Final Case Study

Objectives for Final Case Study on Applying Machine Learning:

1. **Problem Statement**
   Clearly define the problem statement that the case study aims to address using machine learning techniques.
2. **Data Collection**:
   Identify and gather relevant data sources required for the case study, ensuring data quality and integrity.
3. **Data Preprocessing**
   Perform necessary data preprocessing steps such as data cleaning, feature selection, and handling missing values to prepare the data for analysis.
4. **Exploratory Data Analysis**
   Conduct exploratory data analysis to gain insights into the dataset, identify patterns, and understand relationships between variables.
5. **Model Selection**
   Evaluate different machine learning algorithms suitable for the problem at hand and select the most appropriate model(s) based on performance metrics and interpretability.
6. **Model Training**
   Train the selected machine learning model(s) using the prepared dataset, optimizing hyperparameters to achieve the best possible performance.
7. **Model Evaluation**
   Assess the performance of the trained model(s) using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
8. **Model Interpretation**
   Interpret the trained model(s) to gain insights into the underlying patterns and factors driving the predictions, providing explanations for the model's decision-making process.
9. **Model Deployment**
   Develop a mechanism to deploy the trained model(s) in a real-world setting, ensuring scalability, efficiency, and reliability.
10. **Results Analysis**
    Analyze and interpret the results obtained from deploying the model(s), assessing their impact on solving the initial problem statement.
11. **Recommendations**
    Provide actionable recommendations based on the insights gained from the case study, suggesting potential improvements or future directions for further exploration.
12. **Documentation and Presentation**
    Document the entire case study process, including methodologies, findings, and limitations, and present the results in a clear and concise manner.

Remember to adapt these objectives based on your specific case study requirements and domain knowledge. Good luck with your final case study on applying machine learning!

Documentation for Case Study

1. Statement of the Problem
   Define the Problem
2. Structure of the Data
   Prepare Data. Describe the structure of the data set used
3. Model Selection

Evaluate Algorithms

a. Explain the selected machine learning algorithm suitable for the problem at hand based on performance and interpretability
b. Explain training process of the selected machine learning model using the prepared data set including the percentage of data that will be used for training and validation.
c. Discuss the performance of the trained model(s) using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.
d. Interpret the trained model(s) to gain insights into the underlying patterns and factors driving the predictions, providing explanations for the model's decision-making process.
e. Provide an example of how to deploy the trained model(s) in a real-world setting, ensuring scalability, efficiency, and reliability. (Model Deployment in a system)

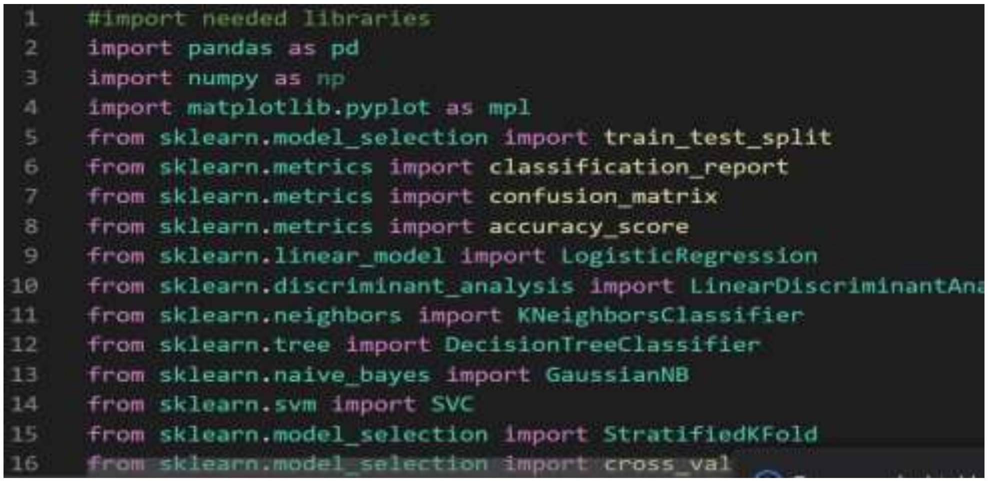
4. Result Analysis

   Improve the Results

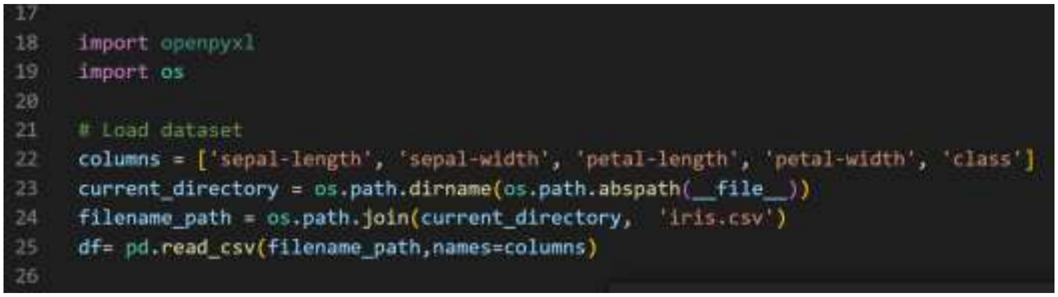
   a. Make prediction using the validation data set and evaluate the results
   b. Summarize the results obtained
   c. Analyze and interpret the results obtained from deploying the model(s), assessing their impact on solving the initial problem statement.

5. Presentation
   a. Present Results by discussing the program created in Python and its output
   b. Include the Python Program, and its output
6. Conclusion and Recommendation
   a. Discuss your conclusion
   b. List your Recommendation

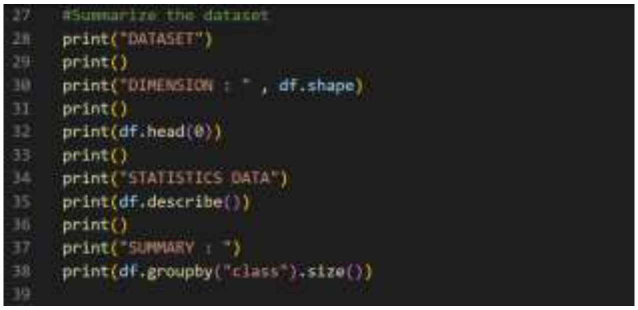
Sample Program:



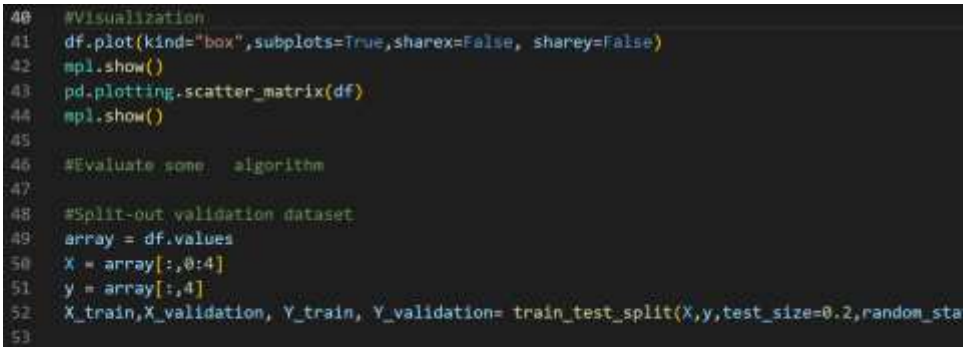
```
#import needed libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as mpl
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.discriminant_analysis import LinearDiscriminantAna
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val
```



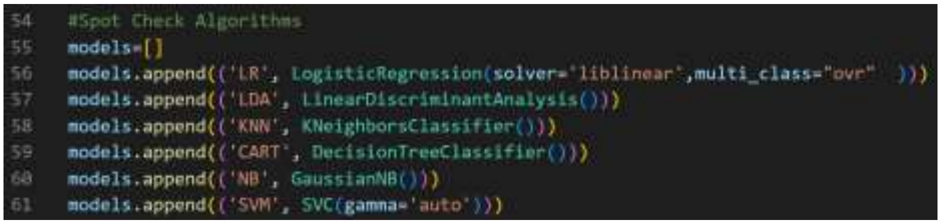
```

import openpyxl
import os

# Load dataset
columns = ['sepal-length', 'sepal-width', 'petal-length', 'petal-width', 'class']
current_directory = os.path.dirname(os.path.abspath(__file__))
filename_path = os.path.join(current_directory,  'iris.csv')
df= pd.read_csv(filename_path,names=columns)

```



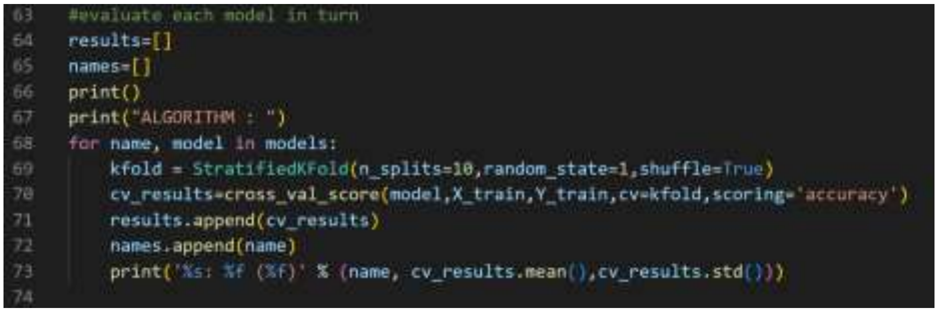
```python
#Summarize the dataset
print("DATASET")
print()
print("DIMENSION : " , df.shape)
print()
print(df.head(0))
print()
print("STATISTICS DATA")
print(df.describe())
print()
print("SUMMARY : ")
print(df.groupby("class").size())

```



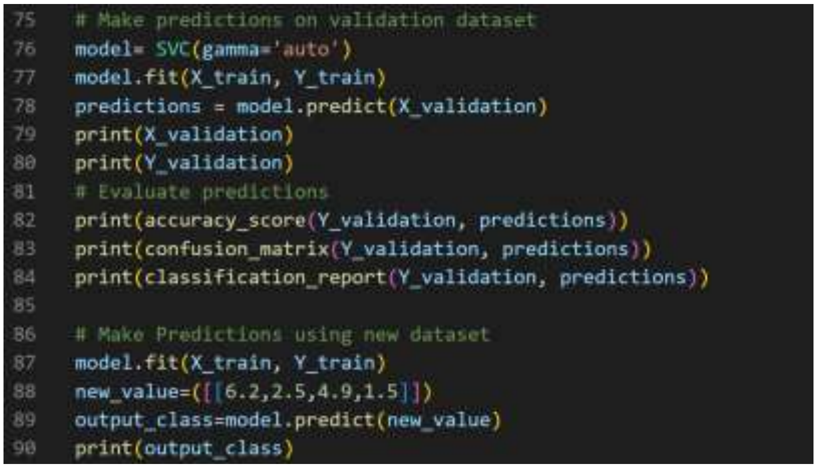
```python
#Visualization
df.plot(kind="box",subplots=True,sharex=False, sharey=False)
mpl.show()
pd.plotting.scatter_matrix(df)
mpl.show()

#Evaluate some   algorithm

#Split-out validation dataset
array = df.values
X = array[:,0:4]
y = array[:,4]
X_train,X_validation, Y_train, Y_validation= train_test_split(X,y,test_size=0.2,random_sta

```



```python
#Spot Check Algorithms
models=[]
models.append(('LR', LogisticRegression(solver='liblinear',multi_class="ovr"  )))
models.append(('LDA', LinearDiscriminantAnalysis()))
models.append(('KNN', KNeighborsClassifier()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC(gamma='auto')))
```



```python
#evaluate each model in turn
results=[]
names=[]
print()
print("ALGORITHM : ")
for name, model in models:
    kfold = StratifiedKFold(n_splits=10,random_state=1,shuffle=True)
    cv_results=cross_val_score(model,X_train,Y_train,cv=kfold,scoring='accuracy')
    results.append(cv_results)
    names.append(name)
    print('%s: %f (%f)' % (name, cv_results.mean(),cv_results.std()))

```



```python
# Make predictions on validation dataset
model= SVC(gamma='auto')
model.fit(X_train, Y_train)
predictions = model.predict(X_validation)
print(X_validation)
print(Y_validation)
# Evaluate predictions
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))

# Make Predictions using new dataset
model.fit(X_train, Y_train)
new_value=([[6.2,2.5,4.9,1.5]])
output_class=model.predict(new_value)
print(output_class)
```

Rubric for Assessment

Criteria	20 pts	15pts	10 pts	5 pts
Output Correctness	The output meets all the requirements specified in the project specification	The output meets at least 75% of the requirements specified in the project specification	The output meets at least 50% of the requirements specified in the project specification	The output meets at least 25% of the requirements specified in the project specification
Principle and Techniques	Included mention of at least 2 principles / techniques in addition to 2 technique / area of study we did NOT cover in class	Included mention of at least 2 principles / techniques in addition to 1 technique / area of study we did NOT cover in class	Included mention of at least 1 principles / techniques in addition to 1 technique / area of study we did NOT cover in class	1 principle / techniques mentioned and no additional non-class technique / are of study included
Analysis	Clear and scientifically accurate description are given for the analysis of every piece of evidence collected.	The description of evidence analysis for one piece of evidence is not detailed	The description of evidence analysis for two-or-more pieces of evidence is not detailed or includes scientifically inaccurate information	No evidence of analysis has been described
Concept Understanding	Answers the questions correct, reasonable and reflective of the code. The justification provided are sound	Answers the questions are correct but some justifications provided are weak	Answers the questions are correct but cannot justify a solution	Correct understanding of the problem, but was unable to explain workings of code provided
Readability	The program conforms to a coding standard that promotes readability. Internal documentation is comprehensive	Minor code formatting does not exhibit consistency in coding standards	Minor code formatting does not exhibit consistency in coding standards	Minimal internal documentation and code readability