

```

import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn import linear_model
from sklearn import metrics
from sklearn.model_selection import train_test_split

#used code snippets from discussion
#use sepal width vs petal width or sepal length vs petal length for best results

iris = datasets.load_iris()
iris_df = pd.DataFrame(iris.data, columns = iris.feature_names)
iris_df['species'] = iris.target

X_train, X_test, Y_train, Y_test = train_test_split(iris_df.drop('species', axis = 1), iris_df['species'],
test_size = 0.3, train_size = 0.7, random_state = 51)

X_train.shape
X_test.shape
Y_train.shape
Y_test.shape

trainData = X_train.iloc[:, [1, 3]] #sepal width vs petal width
testData = X_test.iloc[:, [1, 3]] #sepal width vs petal width

regVersiVirg = linear_model.LinearRegression()
regSetVersi = linear_model.LinearRegression()

trainFilter = (Y_train == 1) | (Y_train == 2)
regVersiVirg.fit(trainData[trainFilter], (Y_train[trainFilter] == 2).astype(int))

```

```
trainFilter = (Y_train == 0) | (Y_train == 1)
regSetVersi.fit(trainData[trainFilter], (Y_train[trainFilter] == 0).astype(int))
```

```
predictVersiVirg = regVersiVirg.predict(testData)
predictSetVersi = regSetVersi.predict(testData)
```

```
prediction = np.zeros_like(predictVersiVirg, dtype = int) + 1
prediction[predictVersiVirg > 0.5] = 2
prediction[predictSetVersi > 0.5] = 0
```

```
accuracy = metrics.accuracy_score(Y_test, prediction)
```

```
versiVirgWeight = regVersiVirg.coef_
versiVirgBias = regVersiVirg.intercept_
```

```
setVersiWeight = regSetVersi.coef_
setVersiBias = regSetVersi.intercept_
```

```
print("Sepal Width vs Petal Width Accuracy", accuracy)
print()
print("Versicolor vs Virginica Weight:", versiVirgWeight)
print("Versicolor vs Virginica Bias:", versiVirgBias)
print()
print("Setosa vs Versicolor Weight:", setVersiWeight)
print("Setosa vs Versicolor:", setVersiBias)
```