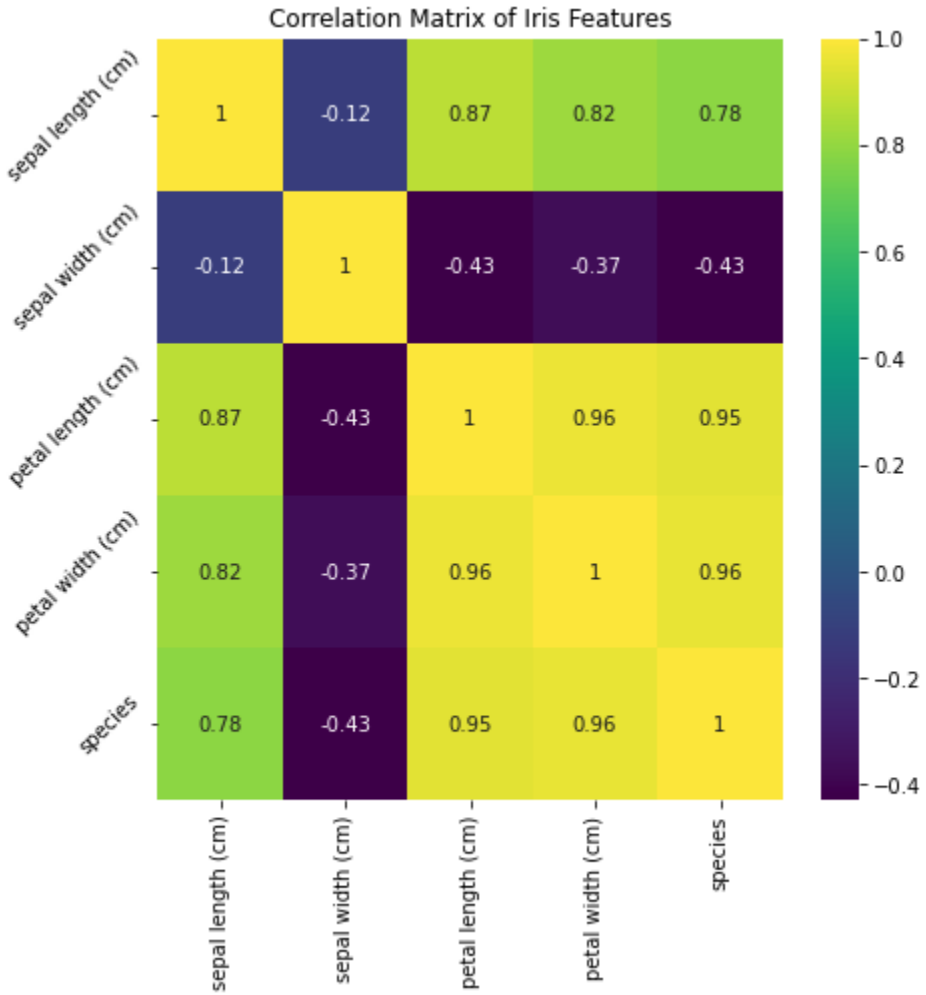
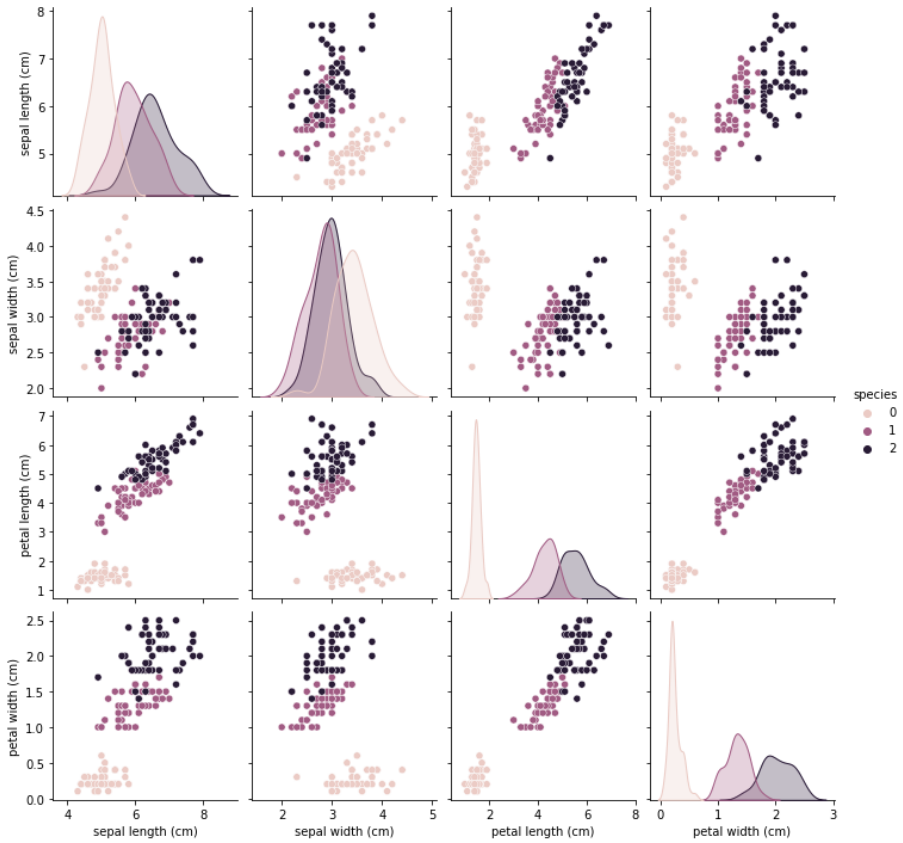
1.a)

i) Ans


Correlation Matrix of Iris Features
sepal length (cm)
sepal width (cm)
petal length (cm)
petal width (cm)
species
1
-0.12
0.87
0.82
0.78
-0.12
1
-0.43
-0.37
-0.43
0.87
-0.43
1
0.96
0.95
0.82
-0.37
0.96
1
0.96
0.78
-0.43
0.95
0.96
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

ii) Ans



1b)

A Ans.

In heatmap visualization, we can see the correlation matrix of the Iris features, including the species. The heatmap shows the strength and direction of the linear relationship between pairs of features. For example, we can see that there is a strong positive correlation between 'petal length' and 'petal width', which makes sense since longer petals tend to be wider.

In feature distribution visualization, we can see the distribution of each feature for each species using a pairplot. Each row and column corresponds to a different feature, and the diagonal shows the distribution of that feature for each species. We can see that the 'petal length' and 'petal width' features are good indicators of the species, while the 'sepal length' and 'sepal width' features are less distinct.

The diagonal plots in the illustration depict the distribution of iris-features according to species. The less effective a feature can be as a classifier, the more the plots overlap other features. For instance, we can observe that all three bells are more closely overlapping on the plot for sepal width distribution. There is greater separation in the distribution of classes for a trait the better it is at discriminating between different classes. Petal length, for instance, can be used to distinguish between species since the bell curves for each are in some way different. Between the species "1" and "2," there is still some crossover.

B Ans.

From the correlation matrix heatmap, we can see that 'petal length' and 'petal width' are highly positively correlated with a correlation coefficient of 0.96. This indicates that as the length of the petals increases, the width also tends to increase. Similarly, there is a high positive correlation between 'sepal length' and 'petal length' (0.87), as well as 'sepal length' and 'petal width' (0.82).

From the feature distribution analysis, we can see that 'petal length' and 'petal width' are good indicators of the species, with Setosa having the smallest petals and Virginica having the largest petals. In contrast, 'sepal length' and 'sepal width' are less distinct indicators of the species.

These inferences help us understand the relationships between the different features and how they can be used to classify the different species of Iris. We can also use this information to select the most important features for our analysis and avoid redundant features. Furthermore, these inferences can help us identify outliers and anomalies in the data that may affect our analysis.

In addition to the strong positive correlations discussed earlier, the correlation matrix heatmap also shows weak and negative correlations between some features. For example, there is a weak negative correlation between 'sepal width' and 'petal length' (-0.37), which indicates that as the sepal width increases, the petal length tends to decrease slightly. Similarly, there is a weak negative correlation between 'sepal width' and 'petal width' (-0.37).

These weak and negative correlations suggest that these features may not be as important in distinguishing between the species of Iris. However, it is important to note that correlation coefficients alone cannot fully determine the importance of a feature, as there may be other complex relationships between the features and the target variable that are not captured by the correlation analysis. Therefore, it is important to use other feature selection techniques and data analysis methods to fully understand the relationship between the features and the target variable.

2.a) Ans

```
Case I: 30% samples
RMSE is : 0.33421561706995867
Intercept is :  0.8557618168628296
coefficient is:  [ 0.62339783 -0.80150855  1.11422801  0.34931187]
**********************************************************
Case II: 70% samples
RMSE is : 0.32455495722112165
Intercept is :  -0.2725185759273132
coefficient is:  [ 0.67760814 -0.53212752  1.00051066  0.50420577]
```

From the given results, we can observe that both cases perform similarly in terms of RMSE. The RMSE for case II (70% training data) is slightly lower than that of case I (30% training data), which suggests that case II provides a slightly better fit to the data. Comparing the LR parameters, we can see that the intercept and coefficient values are different between the two cases. This is expected since the LR model is trained on different subsets of the data in each case. However, the coefficients for each feature (sepal length, sepal width, and petal width) have similar magnitudes in both cases, indicating that these features are important in predicting petal length.

Overall, while both cases provide reasonable predictions with similar RMSE values, we can conclude that case II (70% training data) performs slightly better in terms of RMSE and provides more robust parameter estimates since it is trained on a larger subset of the data. However, further analysis and experimentation may be required to validate these conclusions.

2b) Ans

```
Model:1
Predicted petal length for sample 100 : 5.622383704453052
Actual Value:  6.0
Rmse for prediction of train size 0.3 is 0.3776162955469484
Intercept is :  0.8557618168628296
coefficient is:  [ 0.62339783 -0.80150855  1.11422801  0.34931187]
**********************************************************************
Model:2
Predicted petal length for sample 100 : 5.75008009992904
Actual Value:  6.0
Rmse for prediction of train size 0.7 is 0.2499199000709602
Intercept is :  -0.2725185759273132
coefficient is:  [ 0.67760814 -0.53212752  1.00051066  0.50420577]
```

2c) Ans Based on the RMSE values, Model 2 appears to perform better than Model 1 as it has a lower RMSE value for the larger train size of 0.7. This suggests that Model 2 is better at predicting petal length on unseen data. Additionally, the intercept and coefficients of Model 2 seem to have a higher magnitude compared to those of Model 1. This could indicate that Model 2 is more robust and has a stronger relationship between the predictor variables and the target variable.

Overall, based on the given evaluation parameters, Model 2 appears to perform better than Model 1. However, it's important to note that these evaluation parameters are not sufficient to make a final decision on which model is better, and further analysis and testing may be required.

Appendix:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline
from sklearn.cluster import KMeans
import matplotlib.patches as mpatches
import sklearn.metrics as metrics
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.datasets import load_iris
```

```
#loading iris data
iris=load_iris()

#creating pd dataframes
x=pd.DataFrame(data=iris.data,columns=iris.feature_names)
y=pd.DataFrame(data=iris.target,columns=['species'])
df=pd.concat([x,y],axis=1)
#visualize iris features as a heatmap
plt.figure(figsize= (7,7))
col = ['sepal length','sepal width','petal length','petal width', 'Species']
sns.heatmap(data = df.corr(), annot = True, cmap = 'viridis')
plt.xticks(fontsize = 10, rotation = 90)
plt.yticks(fontsize = 10, rotation = 45)
plt.title('Correlation Matrix of Iris Features')
plt.show()
```

```
#iris feature analysis:
g=sns.pairplot(df,hue='species')

data = pd.DataFrame(load_iris().data,columns=["sepal length","sepal width","petal length","petal width"])
y = pd.DataFrame(load_iris().target,columns=["species"])
df= pd.concat([data,y],axis=1)
X=df.drop(columns=['petal length'])
Y=df['petal length']
```

```
X=df.drop(columns=['petal length'], axis=1)
Y=df['petal length']
```

```
# Split the dataset into training and testing sets (using 30% and 70% of the samples for training)
X_train_30, X_test_30, y_train_30, y_test_30 = train_test_split(X, Y, train_size=0.3, random_state=111)
X_train_70, X_test_70, y_train_70, y_test_70 = train_test_split(X, Y, train_size=0.7, random_state=111)

# Train a Linear Regression model on the 30% training set
lr_model_30 = LinearRegression()
lr_model_30.fit(X_train_30, y_train_30)
lr_model_30.predict(X_test_30)
Y_pred=lr_model_30.predict(X_test_30)
rmse = np.sqrt(mean_squared_error(y_test_30, Y_pred))
print("Case I: 30% samples")
print("RMSE is :",rmse)
print("Intercept is : ",lr_model_30.intercept_)
print("coefficient is: ",lr_model_30.coef_)

print("**********************************************************")

# Train a Linear Regression model on the 30% training set
lr_model_70 = LinearRegression()
lr_model_70.fit(X_train_70, y_train_70)
lr_model_70.predict(X_test_70)
Y_pred=lr_model_70.predict(X_test_70)
rmse = np.sqrt(mean_squared_error(y_test_70, Y_pred))
print("Case II: 70% samples")
print("RMSE is :",rmse)
print("Intercept is : ",lr_model_70.intercept_)
print("coefficient is: ",lr_model_70.coef_)
```

```
#create new data point for prediction
index = set(X_test_70.index.tolist()).intersection(X_test_30.index.tolist())
sample_index = 100
test_X = pd.DataFrame(X, index=[sample_index])
test_y = pd.DataFrame(Y, index=[sample_index])

y_pred_sample = lr_model_30.predict(test_X)
print("Model:1")
print("Predicted petal length for sample", sample_index, ":", y_pred_sample[0])
print("Actual Value: ",test_y.iloc[0]['petal length'])
print(f"Rmse for prediction of train size {0.3} is",np.sqrt(mean_squared_error(test_y, y_pred_sample)))

print("Intercept is : ",lr_model_30.intercept_)
print("coefficient is: ",lr_model_30.coef_)

print("**********************************************************************")

#create new data point for prediction
index = set(X_test_70.index.tolist()).intersection(X_test_30.index.tolist())
test_X = pd.DataFrame(X, index=[sample_index])
test_y = pd.DataFrame(Y, index=[sample_index])

y_pred_sample = lr_model_70.predict(test_X)
print("Model:2")
print("Predicted petal length for sample", sample_index, ":", y_pred_sample[0])
print("Actual Value: ",test_y.iloc[0]['petal length'])
print(f"Rmse for prediction of train size {0.7} is",np.sqrt(mean_squared_error(test_y, y_pred_sample)))
print("Intercept is : ",lr_model_70.intercept_)
print("coefficient is: ",lr_model_70.coef_)
```