Homework 3

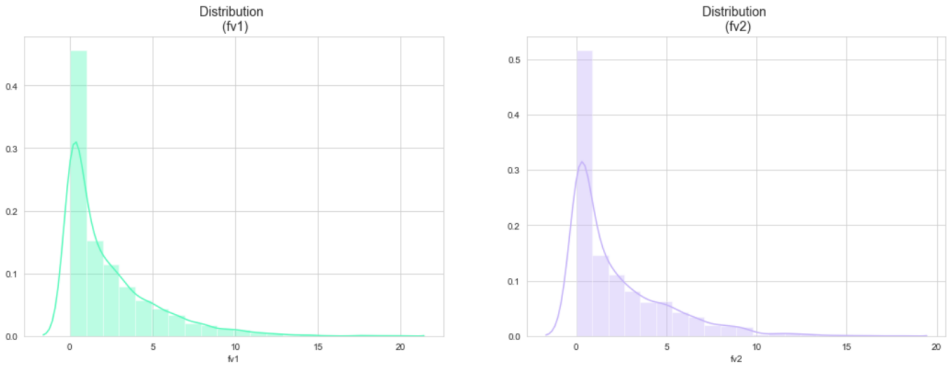
Abhishek Reddy Palle - 128003556

Problem 1)

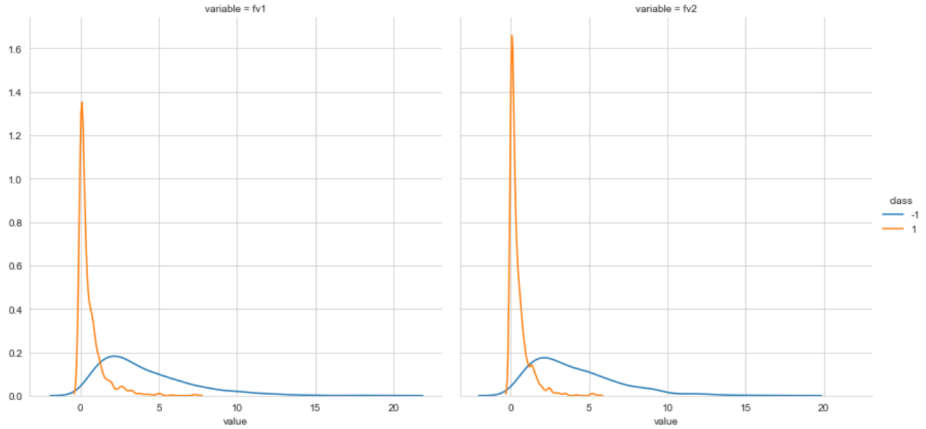
Least squares linear regression can be applied to the task of classification. The model seeks to find a line, (or, higher-dimensionally, a hyperplane) which separates input space into two distinct regions, one for each class. If such a division of space exists, we say that the two classes are linearly separable.

Part A)

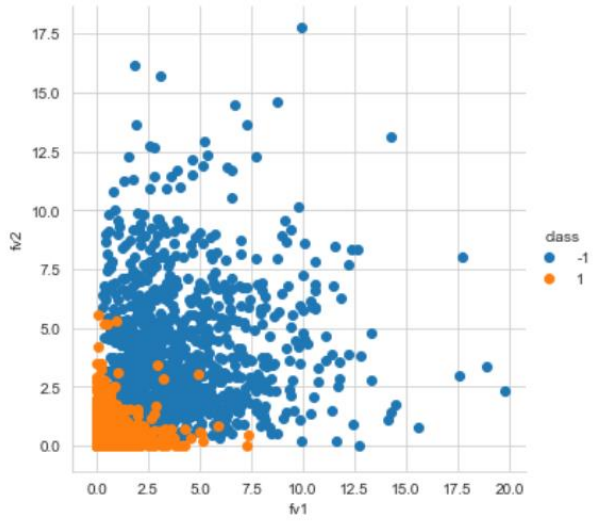
Feature Distribution

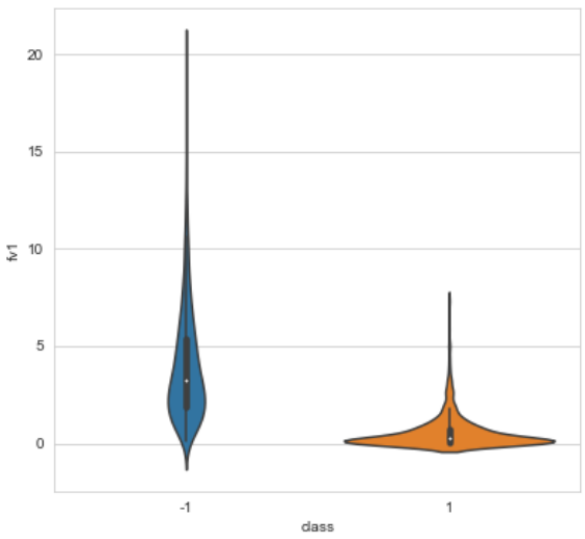


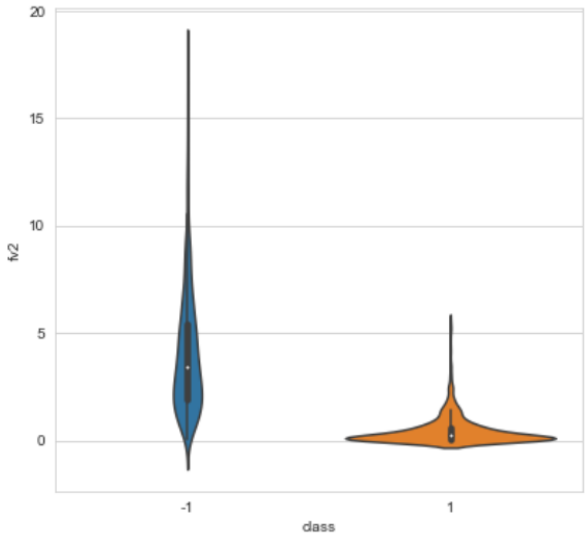
Class Conditioned Probability Distribution



Scatter plot of features – Visualizing Classes







Model Analysis

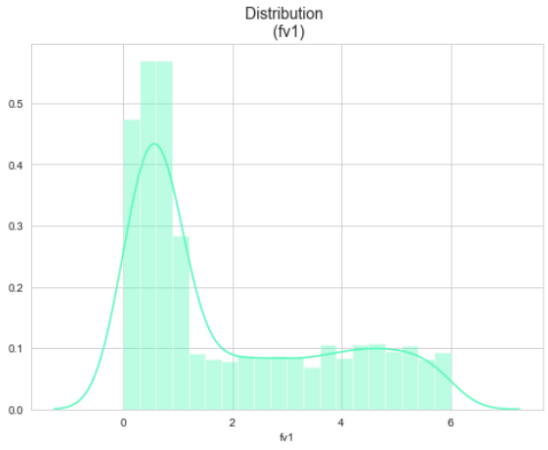
We compare the accuracy of linear least squares and logistic regression after learning a linear classifier. We vary training set sizes as 10, 50, 100, 500. In each case we assess the learnt classifier using the full test set. In each case, we use the same set of training examples (as used for estimation). We get the following insights:

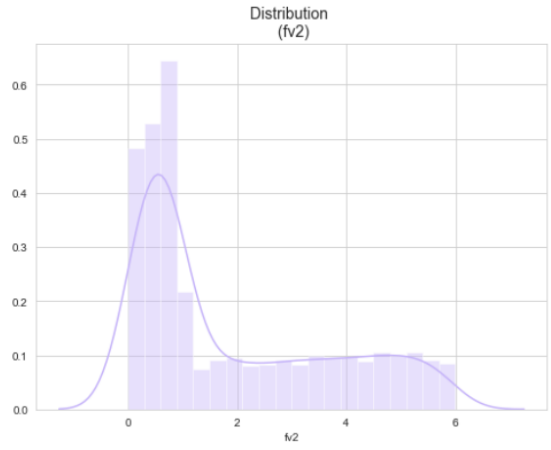
- We observe that accuracy increases as the training sample size increases.
- We observe that Logistic Regression performs better than Linear Least squares
- We also observe that accuracy achieved is decent since the data is pretty much separable as seen in the exploratory data analysis part.

```
Sample Size: 10
Accuracy of Linear Least Squares on test data using 10 training samples: 0.832
Accuracy of Logistic Regression on test data using 10 training samples: 0.844

Sample Size: 50
Accuracy of Linear Least Squares on test data using 50 training samples: 0.87
Accuracy of Logistic Regression on test data using 50 training samples: 0.901

Sample Size: 100
Accuracy of Linear Least Squares on test data using 100 training samples: 0.898
Accuracy of Logistic Regression on test data using 100 training samples: 0.914

Sample Size: 500
Accuracy of Linear Least Squares on test data using 500 training samples: 0.892
Accuracy of Logistic Regression on test data using 500 training samples: 0.928
```

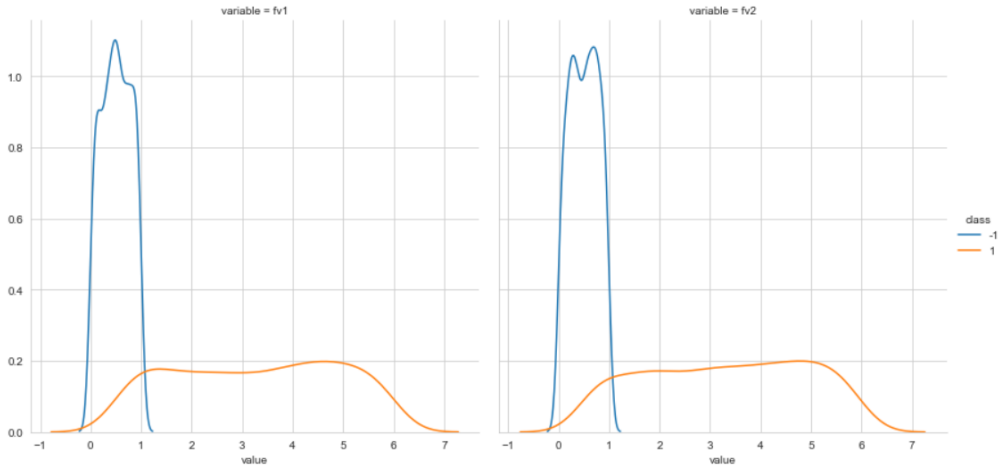
Part B)

Feature Distribution

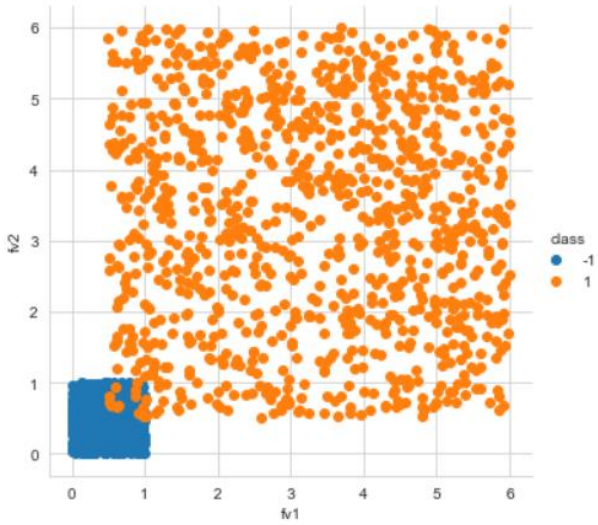


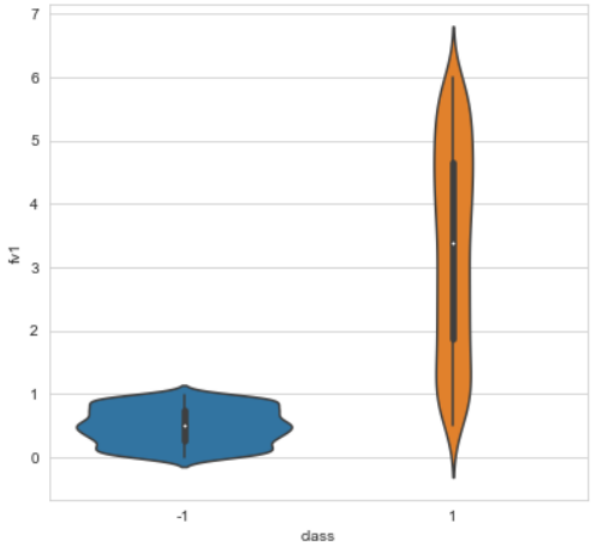


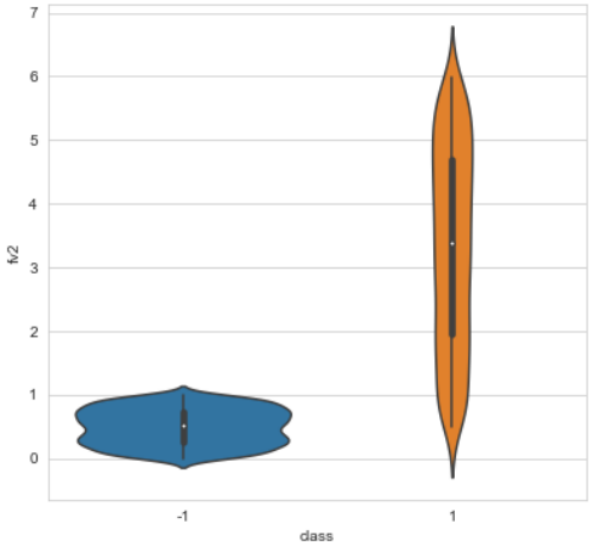
Class Conditioned Probability Distribution



Scatter plot of features – Visualizing Classes







Model Analysis

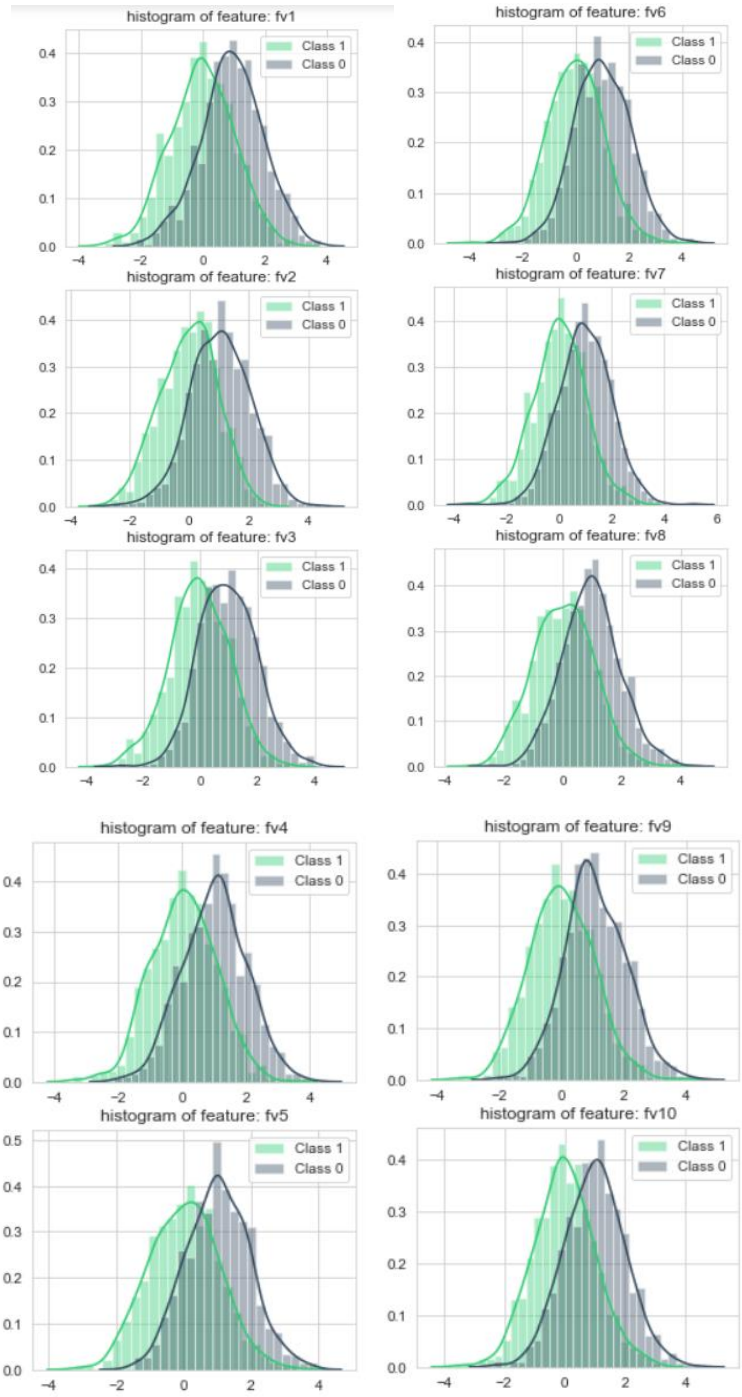
- We observe that accuracy increases as the training sample size increases.
- We observe that Logistic Regression performs better than Linear Least squares
- We also observe that accuracy achieved is decent since the data is pretty much separable as seen in the exploratory data analysis part.

```
Sample Size: 10
Accuracy of Linear Least Squares on test data using 10 training samples: 0.791
Accuracy of Logistic Regression on test data using 10 training samples: 0.936

Sample Size: 50
Accuracy of Linear Least Squares on test data using 50 training samples: 0.918
Accuracy of Logistic Regression on test data using 50 training samples: 0.988

Sample Size: 100
Accuracy of Linear Least Squares on test data using 100 training samples: 0.928
Accuracy of Logistic Regression on test data using 100 training samples: 0.983

Sample Size: 500
Accuracy of Linear Least Squares on test data using 500 training samples: 0.928
Accuracy of Logistic Regression on test data using 500 training samples: 0.983
```

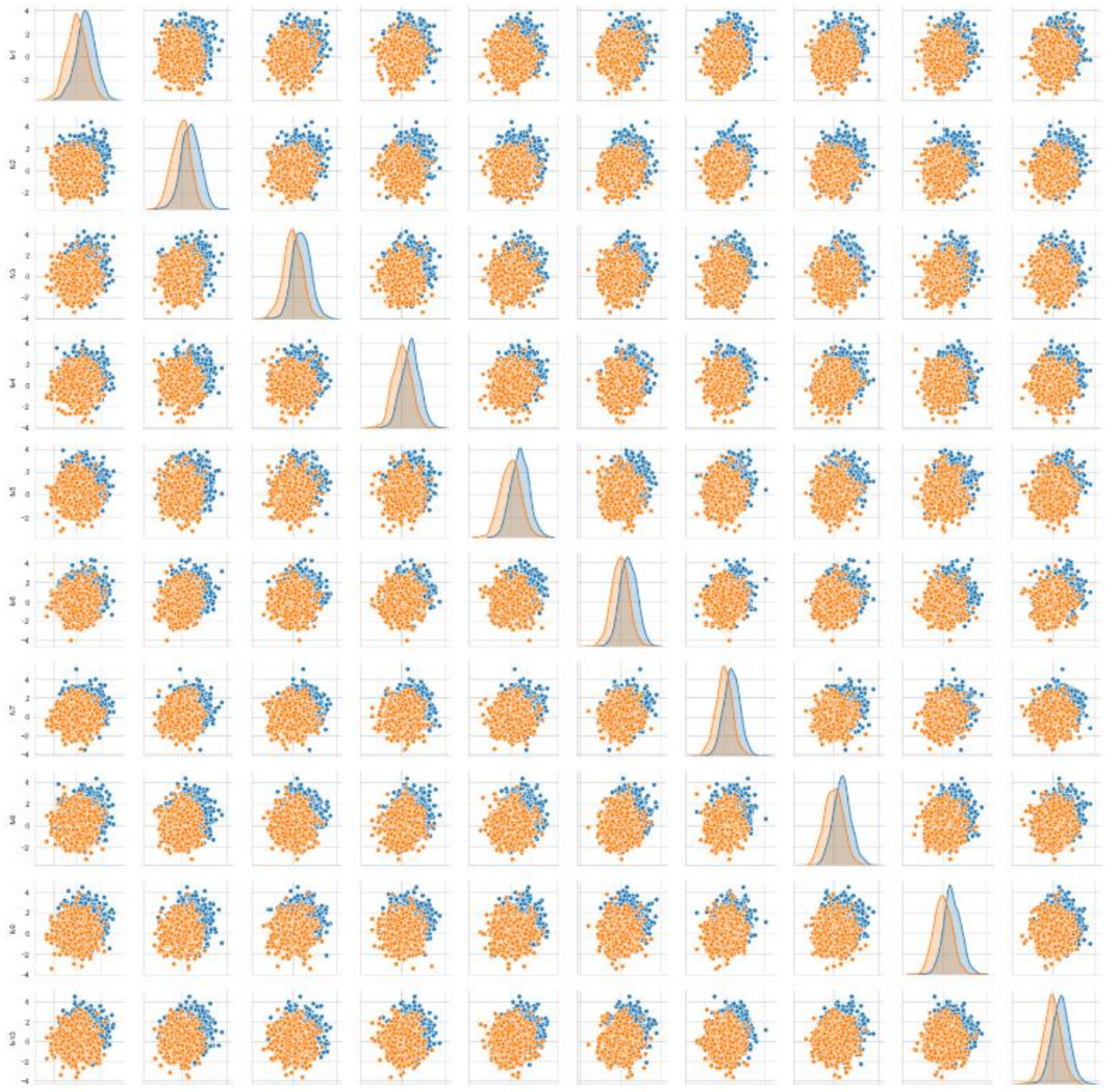
Part C)

Feature Distribution

The mean vector for class-I is all-zeros and that for class-II is all-ones.



Scatter plot of features – Visualizing Classes



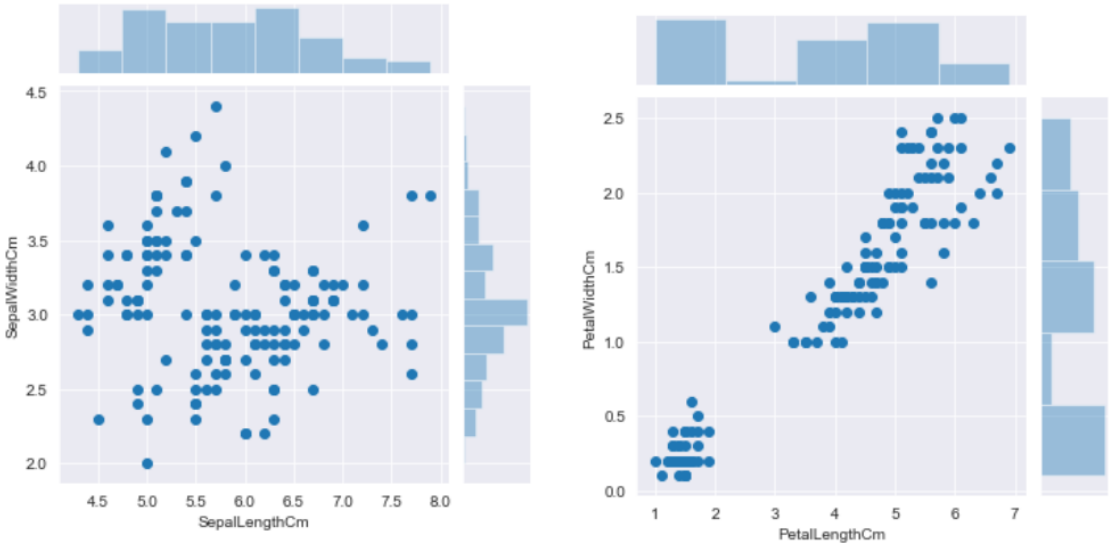
Model Analysis

- We observe that accuracy increases as the training sample size increases.
- We observe that Logistic Regression performs better than Linear Least squares
- We also observe that accuracy achieved is decent since the data is pretty much separable as seen in the exploratory data analysis part.

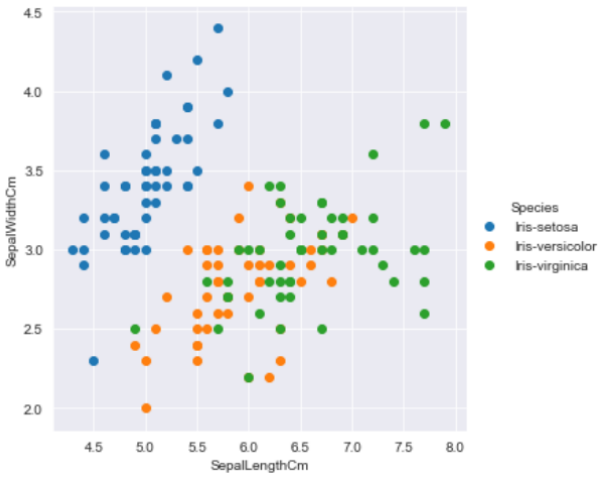
```
Sample Size: 10
Accuracy of Linear Least Squares on test data using 10 training samples: 0.495
Accuracy of Logistic Regression on test data using 10 training samples: 0.753

Sample Size: 50
Accuracy of Linear Least Squares on test data using 50 training samples: 0.916
Accuracy of Logistic Regression on test data using 50 training samples: 0.913

Sample Size: 100
Accuracy of Linear Least Squares on test data using 100 training samples: 0.918
Accuracy of Logistic Regression on test data using 100 training samples: 0.92

Sample Size: 500
Accuracy of Linear Least Squares on test data using 500 training samples: 0.941
Accuracy of Logistic Regression on test data using 500 training samples: 0.94
```

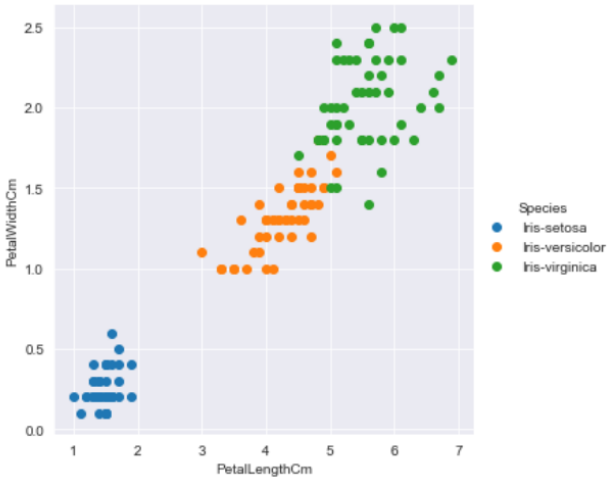
Problem 2)

Scatter plot of features – Visualizing Classes

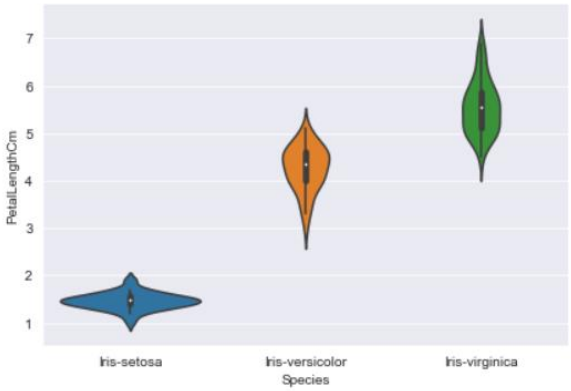


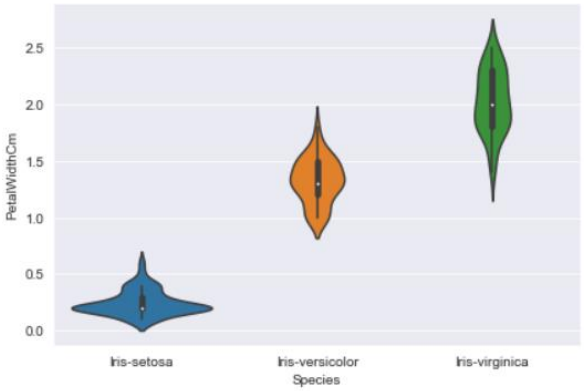
Scatter plot of features – Visualizing Classes

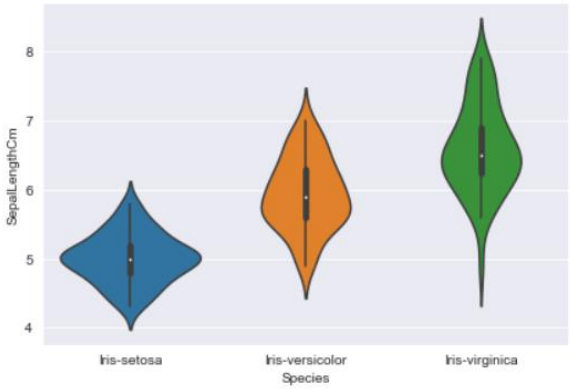


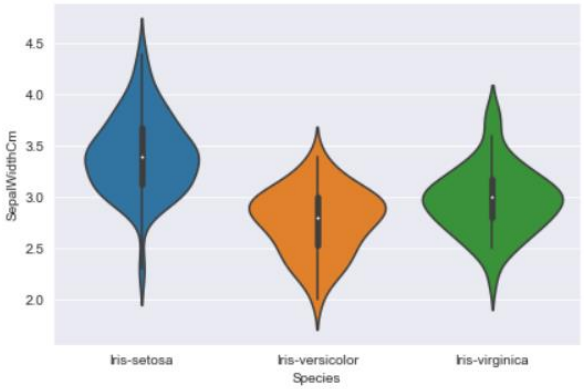


Feature Distribution

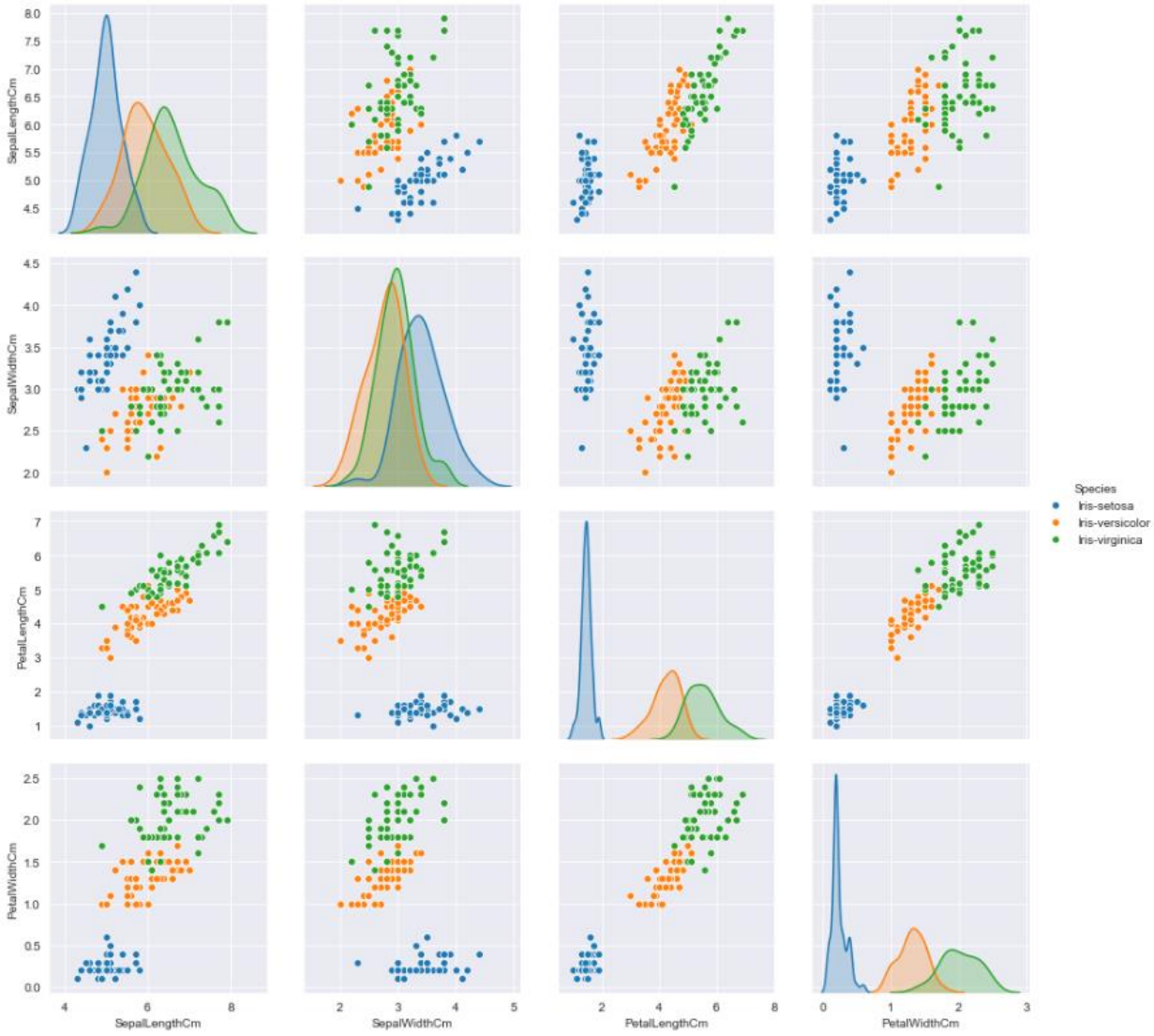








Pair plot among features



Model Analysis

1) **We learn three linear 2-class classifiers using 'OneVsRestClassifier' module in sklearn**

```
from sklearn.linear_model import LinearRegression
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import accuracy_score

ovr_cls = OneVsRestClassifier(LinearRegression())
ovr_cls.fit(X_train, y_train)
ovr_pred = ovr_cls.predict(X_test)

ovr_class = []
for i in ovr_pred:
    if i<=0.5:
        ovr_class.append(0)
    elif i > 0.5 and i <= 1.5:
        ovr_class.append(1)
    else:
        ovr_class.append(2)

print ("Accuracy using Linear Least Squares and 'one vs rest'on test data: {}".format(accuracy_score(ovr_class, y_test)))
```

2) **We learn a 3-class linear classifier (by taking the target or prediction variable as a 3-dimensional one-hot vector) using 'ClassiferChain'.**

```
new_iris = pd.get_dummies(iris, prefix=['Species'])
new_iris.head()
```

```
from sklearn.multioutput import ClassifierChain
randomChainClassifier = [ClassifierChain(LinearRegression(), order='random', random_state=i) for i in range(20)]
for chain in randomChainClassifier:
    chain.fit(X_train,y_train)

cc_pred = np.array([chain.predict(X_test) for chain in randomChainClassifier])
cc_pred = cc_pred.mean(axis=0)
```

Problem 3)

Model Analysis

- We observe that Linear Least squares performs poorly.
- Logistic Regression performs way better than Linear Least squares.

```
from sklearn.cross_validation import train_test_split
from sklearn.preprocessing import LabelEncoder

#Separating predictor and target variables
X = df.drop(['class'], axis=1)
y = df['class']

#Converting to Arrays
X = np.asarray(X)
y = np.asarray(y)

# Label Encoding
encoder = LabelEncoder()
y = encoder.fit_transform(y)

# Train/Test Split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size = 0.3, random_state = 0)
```

```
linreg = LinearRegression()
linreg.fit(X_train,y_train)
linreg_pred = linreg.predict(X_test)

linreg_class = []
for i in linreg_pred:
    if i>=0:
        linreg_class.append(1)
    else:
        linreg_class.append(2)
print ("Accuracy of Linear Least Squares on test data : {}".format(accuracy_score(linreg_class, y_test)))

# Train and predict the output using Logistic Regression
logreg = LogisticRegression()
logreg.fit(X_train,y_train)
logreg_pred = logreg.predict(X_test)
print ("Accuracy of Logistic Regression on test data : {}".format(accuracy_score(logreg_pred, y_test)))
```

```
Accuracy of Linear Least Squares on test data : 0.283333333333
Accuracy of Logistic Regression on test data : 0.773333333333
```

Problem 4)

Here we use linear least squares to fit different polynomial functions (degree = 1,2,3) to the data.