



Model Development Phase Template

Date	21 June 2024	
Team ID	739783	
Project Title	Opticrop: Smart Agricultural Production Optimization Engine	
Maximum Marks	4 Marks	

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```
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=0)
print("The shape of x train",x_train.shape)
print("The shape of x test", x_test.shape)
print("The shape of y train",x_train.shape)
print("The shape of y test", x_test.shape)
```

```
plt.rcParams['figure.figsize']=(10,4)
WCSS=[]
for i in range(1,11):
    km=KMeans(n_clusters=i,init="k-means++", max_iter=300,n_init=10,random_state=0)
    km.fit(x)
    WCSS.append(km.inertia_)
plt.plot(range (1,11), WCSS)
plt.title("The Elbow method", fontsize=20)
plt.xlabel("No of clusters")
plt.ylabel("WCSS")
plt.show()
```





```
km=KMeans(n_clusters=4,init="k-means++", max_iter=300,n_init=10,random_state=0)
y_means=km.fit_predict(x)
a=df['label']
y_means=pd.DataFrame(y_means)
z=pd.concat([y_means,a],axis=1)
z=z.rename(columns={0:'cluster'})
print("lets check the results after applying the K-Means clustering analysis \n")
print("Crops in First cluster:", z[z['cluster']==0] ['label'].unique())
print("-----")
print("Crops in Second cluster:", z[z['cluster']==1]['label'].unique())
print("----")
print("Crops in Third cluster:", z[z['cluster']==2]['label'].unique())
print("----")
print("Crops in Fourth cluster:", z[z['cluster']==3]['label'].unique())
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model.fit(x_train,y_train)
y_pred=model.predict(x_test)
from sklearn.metrics import confusion_matrix
plt.rcParams["figure.figsize"]=(10,10)
cm=confusion_matrix(y_test,y_pred)
sns.heatmap(cm,annot=True,cmap='Wistia')
plt.title("Confusion matrix for logistic regression", fontsize=15)
plt.show()
from sklearn.metrics import classification report
cr=classification_report(y_test,y_pred)
print(cr)
```

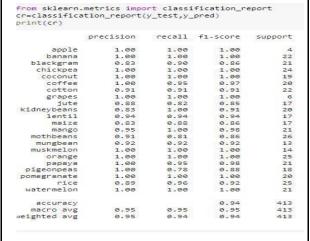
Model Validation and Evaluation Report:

Model	Classification Report		Confusion Matrix
		F1	
		Score	





Kmeans,
Linear
regression



94%

