(AIDI 2004) AI In Enterprise Systems

Wheat Seeds Prediction

Member Name	ID
Rushikumar Patel	100852498
Jaivardhan Singh	100849063
Alvin Henry	100709877

AUGUST 10th, 2022

DURHAM COLLEGE

EXECUTIVE SUMMARY

Our project is designed to implement, and long-form work or concept in the area of artificial intelligence. Our team has taken the dataset from Kaggle – Link to Dataset. The dataset consists of seven geometric parameters of wheat kernels and their associated varieties. The objective of our group is applying the learning algorithms that was introduced in the course to be implemented on a far more practical level dataset, and to determine the results that has obtained from different visualization that has been made because of those algorithms to clearly identify our learnings throughout the course. On a dataset of wheat seeds with seven attributes for three different kinds of wheat kernels, we will apply EDA: (Kama, Rosa, Canadian). Because it enables us to more clearly see the fundamental patterns, structures, and relationships in our data set.

After the implementation of algorithm in the dataset we should be able to include all the determining factors as the datasets have total 7 variables to measure and will be able to identify and differentiate though them.

INTRODUCTION

Three different wheat types, Kama, Rosa, and Canadian, each with 70 elements, were randomly chosen for the experiment, and made up the studied group. Using a soft X-ray approach, high quality visualization of the interior kernel structure was found. Compared to other, more advanced imaging techniques like scanning microscopy or laser technology, it is non-destructive and far less expensive. The photos were captured on X-ray KODAK plates measuring 13x18 cm. Wheat grain from experimental fields studied at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin was used in the studies.

The data collection is suitable for classification and cluster analysis activities. Through the inception of our course, we have been introduced with several artificial intelligence and machine learning algorithms and we have gathered the skills to apply them tothe real-life problems, and to test our knowledge we have come across this vast dataset that hasthe real-life key parameters.

The goal was to approach the dataset, perform an exploratory data analysis and to train and test the data and apply the clustering and classification for the best combination of hyperparameters.

OBJECTIVES

- Make an Exploratory Analysis of each variable showing the prevalence of wheat seeds for each category.
- Apply Clustering of hyperparameters for SVM classification, that maximizes the recall avoiding compromising othermetrics results as accuracy and specificity.

METHODOLOGY - AGILE

Agile project management is an iterative method of managing projects that emphasises the division of large projects into smaller, more manageable tasks that are finished in rapid iterations over the course of the project life cycle. Teams that use the Agile technique may work more quickly, adjust to shifting project needs, and streamline their workflow. Agile enables teams to be better prepared to quickly alter focus and direction, as the term implies.

Agile team roles: -

Scrum Master. The Scrum Master ensures that each sprint stays on track and helps to remove or resolve any issues or challenges that may come up. This role was set to Jaivardhan Singh

Team members. The people on this team are the ones who execute the work in each sprint. These teams, usually of three to seven people, can be composed of different specialties and strengths, or they can be teams of people with the same job roles. In our case, the team members were Rushi Patel and Alvin.

GANTT PLAN

S.No.	Sprint	Projected Plan	Verification
1	9th July – 18th July	Exploratory Data Analysis	Done
2	18th July – 24th July	K-Means Clustering Analysis	Done
3	24th July – 28th July	Model Development for SVM	Done
4	28th July – 31st July	Deployment and Testing	Done
5	31st July- 7th August	Implementing and deploying Flask Application	Done
6	7th August – 10th August	Final project validation and documentation	Done

Sprint meetings were held often to communicate the outstanding tasks, and to report the amount of work that has been completed, and finally to debate the scope of improvement. Our procedure was based on the SCRUM Methodology which consists of:

- Ensuring work quality
- Verifying whether the deployment is accurate and appropriate
- Risk Assessment

DATA

Description of data: This data was acquired from the 'UCI Center for Machine Learning' repository. It contains seven variables for three distinct types of wheat kernels: (Kama, Rosa, Canadian) designated as numerical variables 1, 2 & 3 respectively. The seven seed variables are:

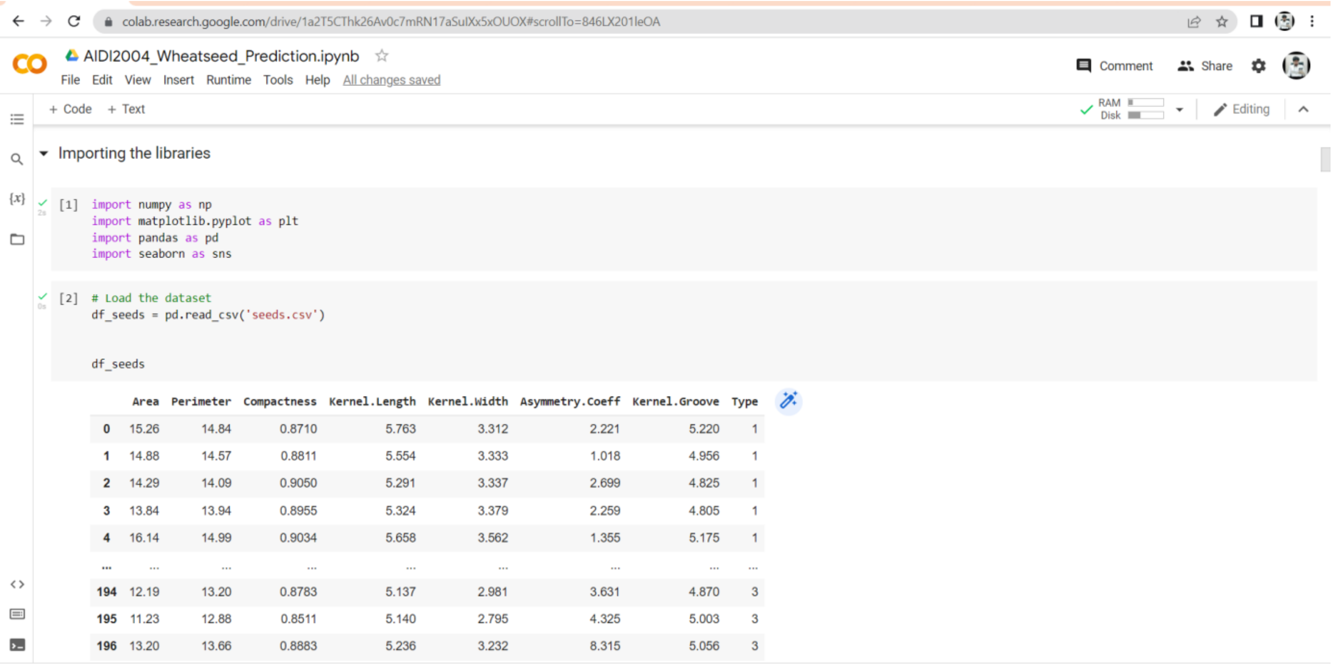
- Area
- Perimeter
- Compactness
- Kernel Length
- Kernel Width
- Asymmetry Coefficient
- Kernel Groove Length

The last column is reserved for the Kernel type. This dataset has 199 entries. Some of these variables are explicitly dependent. For example, compactness: C = 4piArea/(Perimeter)^2 has a linear proportional relationship with area, and also a square proportionality with kernel width.

Let's look at a summary of the dataset:

EXPLORATORY DATA ANALYSIS: -

Imports


colab.research.google.com/drive/1a2T5CThk26Av0c7mRN17aSuIXx5xOUOX#scrollTo=846LX201leOA
CO
AIDI2004_Wheatseed_Prediction.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Importing the libraries

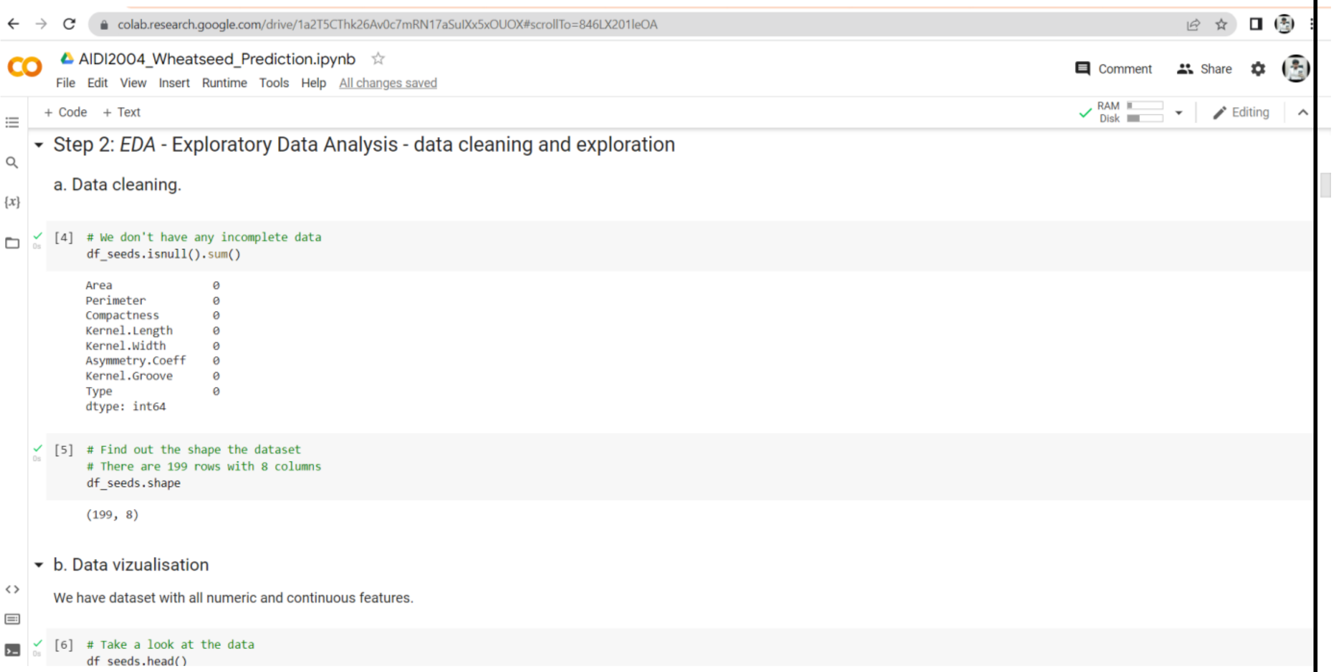
```
[1] import numpy as np
    import matplotlib.pyplot as plt
    import pandas as pd
    import seaborn as sns
```

```
[2] # Load the dataset
    df_seeds = pd.read_csv('seeds.csv')


    df_seeds
```

	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff	Kernel.Groove	Type
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	1
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	1
...	...	...	...	...	...	...	...	...
194	12.19	13.20	0.8783	5.137	2.981	3.631	4.870	3
195	11.23	12.88	0.8511	5.140	2.795	4.325	5.003	3
196	13.20	13.66	0.8883	5.236	3.232	8.315	5.056	3

Dataset Information

Functions



colab.research.google.com/drive/1a2T5CThk26Av0c7mRN17aSulXx5xOUOX#scrollTo=846LX201leOA
AIDI2004_Wheatseed_Prediction.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing

Step 2: *EDA* - Exploratory Data Analysis - data cleaning and exploration

a. Data cleaning.

```
[4] # We don't have any incomplete data
    df_seeds.isnull().sum()

    Area              0
    Perimeter         0
    Compactness       0
    Kernel.Length     0
    Kernel.Width      0
    Asymmetry.Coeff   0
    Kernel.Groove     0
    Type              0
    dtype: int64
```

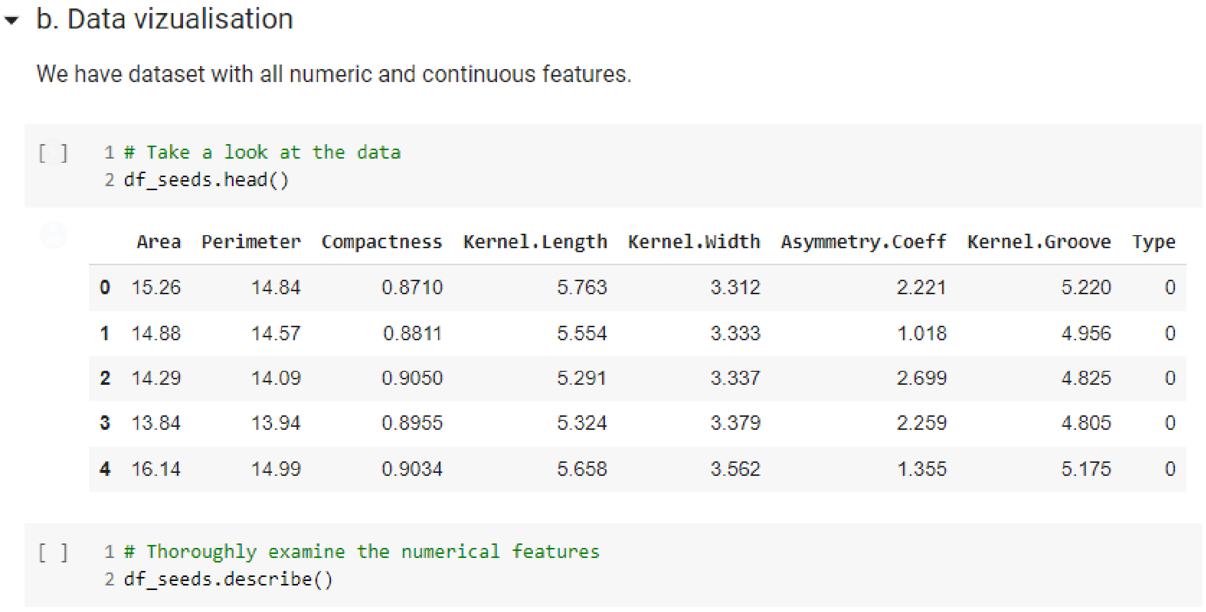
```
[5] # Find out the shape the dataset
    # There are 199 rows with 8 columns
    df_seeds.shape

    (199, 8)
```

b. Data vizualisation

We have dataset with all numeric and continuous features.

```
[6] # Take a look at the data
    df_seeds.head()
```

EDA: -



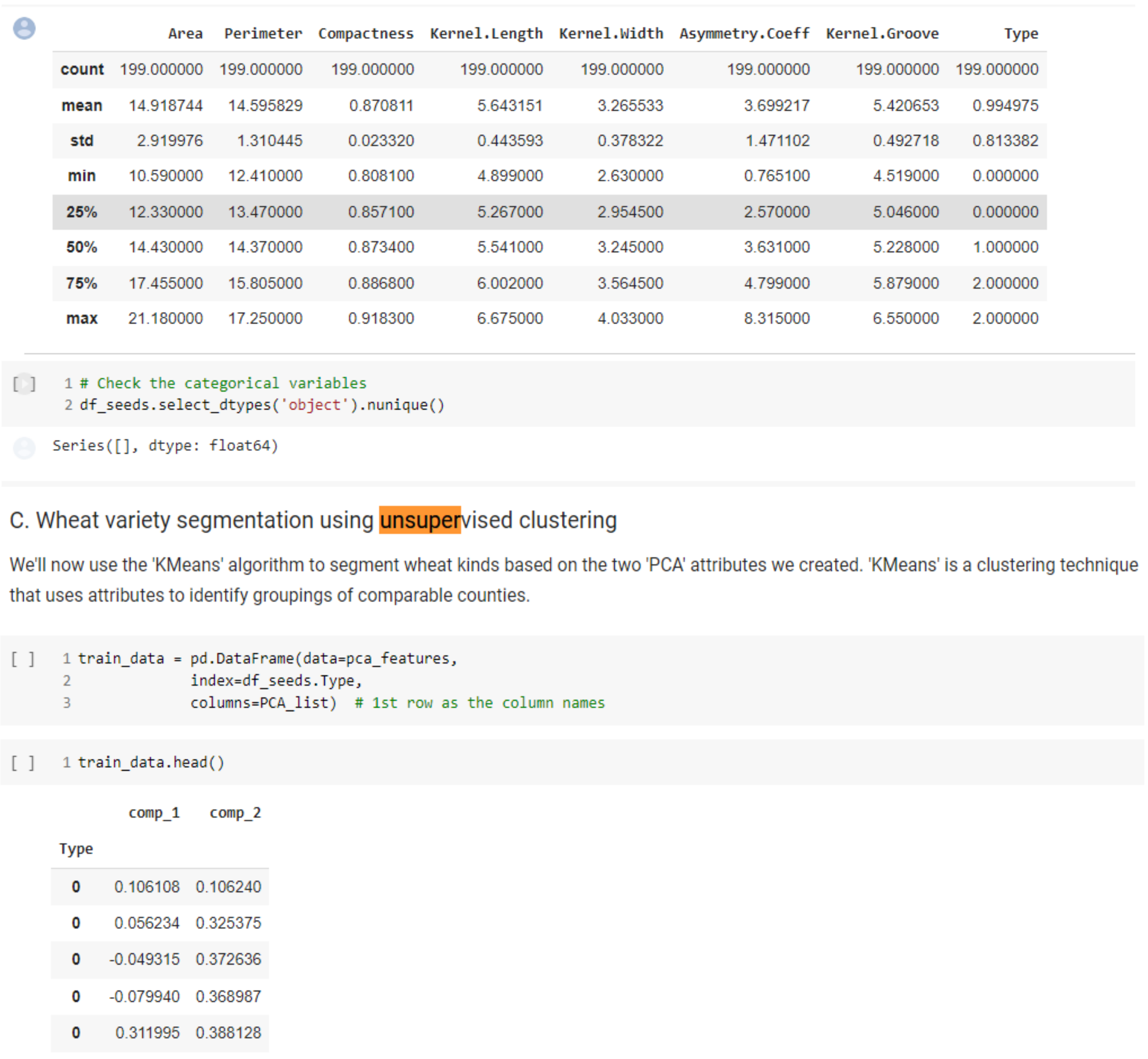
b. Data vizualisation

We have dataset with all numeric and continuous features.

```
[ ] # Take a look at the data
    df_seeds.head()
```

	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff	Kernel.Groove	Type
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	0
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	0
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	0
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	0
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	0

```
[ ] # Thoroughly examine the numerical features
    df_seeds.describe()
```



	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff	Kernel.Groove	Type
count	199.000000	199.000000	199.000000	199.000000	199.000000	199.000000	199.000000	199.000000
mean	14.918744	14.595829	0.870811	5.643151	3.265533	3.699217	5.420653	0.994975
std	2.919976	1.310445	0.023320	0.443593	0.378322	1.471102	0.492718	0.813382
min	10.590000	12.410000	0.808100	4.899000	2.630000	0.765100	4.519000	0.000000
25%	12.330000	13.470000	0.857100	5.267000	2.954500	2.570000	5.046000	0.000000
50%	14.430000	14.370000	0.873400	5.541000	3.245000	3.631000	5.228000	1.000000
75%	17.455000	15.805000	0.886800	6.002000	3.564500	4.799000	5.879000	2.000000
max	21.180000	17.250000	0.918300	6.675000	4.033000	8.315000	6.550000	2.000000

```
# Check the categorical variables
df_seeds.select_dtypes('object').nunique()
```

```
Series([], dtype: float64)
```

C. Wheat variety segmentation using unsupervised clustering

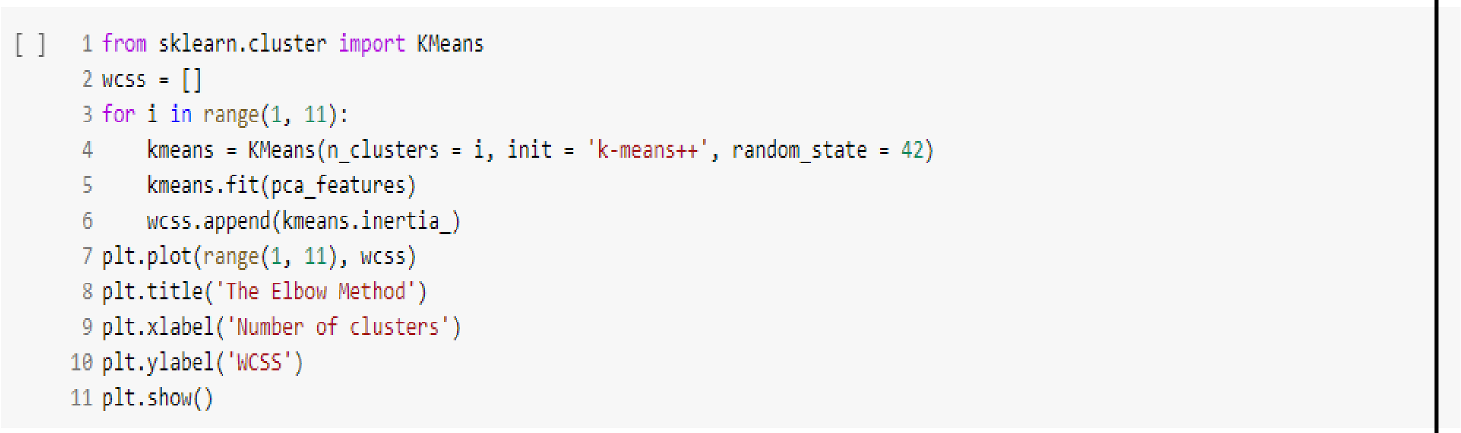
We'll now use the 'KMeans' algorithm to segment wheat kinds based on the two 'PCA' attributes we created. 'KMeans' is a clustering technique that uses attributes to identify groupings of comparable counties.

```
train_data = pd.DataFrame(data=pca_features,
             index=df_seeds.Type,
             columns=PCA_list)  # 1st row as the column names
```

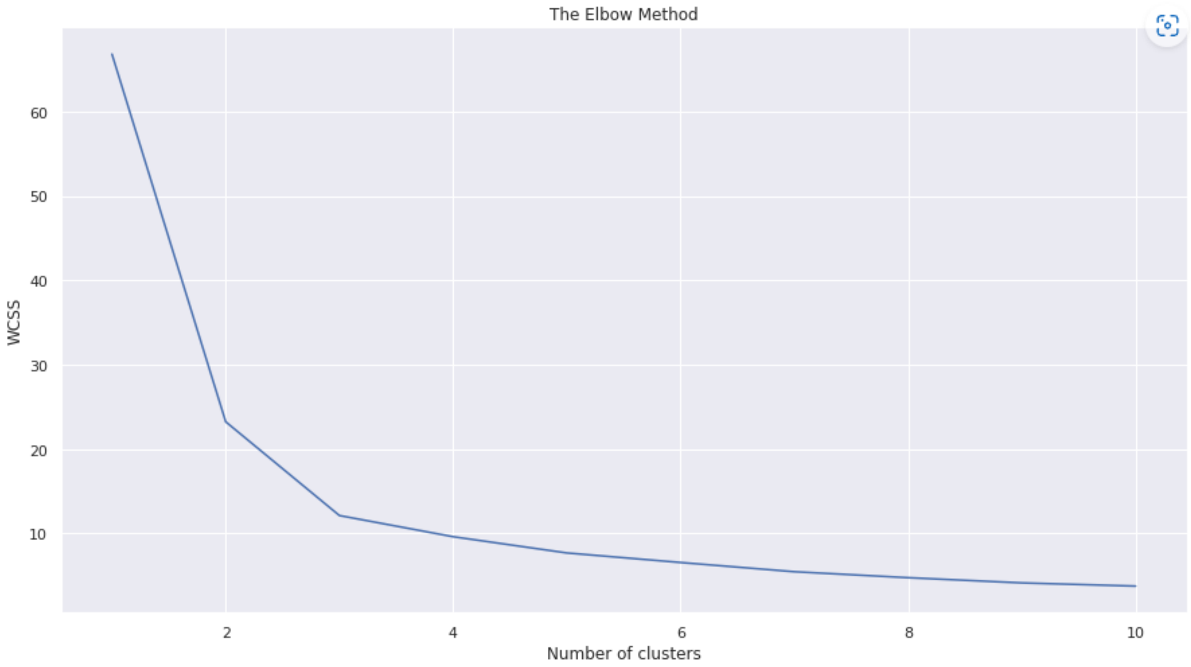
```
train_data.head()
```

	comp_1	comp_2
Type		
0	0.106108	0.106240
0	0.056234	0.325375
0	-0.049315	0.372636
0	-0.079940	0.368987
0	0.311995	0.388128

As with our PCA model, we begin by naming and defining the hyperparameters of our KMeans model. The KMeans technique lets the user define the number of clusters to identify. Using the elbow-inertia approach, we will determine the optimal number of clusters.



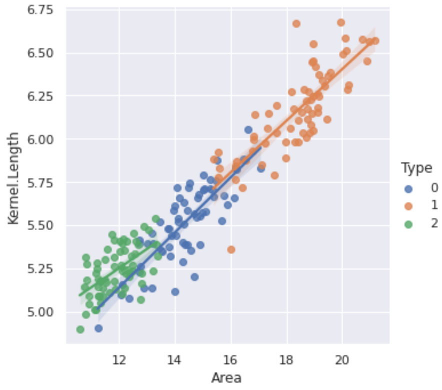
```
from sklearn.cluster import KMeans
wcss = []
for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++', random_state = 42)
    kmeans.fit(pca_features)
    wcss.append(kmeans.inertia_)
plt.plot(range(1, 11), wcss)
plt.title('The Elbow Method')
plt.xlabel('Number of clusters')
plt.ylabel('WCSS')
plt.show()
```



We can clearly notice that ideal number of clusters is 3.

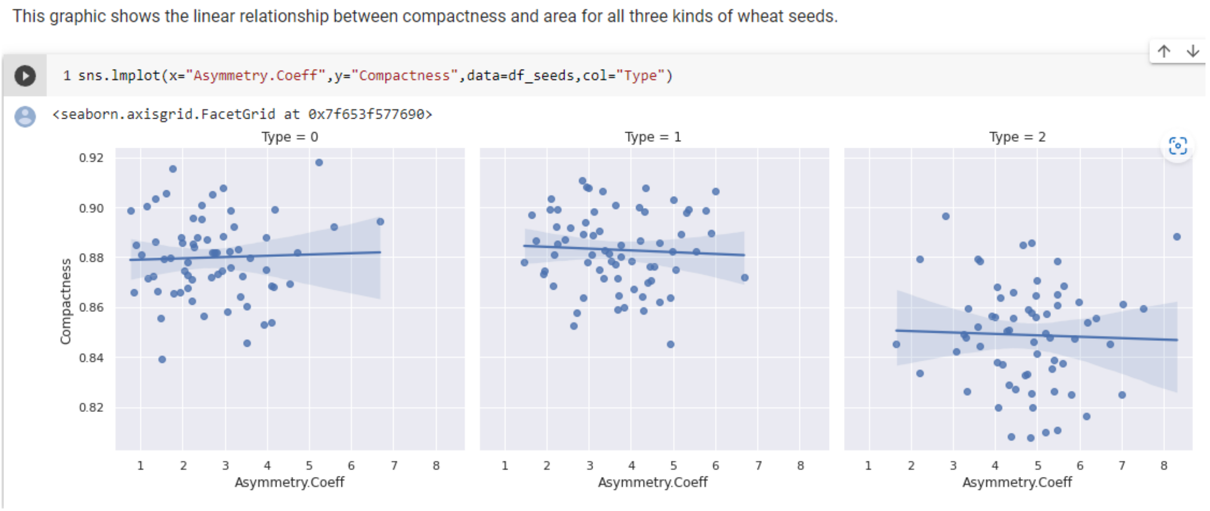
```
sns.lmplot(x="Area",y="Kernel.Length",data=df_seeds,hue="Type")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f6541534910>
```



The accompanying figure clearly shows the separable boundaries between kernel variants depending on area and kernel length. The Canadian

This graphic shows the linear relationship between compactness and area for all three kinds of wheat seeds.

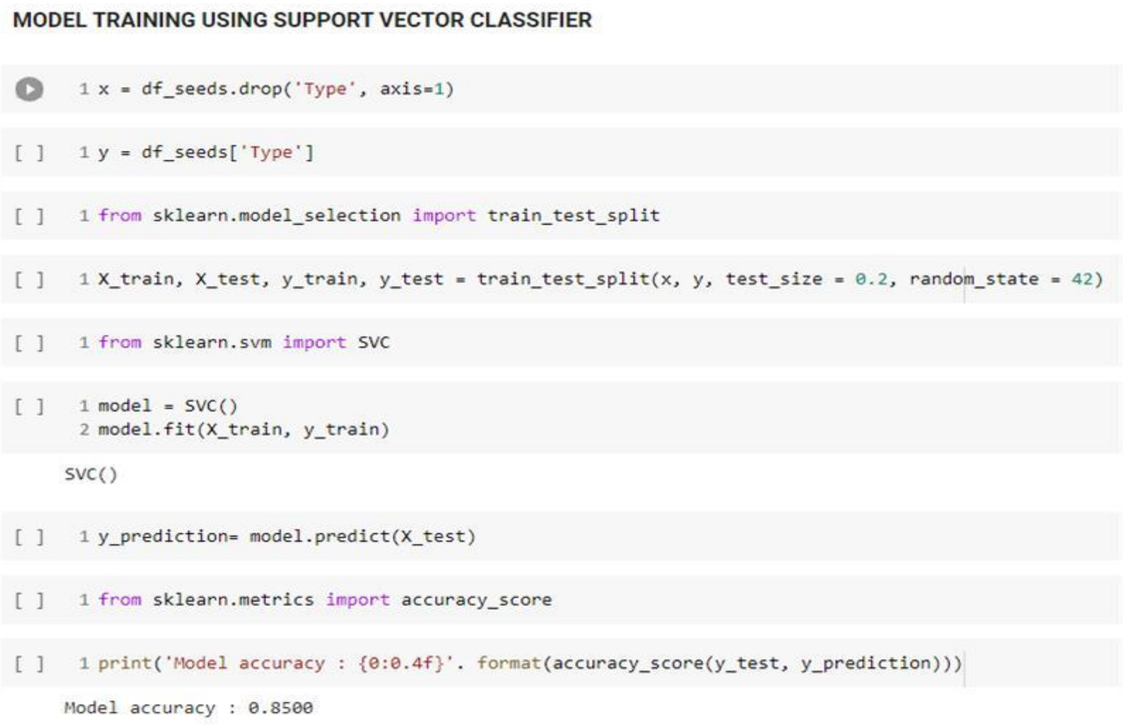


This graphic shows the linear relationship between compactness and area for all three kinds of wheat seeds.

```
sns.lmplot(x="Asymmetry.Coeff",y="Compactness",data=df_seeds,col="Type")
```

```
<seaborn.axisgrid.FacetGrid at 0x7f653f577690>
```

Accuracy: -



MODEL TRAINING USING SUPPORT VECTOR CLASSIFIER

```
x = df_seeds.drop('Type', axis=1)
```

```
y = df_seeds['Type']
```

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 42)
```

```
from sklearn.svm import SVC
```

```
model = SVC()
model.fit(X_train, y_train)
```

```
SVC()
```

```
y_prediction= model.predict(X_test)
```

```
from sklearn.metrics import accuracy_score
```

```
print('Model accuracy : {0:0.4f}'. format(accuracy_score(y_test, y_prediction)))
```

```
Model accuracy : 0.8500
```

Algorithm used: -

- SVC - Support vector machines (SVMs) are a class of supervised learning methods for classification, regression, and detection of outlier.

Some benefits of support vector machines:

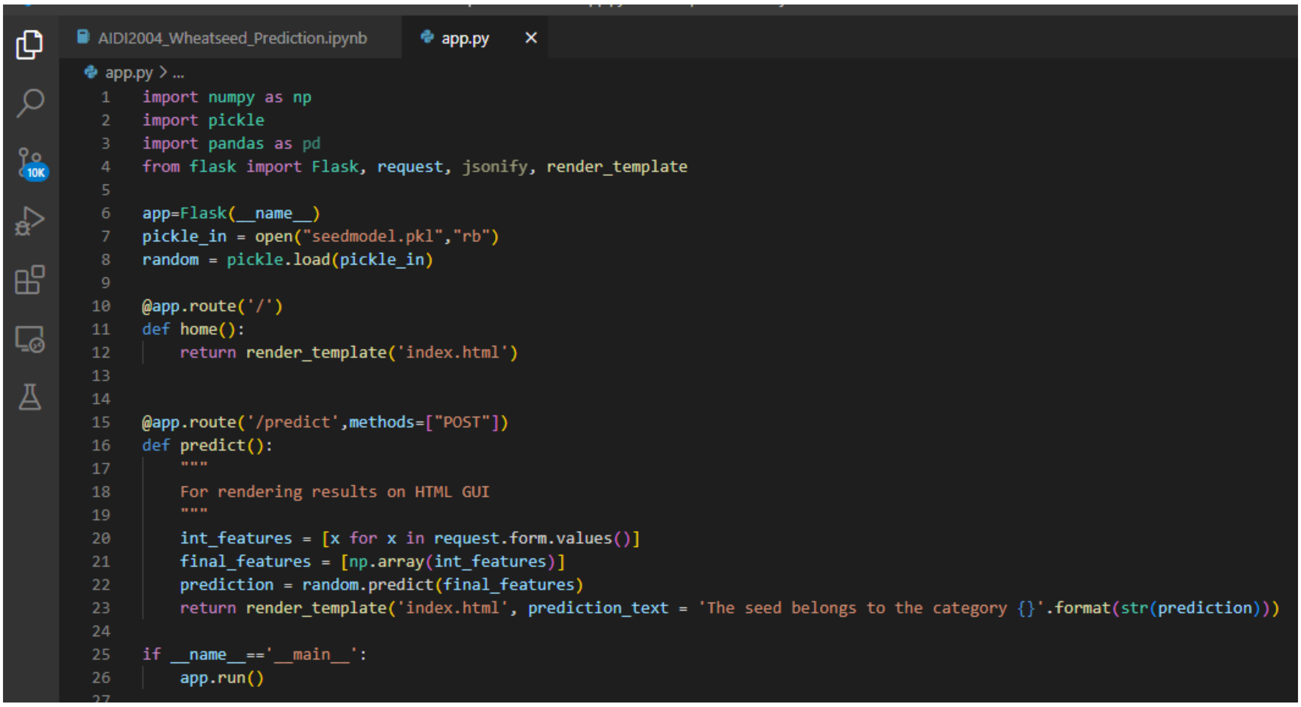
- It works well in three-dimensional spaces.
- When the number of dimensions exceeds the number of samples, it is still effective.
- It is memory efficient since it uses a subset of training points (called support vectors) in the decision function.
- Versatile: For the decision function, several Kernel functions can be provided. Common kernels are included, however custom kernels can also be specified.

Deployment of model: -

1) Flask: -

- One aspect of the data is the building/training of a model using different methods on a sizable dataset. The second step in implementing machine learning in the real world is employing these models in the various applications.

- We must deploy it over the internet so that users outside of our organization can use it in order to utilize it to forecast the new data. This article will discuss how we used Flask to train a machine learning model and build a web application using it.



```python
import numpy as np
import pickle
import pandas as pd
from flask import Flask, request, jsonify, render_template

app=Flask(__name__)
pickle_in = open("seedmodel.pkl","rb")
random = pickle.load(pickle_in)

@app.route('/')
def home():
    return render_template('index.html')


@app.route('/predict',methods=["POST"])
def predict():
    """
    For rendering results on HTML GUI
    """
    int_features = [x for x in request.form.values()]
    final_features = [np.array(int_features)]
    prediction = random.predict(final_features)
    return render_template('index.html', prediction_text = 'The seed belongs to the category {}'.format(str(prediction)))

if __name__=='__main__':
    app.run()
```

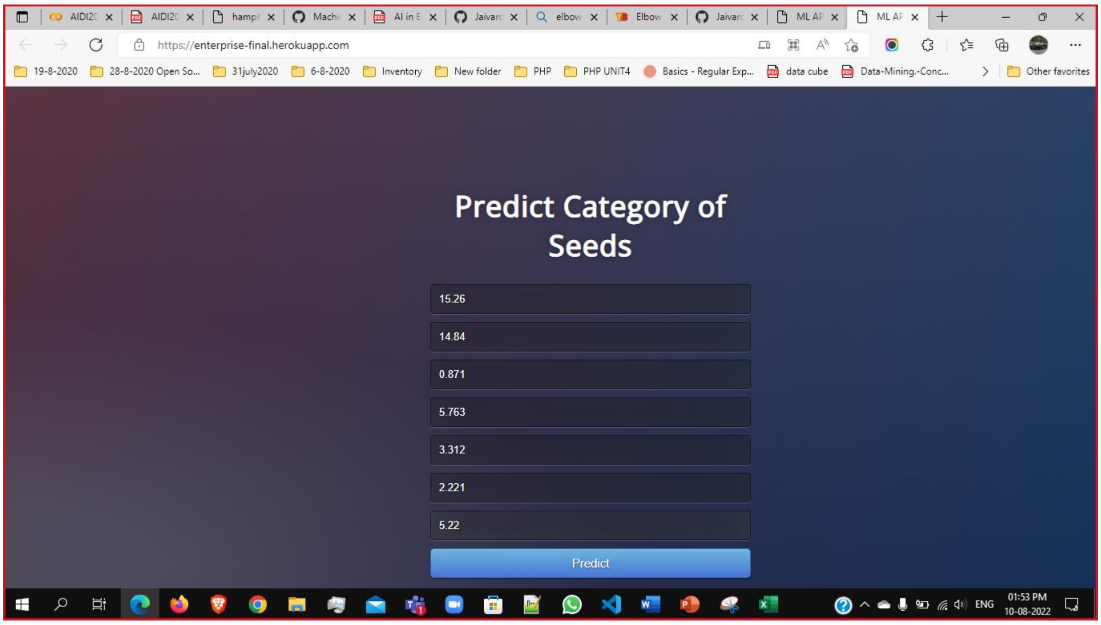
2) GitHub: -

GitHub is an online software development platform used for storing, tracking, and collaborating on software projects. It enables developers to upload their own code files and to collaborate with fellow developers on open-source projects. GitHub also serves as a social networking site in which developers can openly network, collaborate, and pitch their work.

https://github.com/Jaivardhan3/AI-Enterprise-Final

3) Heroku: -

- Heroku is well recognized for running applications in dynos, which are essentially just virtual machines that can be powered on or off depending on how big your application is. Dynos can be used as movable building blocks to run your project.
- Link of Heroku: - ML API (enterprise-final.herokuapp.com)



https://enterprise-final.herokuapp.com

Predict Category of Seeds

15.26

14.84

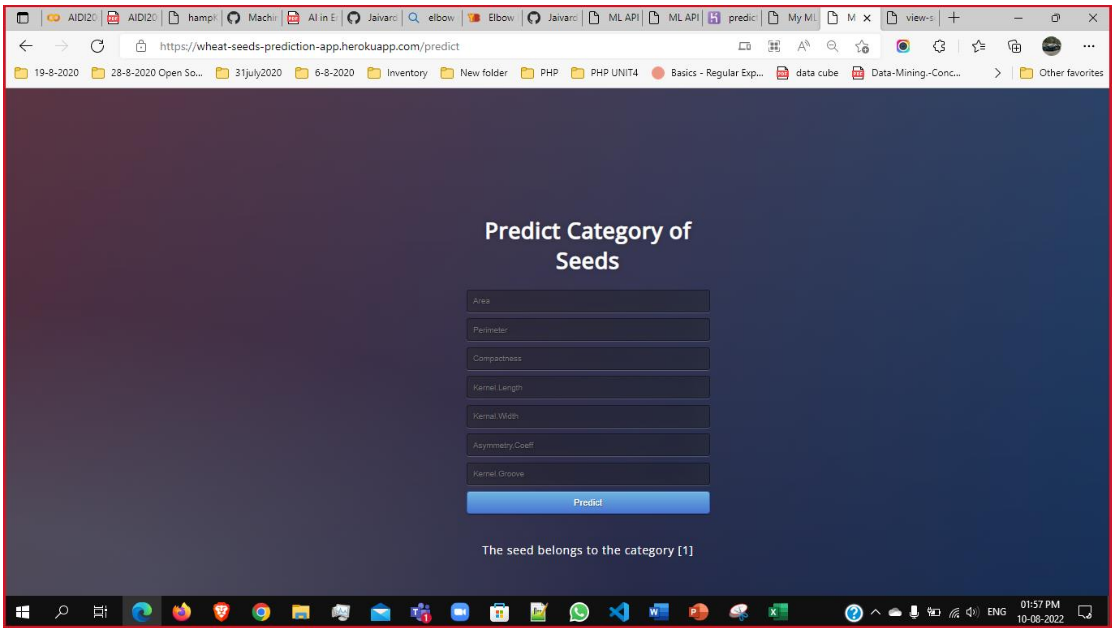
0.871

5.763

3.312

2.221

5.22



Predict Category of Seeds

Area

Perimeter

Compactness

Kernel.Length

Kernal.Width

Asymmetry.Coeff

Kernel.Groove

Predict

The seed belongs to the category [1]

CONCLUSION: -

We may make a heatmap depicting the locations of the centroids in the changed feature space. This provides the characteristics that distinguish each cluster. Unsupervised learning results are usually difficult to interpret. This is one approach of integrating the results of PCA with clustering techniques. Because we were able to analyze the makeup of each PCA component, we can understand what each centroid signifies in terms of the 'PCA' components that we interpreted earlier.

Explanation of modelling results is a critical step in putting our research to use. We may make precise inferences based on the data by integrating PCA and KMeans, as well as the information included in the model features.

References: -

- https://scikit-learn.org/stable/modules/svm.html
- https://www.kaggle.com/datasets/jmcaro/wheat-seedsuci
- https://www.workfront.com/project-management/methodologies/agile
- https://www.freecodecamp.org/news/svm-machine-learning-tutorial-what-is-the-support-vector-machine-algorithm-explained-with-code-examples/
- https://faun.pub/clustering-wheat-varieties-using-kmeans-8c8288e6b643
- https://blog.hubspot.com/website/what-is-github-used-for