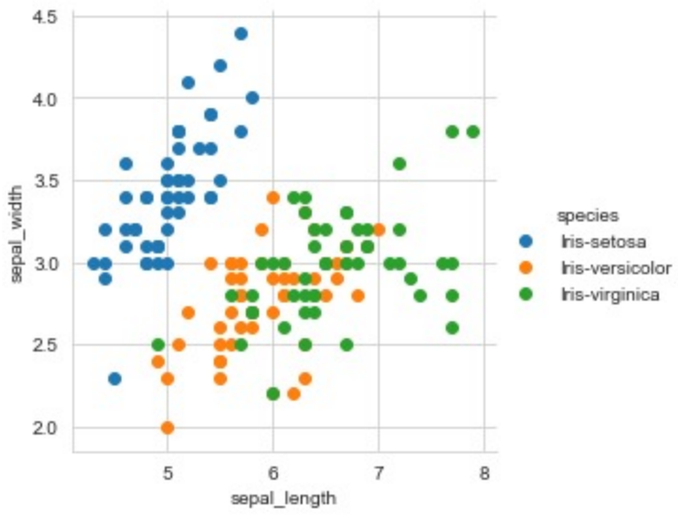
Data Preprocessing

In [1]:
```
#importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import warnings



# ignore all warnings
warnings.filterwarnings('ignore')
```

In [2]:
```
#import dataset
dataset = pd.read_csv('iris.data')
```

Visualizations

In [3]:
```
sns.set_style("whitegrid");
sns.FacetGrid(dataset, hue="species", size=4) \
   .map(plt.scatter, "sepal_length", "sepal_width") \
   .add_legend();
```



In [4]:
```
#pairplot
sns.set_style("whitegrid");
sns.pairplot(dataset, hue="species", size=3);
```

Splitting the data into test and train

In [5]:
```
from sklearn.model_selection import train_test_split
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state =452 )
```

Applying K Nearest Neighbourhood ¶

In [6]:
```
#Using KNeighborsClassifier Method of neighbors class to use Nearest Neighbor algorithm
from sklearn.neighbors import KNeighborsClassifier
classifier = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

# Importing metrics for evaluation
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report

# Summary of the predictions made by the classifier
print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

# Accuracy score
from sklearn.metrics import accuracy_score
print('\nAccuracy is :-',accuracy_score(y_pred,y_test))
```

```
                precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        10
Iris-versicolor       0.92      1.00      0.96        11
 Iris-virginica       1.00      0.89      0.94         9

      micro avg       0.97      0.97      0.97        30
      macro avg       0.97      0.96      0.97        30
   weighted avg       0.97      0.97      0.97        30

[[10  0  0]
 [ 0 11  0]
 [ 0  1  8]]

Accuracy is :- 0.9666666666666667
```