Users\teres\Downloads\creditcard.csv')
df.head()
```

Out[7]:

```
      Time        V1         V2         V3         V4         V5         V6         V7         V8

0     0.0       -1.359807      -0.072781      2.536347       1.378155       -0.338321
        0.462388       0.239599       0.098698

1     0.0       1.191857       0.266151       0.166480       0.448154       0.060018       -
      0.082361  -0.078803      0.085102

2     1.0       -1.358354      -1.340163      1.773209       0.379780       -0.503198
        1.800499       0.791461       0.247676

3     1.0 -0.966272 -0.185226 1.792993 -0.863291 -0.010309 1.247203 0.237609 0.377436 4 2.0 -1.158233

      0.877737 1.548718 0.403034 -0.407193 0.095921 0.592941 -0.270533

5 rows × 31 columns
```

[3]:

```
class SVM:           def __init__(self, learning_rate= 0.001, lambda_param = 0.01,
n_iters = 1000):
        self.lr = learning_rate
self.lambda_param = lambda_param
self.n_iters = n_iters          self.w =
None          self.b = None
            def
fit(self, X, y):
        n_samples, n_features = X.shape

        #Basic check to convert 0, 1 to -1, 1
y_ = np.where(y <= 0, -1, 1)

        # initialize the weights and bias
self.w = np.zeros(n_features)
self.b = 0

        #Gradient descenting for finding minimum error and maximum accuracy
for _ in range(self.n_iters):
```

```
In
            #On every iteration go to each record and perform the below
for idx, x_i in enumerate(X):
                condition = y[idx] * (np.dot(x_i, self.w) - self.b) >=1

if condition:
                    self.w -= self.lr * (2 * self.lambda_param * self.w)
else:
                    self.w -= self.lr * (2 * self.lambda_param * self.w - np.dot(x_i, y_[id
self.b -= self.lr * y_[idx]

def predict(self, X):
# w * x - b
        linear_model = np.dot(X, self.w) - self.b

return np.sign(linear_model) In [5]:
```

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

In [26]:

```
X=df[['Time','V1','V2','V3','V4','V5','V6','V7','V8','V9','V10','V11','V12','V13','V14','V1
```

In [27]:

```
y=df['Class']
```

In [28]:
[29]:

```
y
```

Out[29]:

```
array([-1, -1, -1, ..., -1, -1, -1])
```

In [30]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.3)
```

In [31]:

```
X.shape
```

Out[31]:

```
(284807, 30)
```

In [32]:

```
y = np.where(y==0, -1, 1)
```

In

```
X_train.shape
```

Out[32]:

```
(199364, 30)
```

In [33]:

```
X_test.shape
```

Out[33]:

```
(85443, 30)
```

In [34]:

```
print("Training X: ",X_train.shape)
print("Testing X: ",X_test.shape)
print('Training Y: ',y_train.shape)
print("Testing Y: ",y_test.shape)
```

```
Training X:  (199364, 30)
Testing X:  (85443, 30)
Training Y:  (199364,)
Testing Y:  (85443,)
```

In [35]:

```
clf = SVM()
```

In [36]:

```
X_train = X_train.astype(float)
X_test = X_test.astype(float)
y_train = y_train.astype(float)
y_test = y_test.astype(float)
```

[37]:

```
#Train the model
'''
    X_train: 70 features
    y_train: 70 answers
'''
clf.fit(X_train,y_train)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-37-0aac09a17d23> in <module>
 4     y_train: 70 answers
 5     '''
----> 6 clf.fit(X_train,y_train)

<ipython-input-3-b42d52bae29e> in fit(self, X, y)
  22   #On every iteration go to each record and perform the be low
```

```
In
  23     for idx, x_i in enumerate(X):
---> 24                 condition = y[idx] * (np.dot(x_i, self.w) - self.b)
>=1
     25
     26                 if condition:

<__array_function__ internals> in dot(*args, **kwargs)

TypeError: Cannot cast array data from dtype('float64') to dtype('<U32') acc

ording to the rule 'safe' In [38]:
```

```
predictions = clf.predict(X)
```

In [39]:

```
print("Weights: ", clf.w, "Bias: ", clf.b)
```

```
Weights:  [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0.
 0. 0. 0. 0. 0. 0.] Bias:  0
```

[40]:

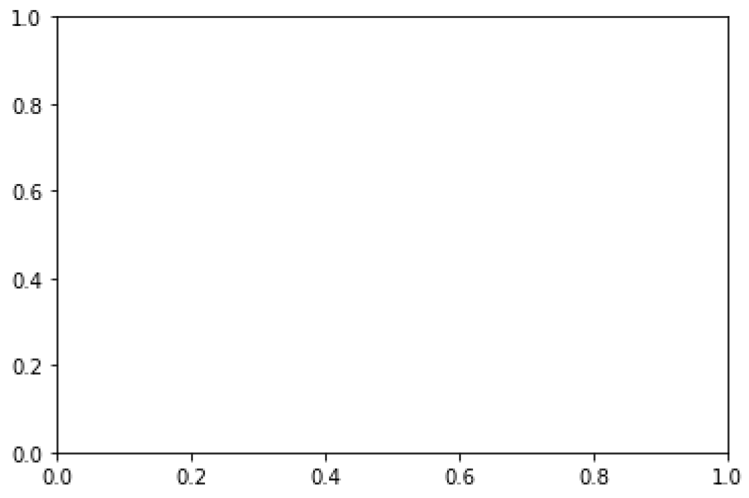
```
def visualize():

    def get_hyperplane_value(x, w, b, offset):
        return (-w[0] * x + b + offset)/w[1]

    fig = plt.figure()
    ax = fig.add_subplot(1,1,1)
    plt.scatter(X[:,0], X[:,1], marker = 'o', c=y)

    x0_1 = np.amin(X[:, 0])
    x0_2 = np.amax(X[:, 0])

    x1_1 = get_hyperplane_value(x0_1, clf.w, clf.b, 0)
    x1_2 = get_hyperplane_value(x0_2, clf.w, clf.b, 0)

    x1_1_m = get_hyperplane_value(x0_1, clf.w, clf.b, -1)
    x1_2_m = get_hyperplane_value(x0_2, clf.w, clf.b, -1)

    x1_1_p = get_hyperplane_value(x0_1, clf.w, clf.b, 1)
    x1_2_p = get_hyperplane_value(x0_2, clf.w, clf.b, 1)

    ax.plot([x0_1, x0_2], [x1_1, x1_2], 'y--')
    ax.plot([x0_1, x0_2], [x1_1_m, x1_2_m], 'k')
    ax.plot([x0_1, x0_2], [x1_1_p, x1_2_p], 'k')

    x1_min = np.amin(X[:, 1])
    x1_max = np.amax(X[:, 1])

    ax.set_ylim([x1_min-3, x1_max+3])

    plt.show()
```

[14]:

In

```
%matplotlib inline
visualize()
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-14-06d501a5d9de> in <module>
      1 get_ipython().run_line_magic('matplotlib', 'inline')
----> 2 visualize()

<ipython-input-13-9aab59215487> in visualize()
  6     fig = plt.figure()
  7     ax = fig.add_subplot(1,1,1)
----> 8     plt.scatter(X[:,0], X[:,1], marker = 'o', c=y)
      9
     10     x0_1 = np.amin(X[:, 0])

~\anaconda3\lib\site-packages\pandas\core\frame.py in __getitem__(self, key)
 2798                 if self.columns.nlevels > 1:
 2799                 return self._getitem_multilevel(key)
-> 2800             indexer = self.columns.get_loc(key)
 2801                 if is_integer(indexer):
 2802                 indexer = [indexer]

~\anaconda3\lib\site-packages\pandas\core\indexes\base.py in get_loc(self, k
ey, method, tolerance)   2644                     )
   2645             try:
-> 2646                 return self._engine.get_loc(key)
 2647                 except KeyError:
 2648                 return self._engine.get_loc(self._maybe_cast_indexer
(key)) pandas\_libs\index.pyx in

pandas._libs.index.IndexEngine.get_loc() pandas\_libs\index.pyx in

pandas._libs.index.IndexEngine.get_loc()

TypeError: '(slice(None, None, None), 0)' is an invalid key
```



[ ]:

In
In [ ]:

```
```