Payal Patel; Big Data HW Assignment; Random Forest with Iris Dataset (importing/cleaning data, data exploration, model building and results)

In [1]:
```
spark.version
```

```
Starting Spark application
```

| ID | YARN Application ID | Kind | State | |
|---|---|---|---|---|
| 0 | application_1581123385763_0001 | pyspark | idle | Link (http://143.ec2.internal:20888/proxy/application_158112338 |

```
SparkSession available as 'spark'.
```

```
'2.4.4'
```

In [2]:
```
#view loaded libraries/packages
sc.list_packages()
```

```
Package                    Version
-------------------------- -------
beautifulsoup4             4.8.1
boto                       2.49.0
jmespath                   0.9.4
lxml                       4.4.2
mysqlclient                1.4.6
nltk                       3.4.5
nose                       1.3.4
numpy                      1.14.5
pip                        20.0.2
py-dateutil                2.2
python36-sagemaker-pyspark 1.2.6
pytz                       2019.3
PyYAML                     3.11
setuptools                 45.1.0
six                        1.13.0
soupsieve                  1.9.5
wheel                      0.34.2
windmill                   1.6
```

In [3]:
```
#install libraries/packages
sc.install_pypi_package("pandas==0.25.1") #Install pandas version 0.25.1
sc.install_pypi_package("matplotlib", "https://pypi.org/simple")
sc.install_pypi_package("sklearn")
sc.install_pypi_package("seaborn")
```

```
Collecting pandas==0.25.1
  Downloading pandas-0.25.1-cp36-cp36m-manylinux1_x86_64.whl (10.5 MB)
Collecting python-dateutil>=2.6.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib64/python3.6/si
te-packages (from pandas==0.25.1) (1.14.5)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/site-
packages (from pandas==0.25.1) (2019.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/site-pack
ages (from python-dateutil>=2.6.1->pandas==0.25.1) (1.13.0)
Installing collected packages: python-dateutil, pandas
Successfully installed pandas-0.25.1 python-dateutil-2.8.1

Collecting matplotlib
  Downloading matplotlib-3.1.3-cp36-cp36m-manylinux1_x86_64.whl (13.1 MB)
Collecting pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1
  Downloading pyparsing-2.4.6-py2.py3-none-any.whl (67 kB)
Requirement already satisfied: numpy>=1.11 in /usr/local/lib64/python3.6/site
-packages (from matplotlib) (1.14.5)
Requirement already satisfied: python-dateutil>=2.1 in /mnt/tmp/1581123628188
-0/lib/python3.6/site-packages (from matplotlib) (2.8.1)
Collecting kiwisolver>=1.0.1
  Downloading kiwisolver-1.1.0-cp36-cp36m-manylinux1_x86_64.whl (90 kB)
Collecting cycler>=0.10
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/site-pack
ages (from python-dateutil>=2.1->matplotlib) (1.13.0)
Requirement already satisfied: setuptools in /mnt/tmp/1581123628188-0/lib/pyt
hon3.6/site-packages (from kiwisolver>=1.0.1->matplotlib) (45.1.0)
Installing collected packages: pyparsing, kiwisolver, cycler, matplotlib
Successfully installed cycler-0.10.0 kiwisolver-1.1.0 matplotlib-3.1.3 pypars
ing-2.4.6

Collecting sklearn
  Downloading sklearn-0.0.tar.gz (1.1 kB)
Collecting scikit-learn
  Downloading scikit_learn-0.22.1-cp36-cp36m-manylinux1_x86_64.whl (7.0 MB)
Collecting scipy>=0.17.0
  Downloading scipy-1.4.1-cp36-cp36m-manylinux1_x86_64.whl (26.1 MB)
Collecting joblib>=0.11
  Downloading joblib-0.14.1-py2.py3-none-any.whl (294 kB)
Requirement already satisfied: numpy>=1.11.0 in /usr/local/lib64/python3.6/si
te-packages (from scikit-learn->sklearn) (1.14.5)
Building wheels for collected packages: sklearn
  Building wheel for sklearn (setup.py): started
  Building wheel for sklearn (setup.py): finished with status 'done'
  Created wheel for sklearn: filename=sklearn-0.0-py2.py3-none-any.whl size=1
315 sha256=68d3feef3ce72277f4982e14c6618f589136830254aa28173ec42a96e86d4b59
  Stored in directory: /mnt/var/lib/livy/.cache/pip/wheels/23/9d/42/5ec745cbb
b17517000a53cecc49d6a865450d1f5cb16dc8a9c
Successfully built sklearn
Installing collected packages: scipy, joblib, scikit-learn, sklearn
Successfully installed joblib-0.14.1 scikit-learn-0.22.1 scipy-1.4.1 sklearn-
0.0

Collecting seaborn
  Downloading seaborn-0.10.0-py3-none-any.whl (215 kB)
```

```
Requirement already satisfied: scipy>=1.0.1 in /mnt/tmp/1581123628188-0/lib6
4/python3.6/site-packages (from seaborn) (1.4.1)
Requirement already satisfied: matplotlib>=2.1.2 in /mnt/tmp/1581123628188-0/
lib64/python3.6/site-packages (from seaborn) (3.1.3)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib64/python3.6/si
te-packages (from seaborn) (1.14.5)
Requirement already satisfied: pandas>=0.22.0 in /mnt/tmp/1581123628188-0/lib
64/python3.6/site-packages (from seaborn) (0.25.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /m
nt/tmp/1581123628188-0/lib/python3.6/site-packages (from matplotlib>=2.1.2->s
eaborn) (2.4.6)
Requirement already satisfied: python-dateutil>=2.1 in /mnt/tmp/1581123628188
-0/lib/python3.6/site-packages (from matplotlib>=2.1.2->seaborn) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /mnt/tmp/1581123628188-0/
lib64/python3.6/site-packages (from matplotlib>=2.1.2->seaborn) (1.1.0)
Requirement already satisfied: cycler>=0.10 in /mnt/tmp/1581123628188-0/lib/p
ython3.6/site-packages (from matplotlib>=2.1.2->seaborn) (0.10.0)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.6/site-
packages (from pandas>=0.22.0->seaborn) (2019.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.6/site-pack
ages (from python-dateutil>=2.1->matplotlib>=2.1.2->seaborn) (1.13.0)
Requirement already satisfied: setuptools in /mnt/tmp/1581123628188-0/lib/pyt
hon3.6/site-packages (from kiwisolver>=1.0.1->matplotlib>=2.1.2->seaborn) (4
5.1.0)
Installing collected packages: seaborn
Successfully installed seaborn-0.10.0
```

In [4]:
```
#Load libraries/packages
from pyspark.sql.types import *
from pyspark.sql.functions import monotonically_increasing_id, col, expr, when
, concat, lit, isnan
from pyspark.ml.linalg import Vectors
from pyspark.ml.regression import GeneralizedLinearRegression
from pyspark.ml.classification import RandomForestClassifier, LogisticRegressi
on
from pyspark.ml.feature import VectorIndexer, VectorAssembler, StringIndexer,
OneHotEncoder
from pyspark.ml.evaluation import MulticlassClassificationEvaluator, Regressio
nEvaluator, BinaryClassificationEvaluator
from pyspark.ml import Pipeline
from pyspark.ml.clustering import KMeans
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pyspark.sql.types import IntegerType
from pyspark.sql.functions import udf
```

In [5]:
```
#Load dataset
iris  = spark.read.load("s3a://ppatel93/Iris.csv", "csv", delimiter=",", infer
Schema=True, header=True)
iris.createOrReplaceTempView("iris")
```

In [6]:
```
#schema / dataset information
print('\r\nTotal Records: ' + str(iris.count()) + '\r\n\r\n')
iris.printSchema()
```

```
Total Records: 150


root
 |-- Id: integer (nullable = true)
 |-- SepalLengthCm: double (nullable = true)
 |-- SepalWidthCm: double (nullable = true)
 |-- PetalLengthCm: double (nullable = true)
 |-- PetalWidthCm: double (nullable = true)
 |-- Species: string (nullable = true)
```

In [7]:
```
#View Dataset (first five records)
sqlDF = spark.sql("select * from iris").show(5)
```

```
+---+-------------+------------+-------------+------------+-----------+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|    Species|
+---+-------------+------------+-------------+------------+-----------+
|  1|          5.1|         3.5|          1.4|         0.2|Iris-setosa|
|  2|          4.9|         3.0|          1.4|         0.2|Iris-setosa|
|  3|          4.7|         3.2|          1.3|         0.2|Iris-setosa|
|  4|          4.6|         3.1|          1.5|         0.2|Iris-setosa|
|  5|          5.0|         3.6|          1.4|         0.2|Iris-setosa|
+---+-------------+------------+-------------+------------+-----------+
only showing top 5 rows
```

In [8]:
```
#View distinct values in species column
sqlDF = spark.sql("select distinct(Species) from iris").show()
```

```
+---------------+
|        Species|
+---------------+
| Iris-virginica|
|    Iris-setosa|
|Iris-versicolor|
+---------------+
```

In [9]:
```
#transform categorical target variable to numeric
def func(item):
    if item == 'Iris-virginica':
        return 0
    if item == 'Iris-setosa':
        return 1
    return 2

func_udf = udf(func, IntegerType())
iris = iris.withColumn('Species2',func_udf(iris['Species']))
```

In [12]:
```
iris.show(5)
```

```
+---+-------------+------------+-------------+------------+-----------+------
--+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|    Species|Specie
s2|
+---+-------------+------------+-------------+------------+-----------+------
--+
|  1|          5.1|         3.5|          1.4|         0.2|Iris-setosa|
1|
|  2|          4.9|         3.0|          1.4|         0.2|Iris-setosa|
1|
|  3|          4.7|         3.2|          1.3|         0.2|Iris-setosa|
1|
|  4|          4.6|         3.1|          1.5|         0.2|Iris-setosa|
1|
|  5|          5.0|         3.6|          1.4|         0.2|Iris-setosa|
1|
+---+-------------+------------+-------------+------------+-----------+------
--+
only showing top 5 rows
```

In [13]:
```
#split into training and test datasets
train_df, test_df = iris.randomSplit([0.7, 0.3])
train_df.show(5)
test_df.show(5)
```

```
+---+-------------+------------+-------------+------------+-----------+------
--+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|    Species|Specie
s2|
+---+-------------+------------+-------------+------------+-----------+------
--+
|  1|          5.1|         3.5|          1.4|         0.2|Iris-setosa|
1|
|  2|          4.9|         3.0|          1.4|         0.2|Iris-setosa|
1|
|  3|          4.7|         3.2|          1.3|         0.2|Iris-setosa|
1|
|  6|          5.4|         3.9|          1.7|         0.4|Iris-setosa|
1|
|  7|          4.6|         3.4|          1.4|         0.3|Iris-setosa|
1|
+---+-------------+------------+-------------+------------+-----------+------
--+
only showing top 5 rows

+---+-------------+------------+-------------+------------+-----------+------
--+
| Id|SepalLengthCm|SepalWidthCm|PetalLengthCm|PetalWidthCm|    Species|Specie
s2|
+---+-------------+------------+-------------+------------+-----------+------
--+
|  4|          4.6|         3.1|          1.5|         0.2|Iris-setosa|
1|
|  5|          5.0|         3.6|          1.4|         0.2|Iris-setosa|
1|
| 11|          5.4|         3.7|          1.5|         0.2|Iris-setosa|
1|
| 20|          5.1|         3.8|          1.5|         0.3|Iris-setosa|
1|
| 22|          5.1|         3.7|          1.5|         0.4|Iris-setosa|
1|
+---+-------------+------------+-------------+------------+-----------+------
--+
only showing top 5 rows
```

## Exploratory Data Analysis

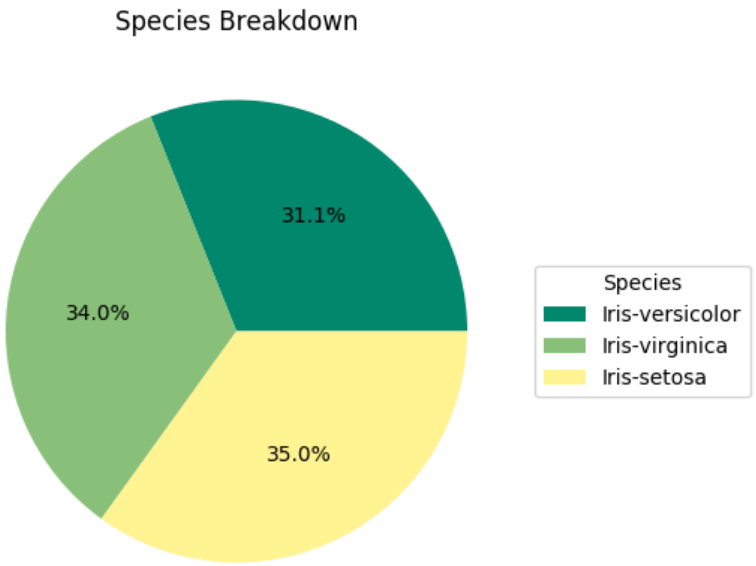
In [14]:
```
#summary stats for numeric/continuous variables in training dataset
train_df.describe(['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWid
thCm']).show()
```

```
+-------+------------------+-------------------+------------------+----------
--------+
|summary|     SepalLengthCm|       SepalWidthCm|     PetalLengthCm|      Peta
lWidthCm|
+-------+------------------+-------------------+------------------+----------
--------+
|  count|               103|                103|               103|
103|
|   mean| 5.798058252427182|  3.055339805825244| 3.708737864077669|1.18446601
94174756|
| stddev|0.8525602252919483|0.43334724457454915|1.7979349154023465|0.77545145
37785928|
|    min|               4.3|                2.0|               1.1|
0.1|
|    max|               7.9|                4.4|               6.9|
2.5|
+-------+------------------+-------------------+------------------+----------
--------+
```

In [15]:
```
#explore target variable
train_df.groupBy("Species").count().show()
```

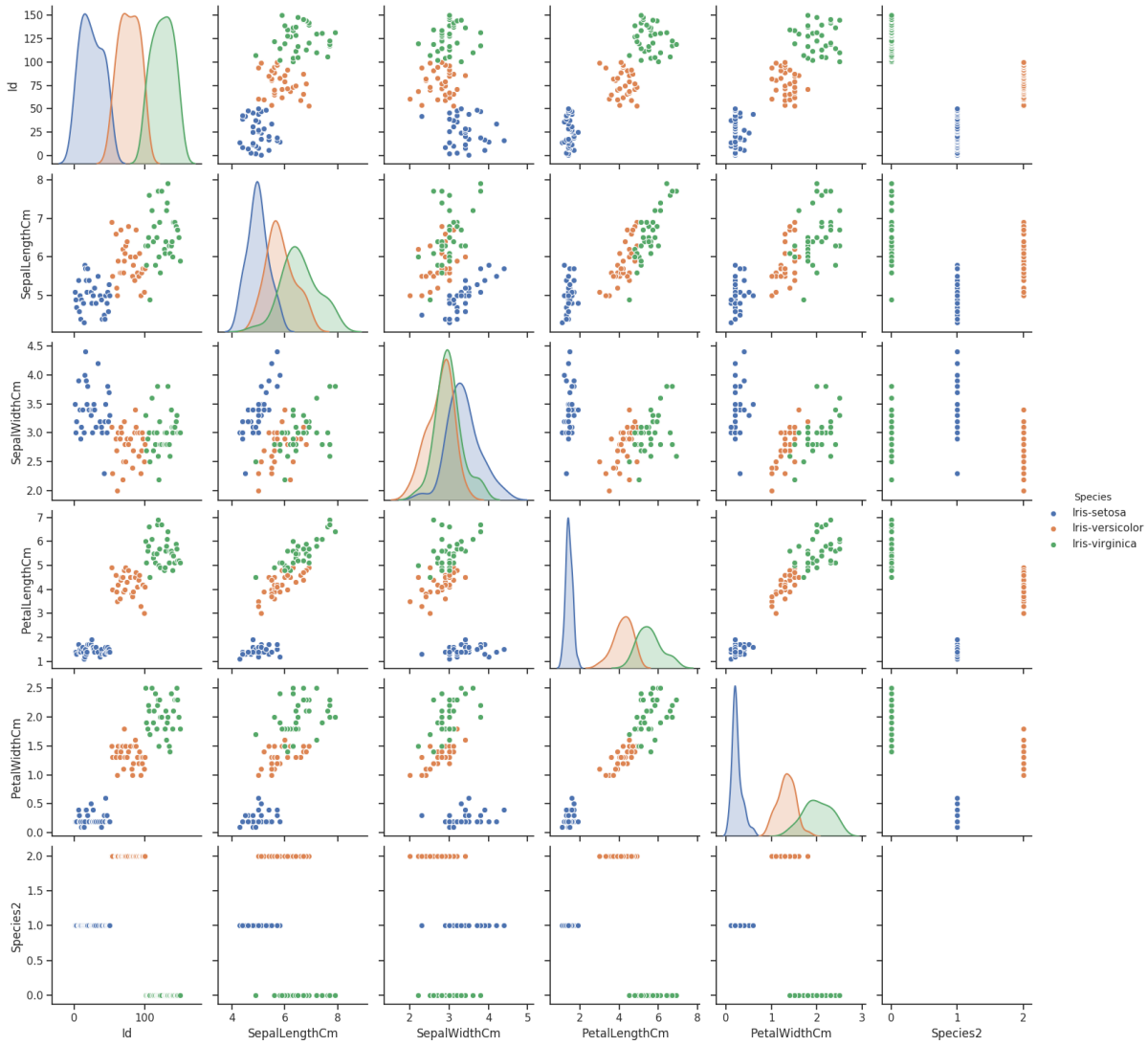
```
+---------------+-----+
|        Species|count|
+---------------+-----+
| Iris-virginica|   35|
|    Iris-setosa|   36|
|Iris-versicolor|   32|
+---------------+-----+
```

In [16]:
```
#Species breakdown in training dataset
species_dist = train_df.groupBy('Species').count().orderBy('count').toPandas()
plt.clf()
labels = [f"{species}" for species in species_dist['Species']]
obs = [num_obs for num_obs in species_dist['count']]
colors = ['#00876c', '#89c079', '#fff392']
fig, ax = plt.subplots(figsize=(8,5))
w,a,b = ax.pie(obs, autopct='%1.1f%%', colors=colors)
plt.title('Species Breakdown')
ax.legend(w, labels, title="Species", loc="center left", bbox_to_anchor=(1, 0,
0.5, 1))
%matplot plt
```



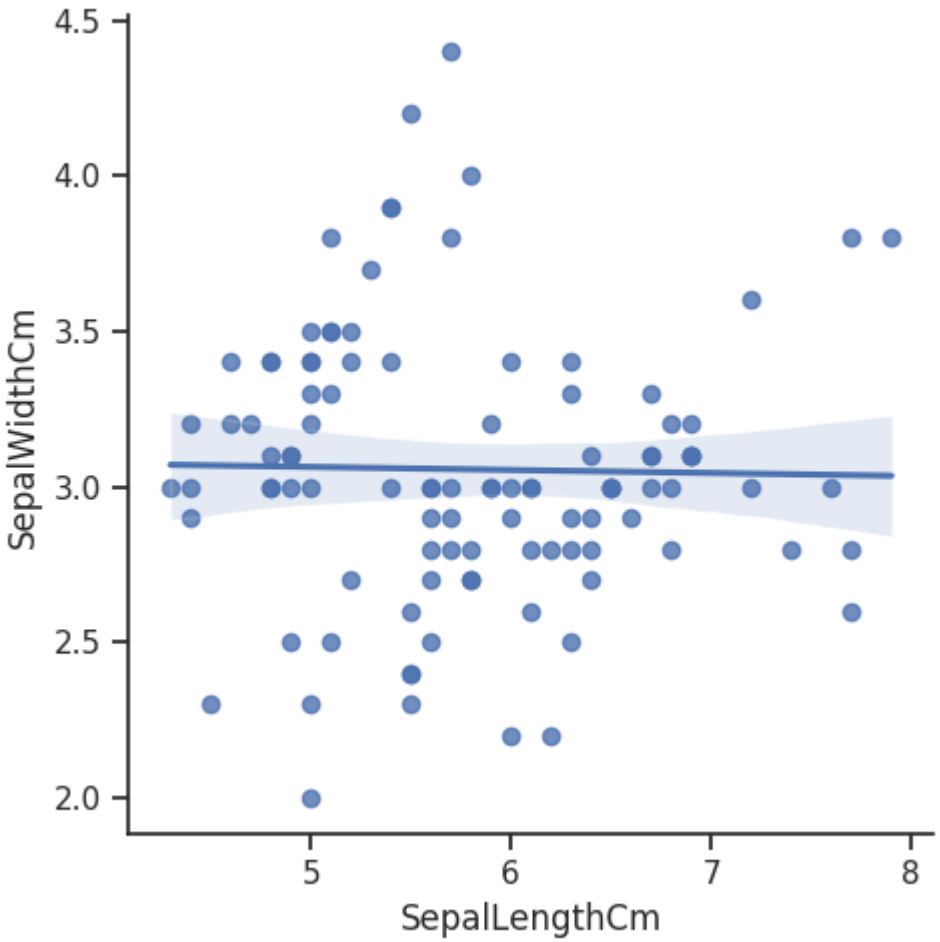
In [17]:
```
#scatterplots, using training data --creating for all variable/column combinat
ions, so not all are helpful
train_df_pandas = train_df.select("*").toPandas()
sns.set(style="ticks")
sns.pairplot(train_df_pandas, hue="Species")
plt.show()

%matplot plt
```

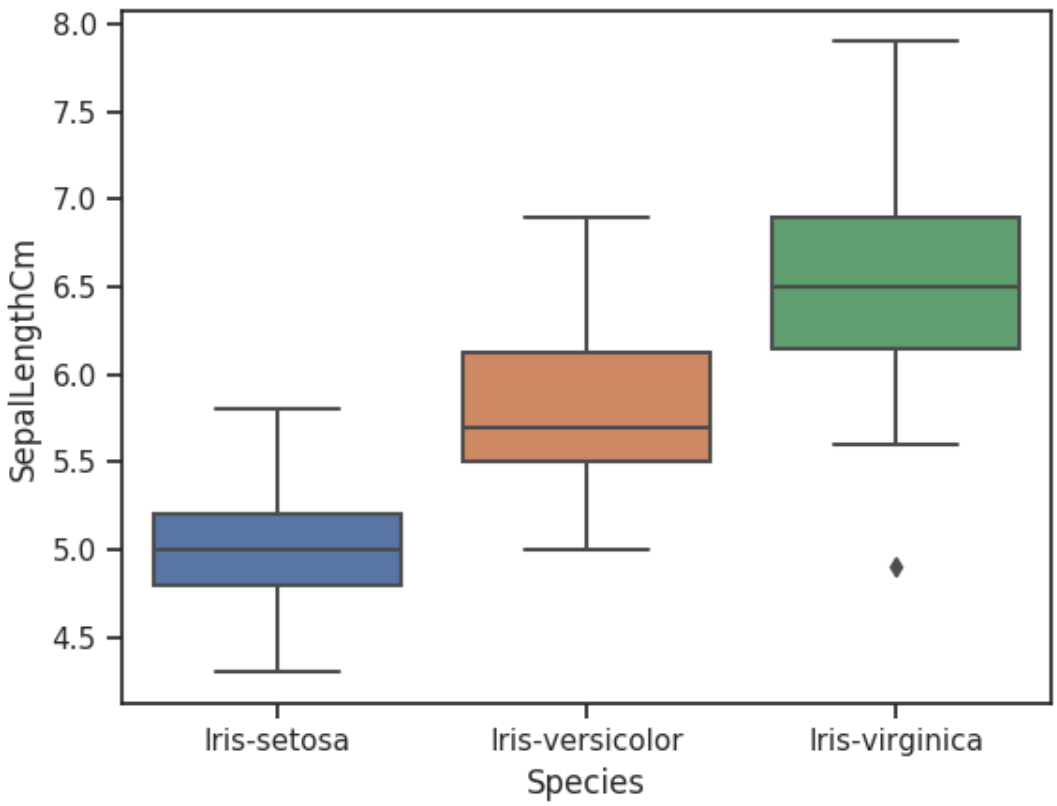



File failed to load: /extensions/MathZoom.js

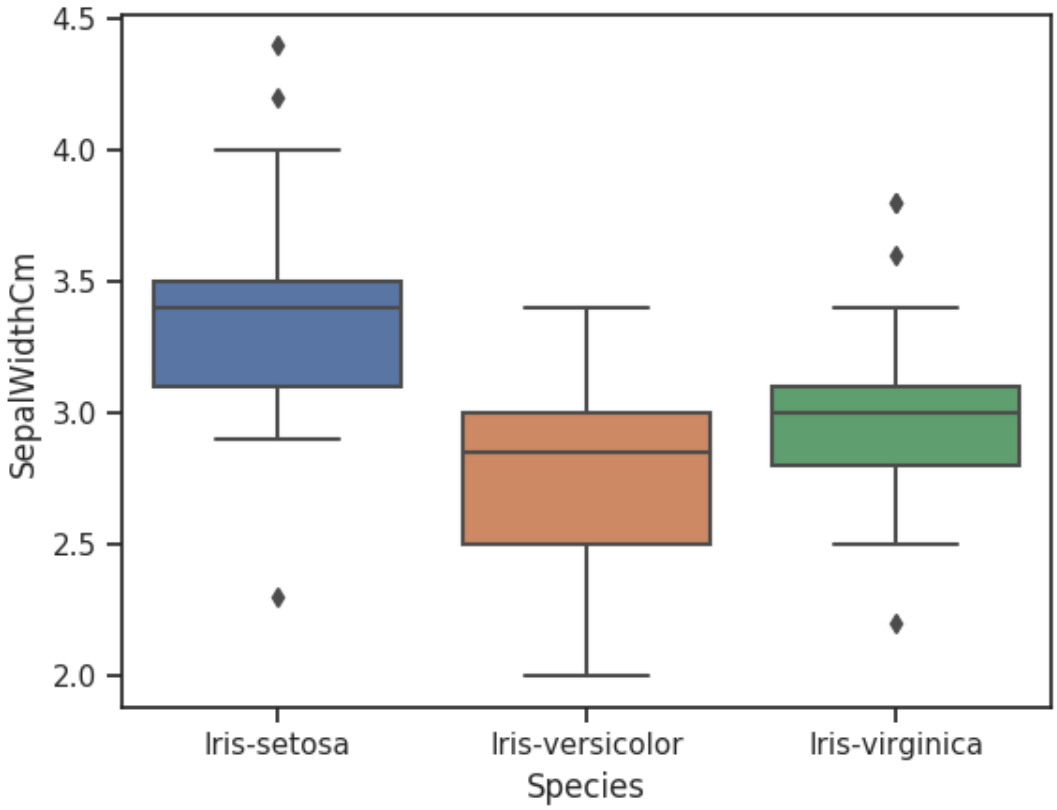
In [18]:
```
sns.lmplot(x='SepalLengthCm', y='SepalWidthCm', data=train_df_pandas)
%matplot plt
```



In [19]:
```
cont_var = ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']
plt.figure(7)
sns.boxplot(data=train_df_pandas, x='Species', y='SepalLengthCm')
%matplot plt
```

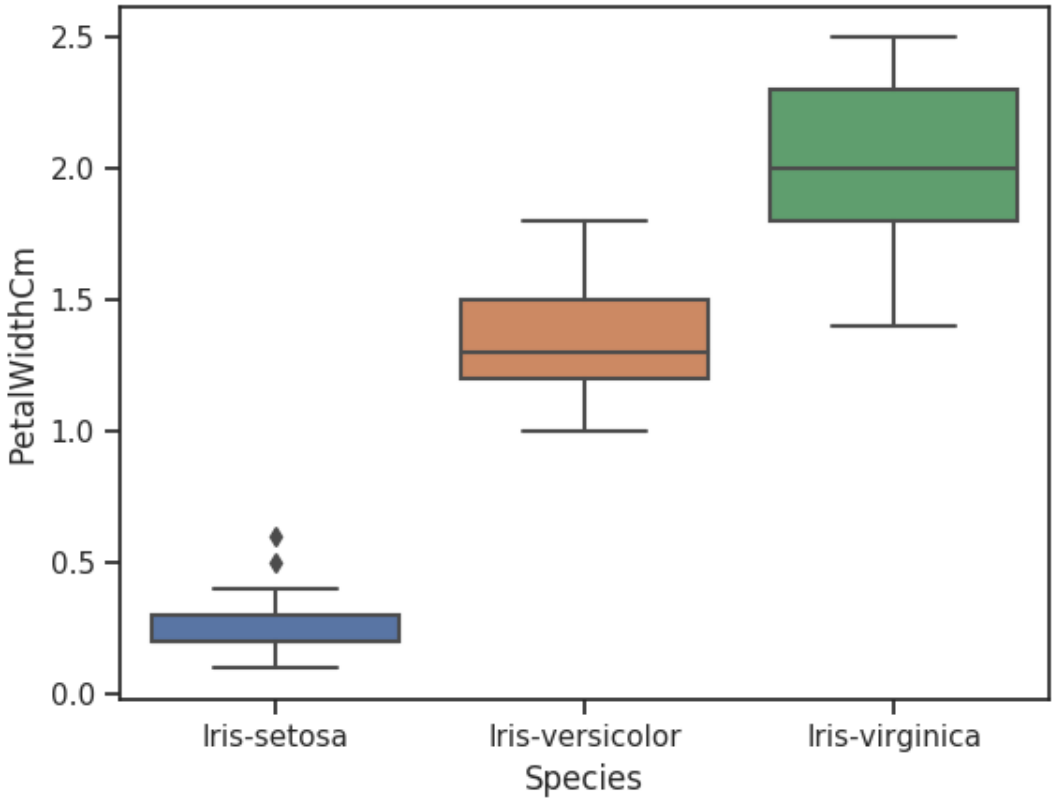


In [20]:
```
plt.figure(8)
sns.boxplot(data=train_df_pandas, x='Species', y='SepalWidthCm')
%matplot plt
```

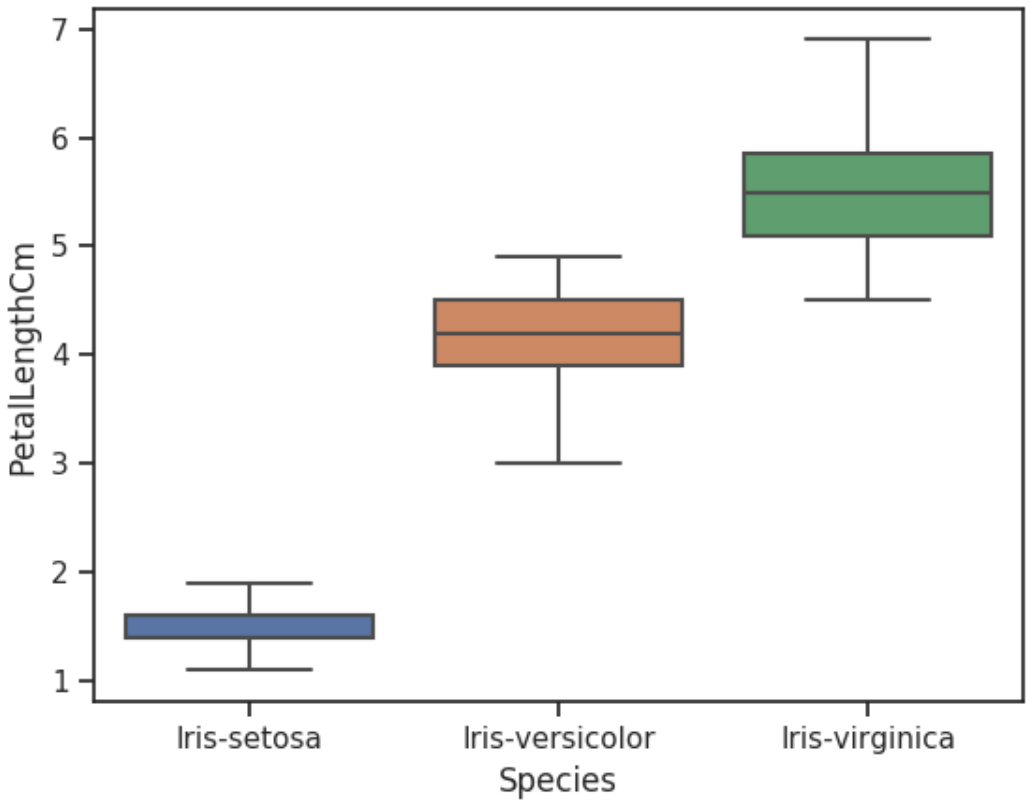



File failed to load: /extensions/MathZoom.js

In [22]:
```
plt.figure(10)
sns.boxplot(data=train_df_pandas, x='Species', y='PetalWidthCm')
%matplot plt
```



In [21]:
```
plt.figure(9)
sns.boxplot(data=train_df_pandas, x='Species', y='PetalLengthCm')
%matplot plt
```




Model Building (Random Forest)

In [23]:
```
#Model Preparation
col_target   =  'Species2'
col_features =  ['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidth
Cm']
# Use a VectorAssembler to combine all the feature columns into a single vecto
r column.
va  = VectorAssembler(inputCols=col_features, outputCol="features")
#Since we will have more than 1 stage of feature transformations, we use a Pip
eline to tie the stages together.
rfc = RandomForestClassifier(featuresCol="features", labelCol=col_target, pred
ictionCol="prediction", probabilityCol="probability", numTrees=25, maxDepth=5,
maxBins=32, seed=12345)
pipeline = Pipeline(stages=[va,rfc])
```

In [24]:
```
model = pipeline.fit(train_df)

predictions = model.transform(test_df)
predictions.createOrReplaceTempView("predictions")
```

In [25]:
```
# Multiclass Evaluator
mc_evaluator = MulticlassClassificationEvaluator(labelCol=col_target, predicti
onCol="prediction", metricName="accuracy")#f1|weightedPrecision|weightedRecall
|accuracy
accuracy     = mc_evaluator.evaluate(predictions)
print("Accuracy:       " + str(accuracy))
```

```
Accuracy:       0.9574468085106383
```

In [26]:
```
# Binary Evaluator to evaluate our model
bi_evaluator = BinaryClassificationEvaluator(labelCol=col_target, metricName=
'areaUnderROC')  # areaUnderROC | areaUnderPR
areaunderroc = bi_evaluator.evaluate(predictions)
print("Area Under ROC: " + str(areaunderroc))
```

```
Area Under ROC: 0.71875
```

In [27]:
```
# Print True Positive vs. False Positives
predictions.groupBy('Species2','prediction').count().show()
```

```
+--------+----------+-----+
|Species2|prediction|count|
+--------+----------+-----+
|       0|       0.0|   15|
|       2|       2.0|   16|
|       1|       1.0|   14|
|       2|       0.0|    2|
+--------+----------+-----+
```

In [28]:
```
# Print Feature Importance
feature_importance_vars = sorted([(col_features[i],feature) for i,feature in e
numerate(model.stages[-1].featureImportances)], key=lambda x: x[1], reverse=Tr
ue)
print('Feature Importances (descending):')
for f in feature_importance_vars:
    print(f)
```

```
Feature Importances (descending):
('PetalLengthCm', 0.4724772637994708)
('PetalWidthCm', 0.4117938357353364)
('SepalLengthCm', 0.1048091068739575)
('SepalWidthCm', 0.010919793591235223)
```