Logistic Regression Project

The Data

We will be analyzing the famous Iris flower data set.(http://en.wikipedia.org/wiki/Iris_flower_data_set (http://en.wikipedia.org/wiki/Iris_flower_data_set)).

The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor), so 150 total samples. Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters.

The iris dataset contains measurements for 150 iris flowers from three different species.

The three classes in the Iris dataset:

```
Iris-setosa (n=50)
Iris-versicolor (n=50)
Iris-virginica (n=50)
```

The four features of the Iris dataset:

```
sepal length in cm
sepal width in cm
petal length in cm
petal width in cm
```

Get the data

Use seaborn to get the iris data by using: iris = sns.load_dataset('iris')

In [55]:
```
import seaborn as sns
iris = sns.load_dataset('iris')
iris.head()
```

Out[55]:

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Exploratory Data Analysis

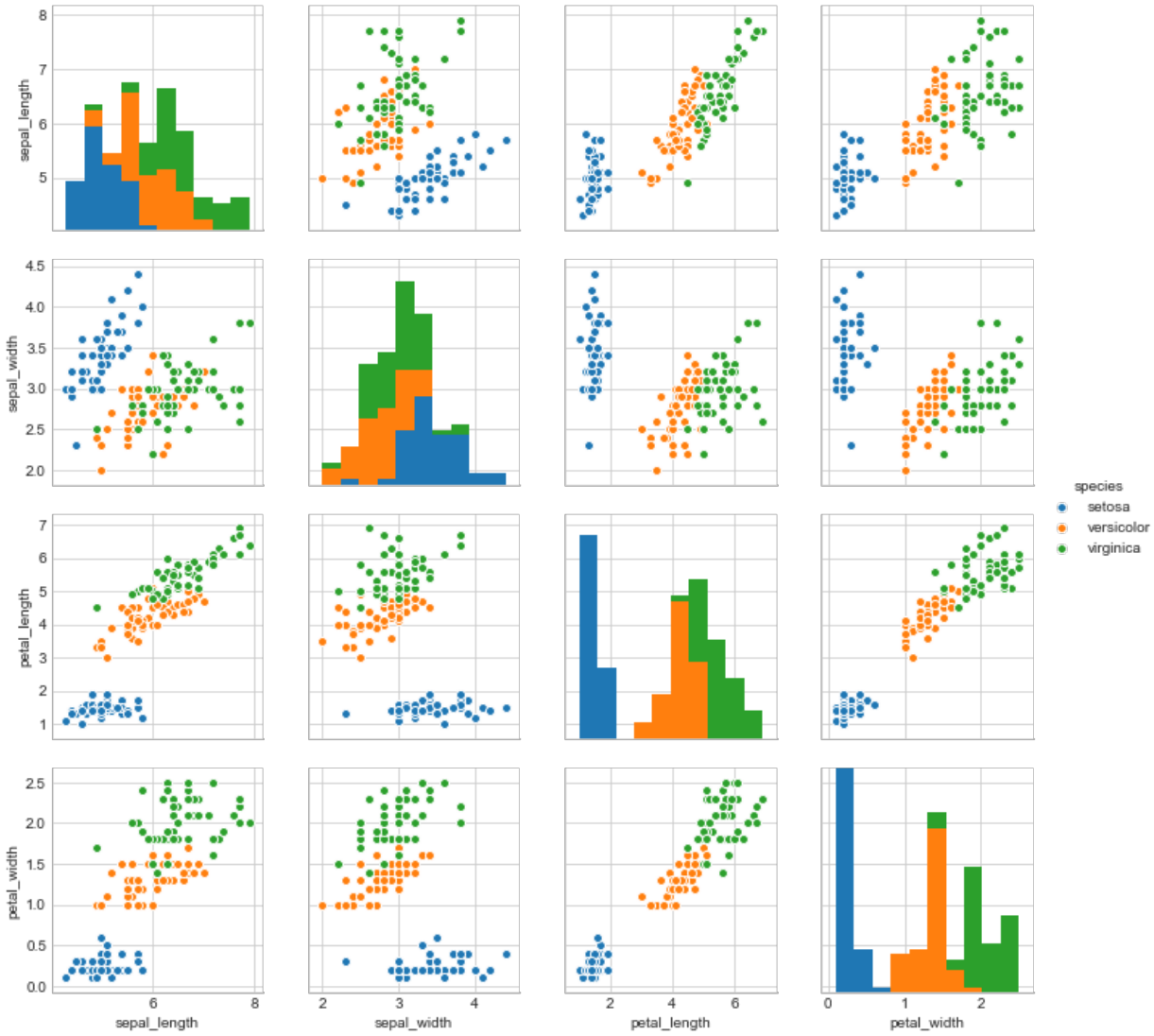
Import libraries needed for data analysis and visualizations.

In [56]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Create a pairplot of the data set.

In [29]:
```
sns.pairplot(iris, hue = 'species')
```

```
Out[29]: <seaborn.axisgrid.PairGrid at 0x1a25667908>
```



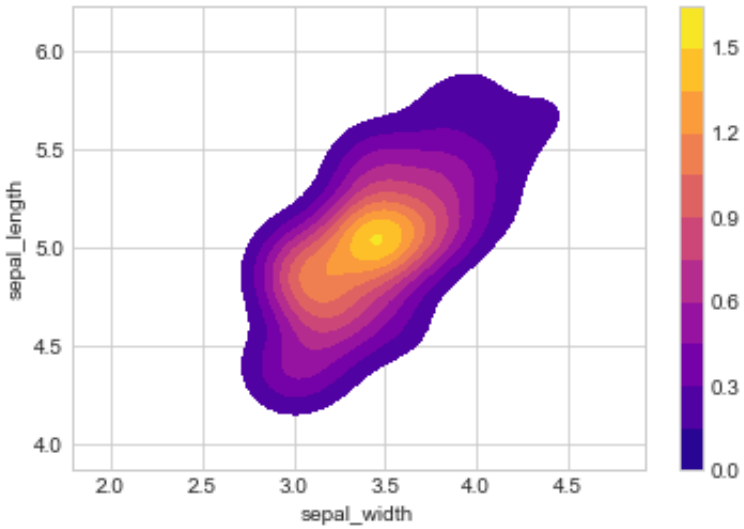
Observation: Setosa seems to be the most separable species.

Create a kde plot of sepal_length versus sepal width for setosa species of flower.

In [59]:
```
sns.set_style('whitegrid')
iris_setosa = iris[iris['species'] == 'setosa' ]
sns.kdeplot(iris_setosa['sepal_width'],
            iris_setosa['sepal_length'],
            cmap = 'plasma', shade_lowest = False,
            shade = True, cbar = 'seismic')
```

```
/Users/Jayashri/anaconda/lib/python3.6/site-packages/scipy/stats/sta
ts.py:1713: FutureWarning: Using a non-tuple sequence for multidimen
sional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr
[seq]`. In the future this will be interpreted as an array index, `a
rr[np.array(seq)]`, which will result either in an error or a differ
ent result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumva
l
```

Out[59]: <matplotlib.axes._subplots.AxesSubplot at 0x1a27801940>



Train Test Split

Split the data into a training set and a testing set. Use test_size = 30%.

In [60]:
```
iris.columns
```

Out[60]:
```
Index(['sepal_length', 'sepal_width', 'petal_length', 'petal_width',
       'species'],
      dtype='object')
```

In [61]:
```
X = iris.drop('species', axis = 1)
y = iris['species']

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3,
                                                    random_state=101)
```

Train the Model

Training and Predicting

In [63]:
```
from sklearn.linear_model import LogisticRegression
```

In [64]:
```
logmodel = LogisticRegression()
```

In [65]:
```
logmodel.fit(X_train, y_train)
```

Out[65]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_interce
pt=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jo
bs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0
.0001,
          verbose=0, warm_start=False)
```

In [66]:
```
predictions = logmodel.predict(X_test)
```

Evaluation

In [ ]:
```
from sklearn.metrics import classification_report, confusion_matrix
```

In [53]:
```
print(confusion_matrix(y_test, predictions))
```

```
[[13  0  0]
 [ 0 18  2]
 [ 0  0 12]]
```

In [54]:
```
print(classification_report(y_test, predictions))
```

```
             precision    recall  f1-score   support

     setosa       1.00      1.00      1.00        13
 versicolor       1.00      0.90      0.95        20
  virginica       0.86      1.00      0.92        12

avg / total       0.96      0.96      0.96        45
```

Observation: The model seems to be identify setosa well as it is distinct from the other two species. On the whole the model seems to be a good fit.