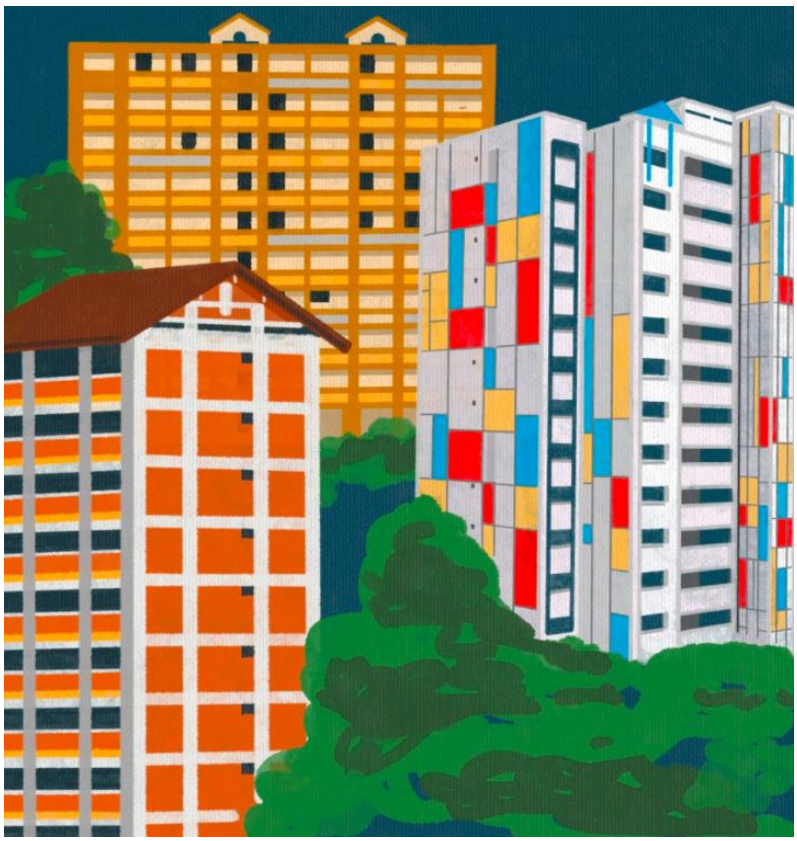
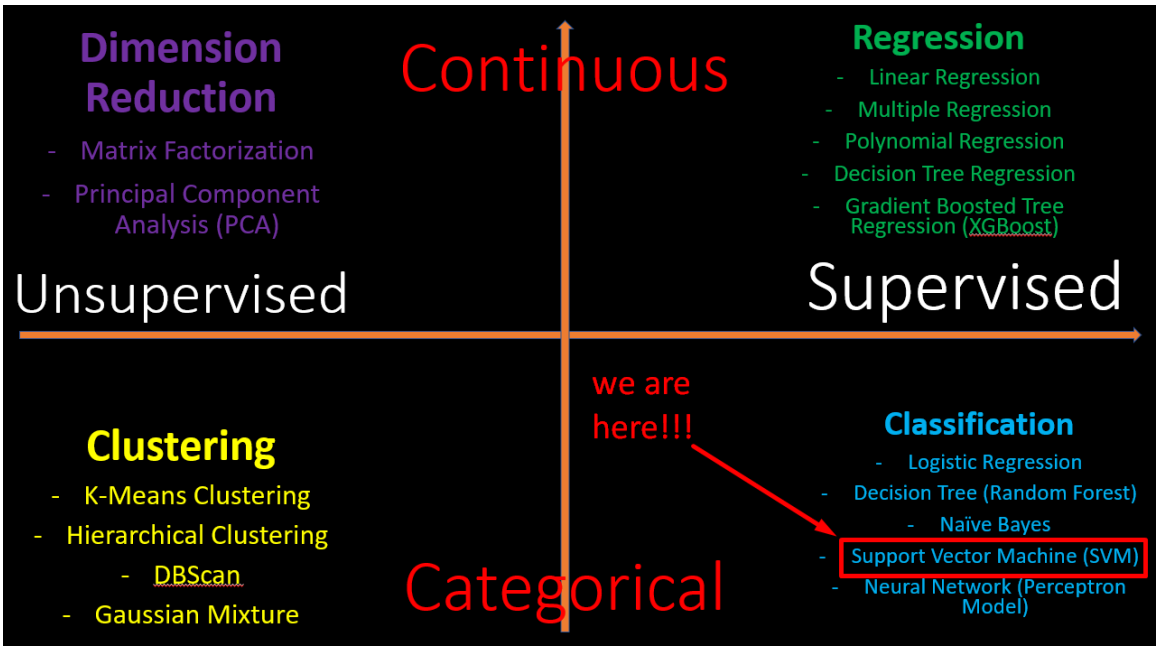
SIMPLE SVM APPLIED TO IRIS DATASET

WITH PYTHON

DR. ALVIN ANG



CONTENTS



This manuscript is the Second Part.

First part is https://www.alvinang.sg/s/Understanding-SVM-with-Python-by-Dr-Alvin-Ang.pdf

The third part (after this manuscript) will be "Hyperparameter Optimization with SVM".

II. STEP 1: IMPORT IRIS DATASET

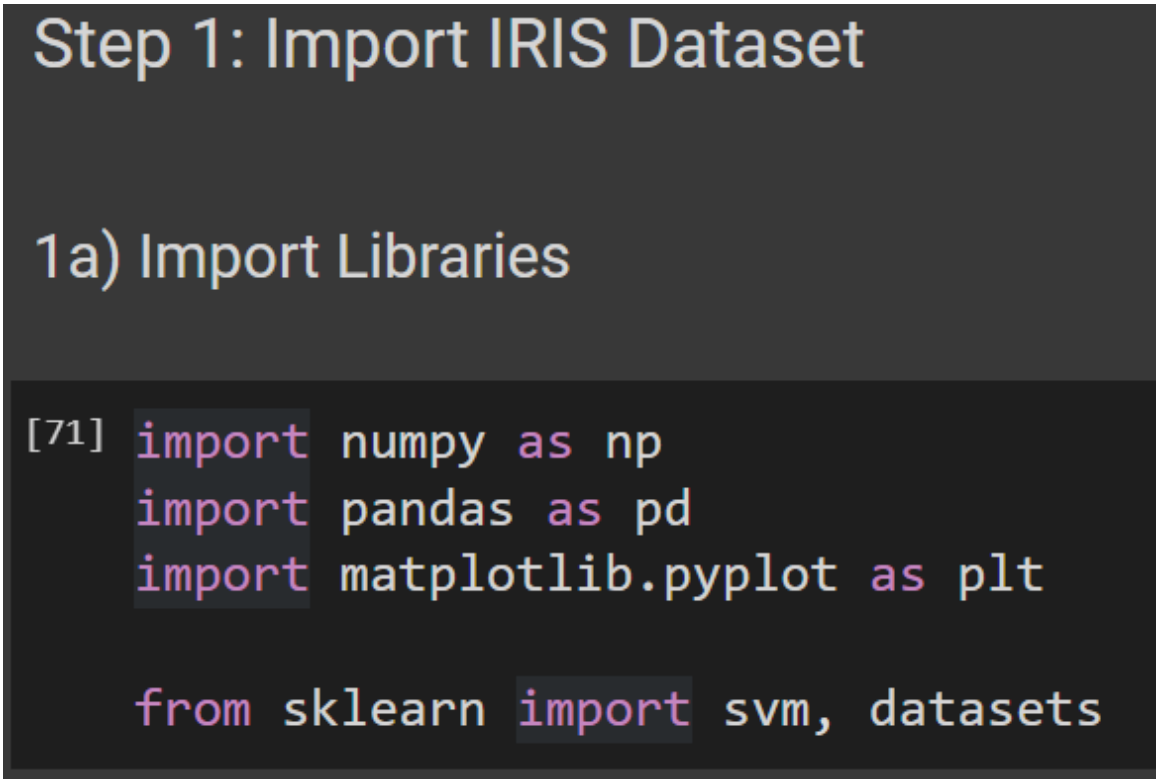
https://www.alvinang.sg/s/Simple_SVM_Applied_to_Iris_Dataset_with_Python_by_Dr_Alvin_Ang.ipynb

References

https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/

https://stackabuse.com/implementing-svm-and-kernel-svm-with-pythons-scikit-learn/

A. IMPORT LIBRARIES



B. LOAD IRIS DATASET

```
1b) Load Iris Dataset

[72] iris = datasets.load_iris()
```

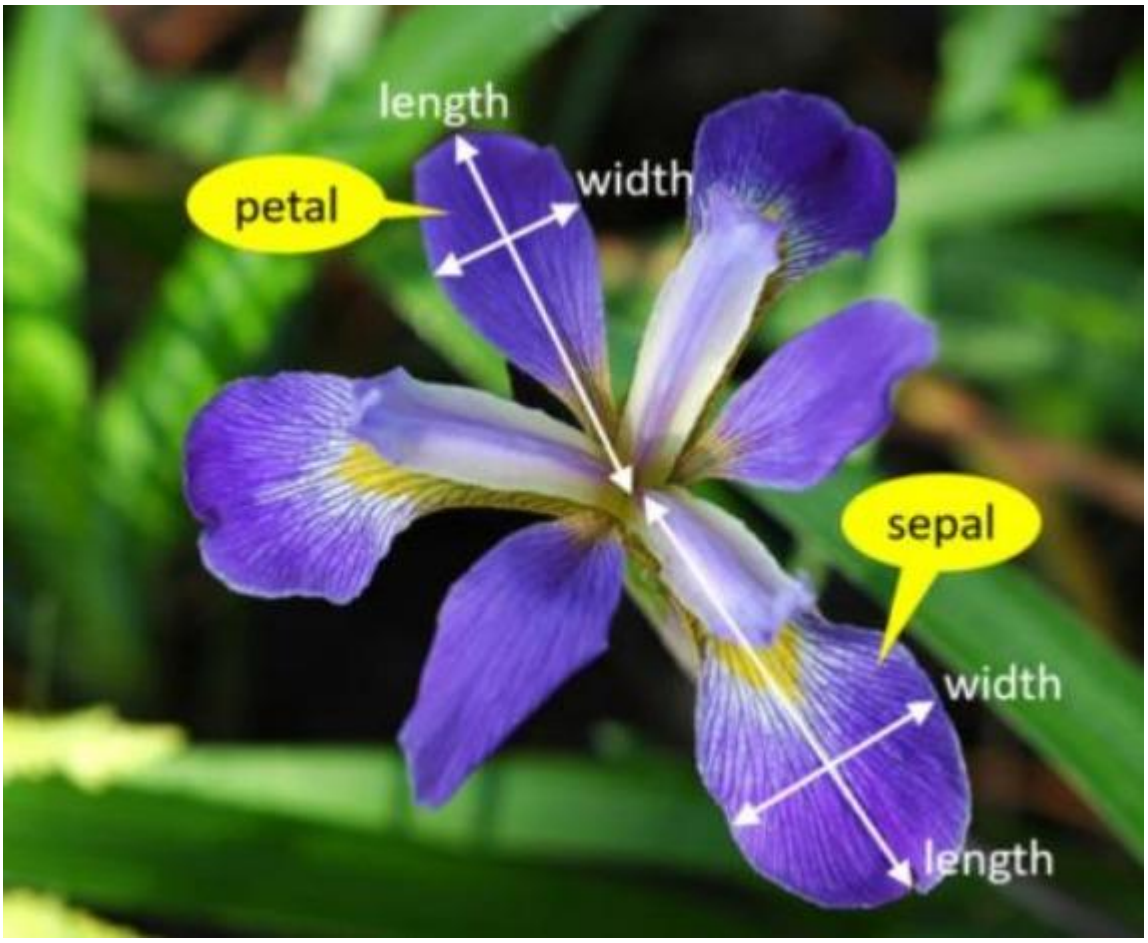
C. SLICING OUT FIRST TWO FEATURES (FOR X)

```
1c) Slicing Out First Two Features (for X)

X = iris.data[:, :2]
# we only take the first two features.
# we slice the first two columns

# 'feature_names': ['sepal length (cm)',
#  'sepal width (cm)',
#  'petal length (cm)',
#  'petal width (cm)'],

#we only take SEPAL LENGTH and SEPAL WIDTH (as features..X)
#to predict the TARGET (y)
```



D. MAKING THE PREDICTED TARGET Y

1d) Making the Predicted Target y

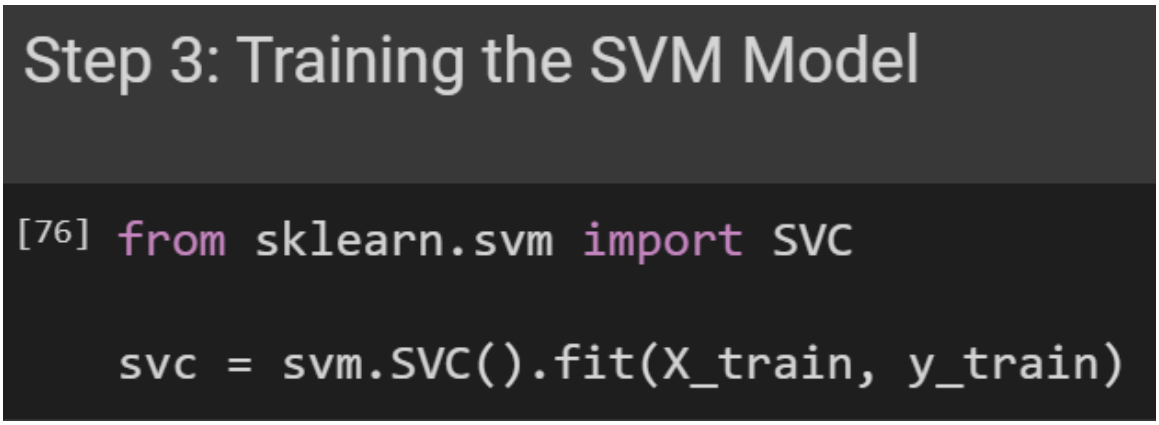
```
[74] y = iris.target

    #'target_names': array(['setosa', 'versicolor', 'virginica']
    # 0 = setosa
    # 1 = versicolor
    # 2 = virginica
```

Step 2: Train Test Split

```
[75] 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)

     #20% Testing, 80% Training
```

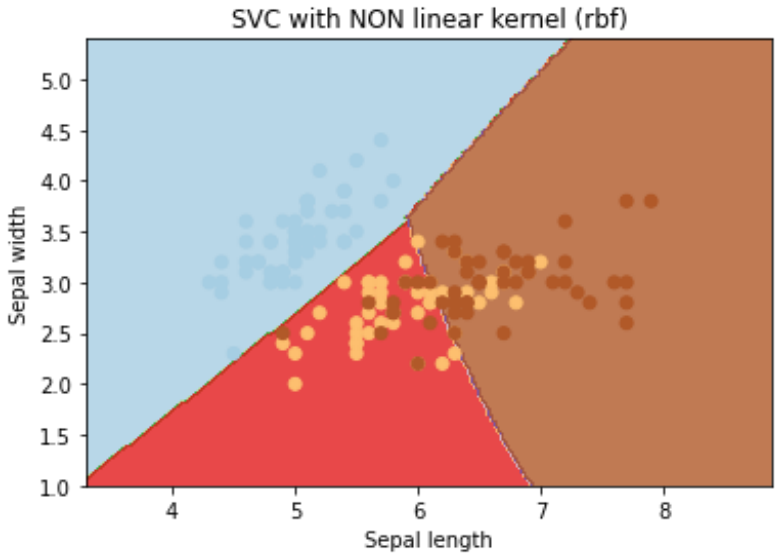


V. STEP 4: PLOTTING THE SVM

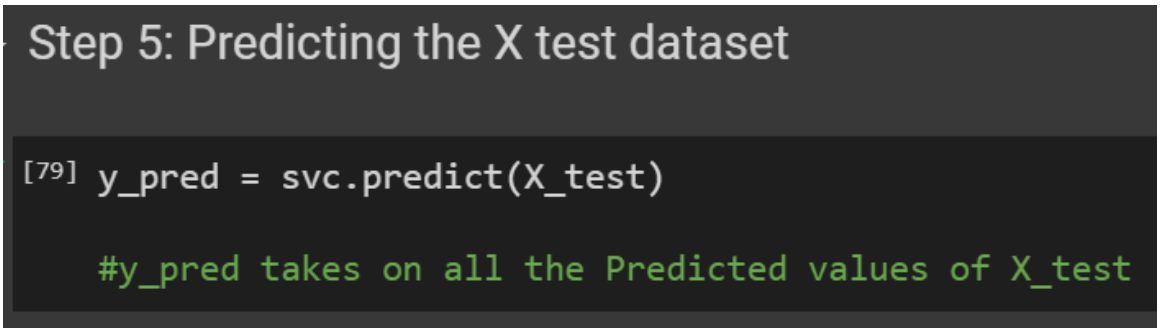
Step 4: Plotting the SVM

```
# create a mesh to plot in
x_min, x_max = X[:, 0].min() - 1, X[:, 0].max() + 1
y_min, y_max = X[:, 1].min() - 1, X[:, 1].max() + 1
h = (x_max / x_min)/100
xx, yy = np.meshgrid(np.arange(x_min, x_max, h),
 np.arange(y_min, y_max, h))
plt.subplot(1, 1, 1)
Z = svc.predict(np.c_[xx.ravel(), yy.ravel()])
Z = Z.reshape(xx.shape)

plt.contourf(xx, yy, Z, cmap=plt.cm.Paired, alpha=0.8)
plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Paired)
plt.xlabel('Sepal length')
plt.ylabel('Sepal width')
plt.xlim(xx.min(), xx.max())
plt.title('SVC with NON linear kernel (rbf)')
plt.show()
```




```
[78] #Here u see that the SVM has non-linear boundaries
     #It manages to classify into 3 categories:
     # 0 = setosa
     # 1 = versicolor
     # 2 = virginica
     #(but i'm not sure how the color matches those categories...)
```



Step 6: Confusion Matrix

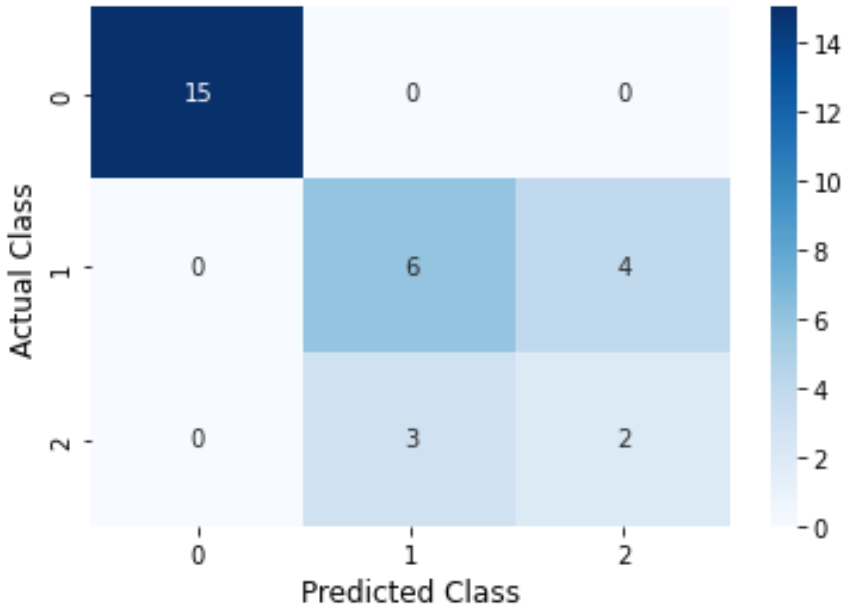
```
[80] from sklearn.metrics import classification_report, confusion_matrix
     print(confusion_matrix(y_test, y_pred))

     [[15  0  0]
      [ 0  6  4]
      [ 0  3  2]]
```

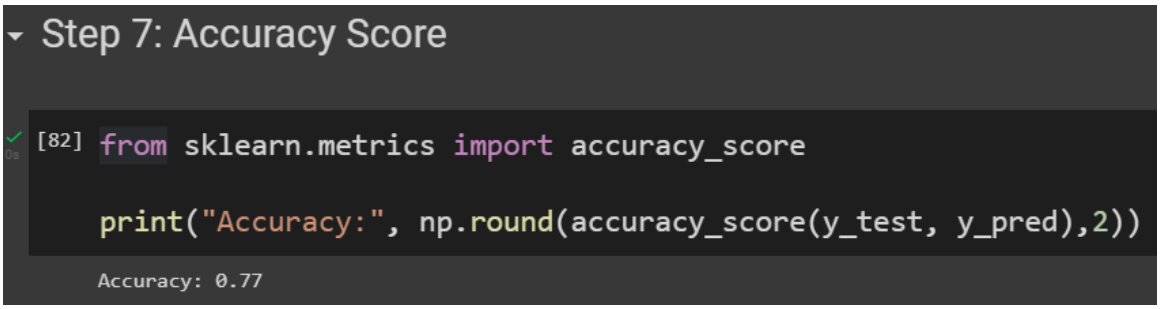
```
import seaborn as sns
from sklearn.metrics import confusion_matrix

cf_matrix = confusion_matrix(y_test, y_pred)

sns.heatmap(cf_matrix, annot=True, cmap='Blues')
plt.xlabel('Predicted Class', fontsize=12)
plt.ylabel('Actual Class', fontsize=12)
```



VIII. STEP 7: ACCURACY SCORE



Step 8: Future Work -- Hyperparameter Tuning

```
# Check default values
import pandas as pd

svc = SVC()
params = svc.get_params()
params_df = pd.DataFrame(params, index=[0])
params_df.T

#below shows the default hyperparameter values inside the SVC model
#we didn't change anything
```

	0
C	1.0
break_ties	False
cache_size	200
class_weight	None
coef0	0.0
decision_function_shape	ovr
degree	3
gamma	scale
kernel	rbf
max_iter	-1
probability	False
random_state	None
shrinking	True
tol	0.001
verbose	False

X. STEP 9: CONCLUSION

Step 9: Conclusion

```
#We used a Simple SVM applied to Iris Dataset
#(only using Sepal Length and Sepal Width as features to predict the Class)

#Accuracy = 77%

#We left ALL HYPERPARAMETERS as DEFAULT, notably:
#   Regularization Parameter C = 1
#   Gamma = 'scale', i.e. gamma = 1 / (n_features * X.var())
#   we shall explore how C and Gamma affects the model in the next manuscript

#   Kernel = 'rbf' (meaning its NON LINEAR)
#   we shall explore the different types of Kernel in the next manuscript
```

ABOUT DR. ALVIN ANG

Dr. Alvin Ang earned his Ph.D., Masters and Bachelor degrees from NTU, Singapore. He is a scientist, entrepreneur, as well as a personal/business advisor. More about him at www.AlvinAng.sg.