Ahmedabad
University

Subject:- CSE523 Machine Learning

Weekly Report 6

Section-1

Submitted to faculty: Prof. Mehul Raval

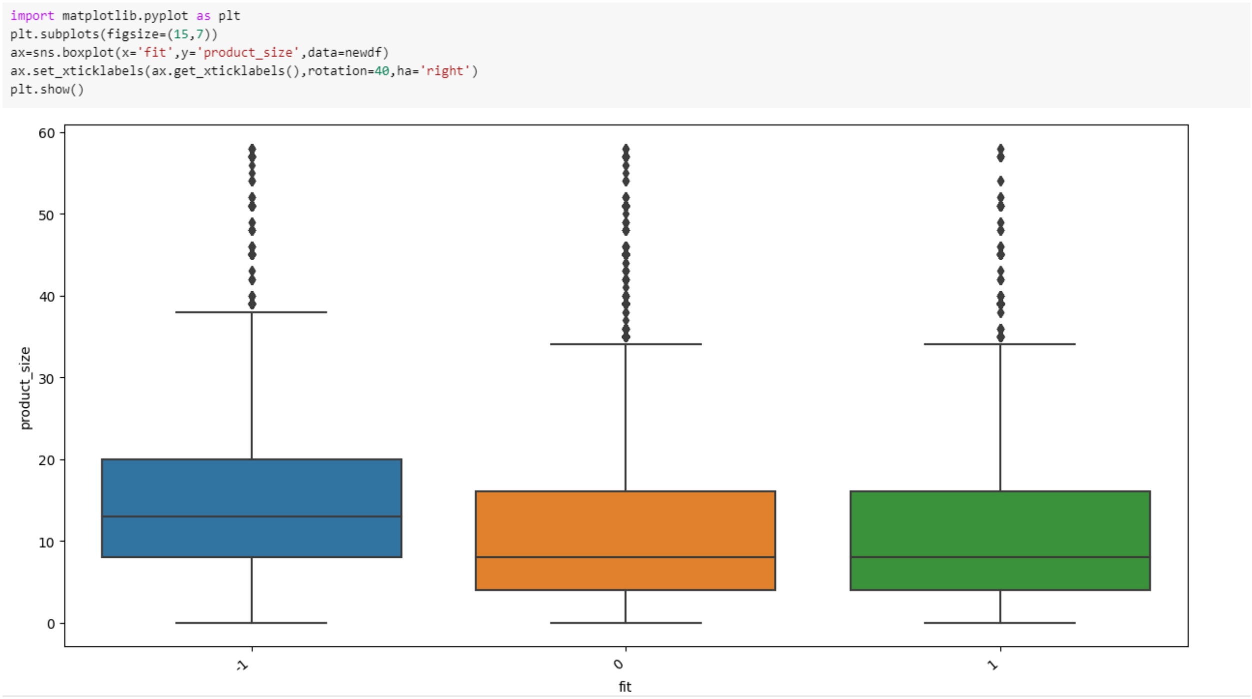
Date of Submission: 01-04-2023

Student Details

Roll No.	Name of the Student	Name of the Program
AU2040122	Aditi Vasa	Btech CSE
AU2040002	Shrey Somani	Btech CSE
AU2040196	Vandan Shah	Btech CSE
AU2040048	Ronit Shah	Btech CSE

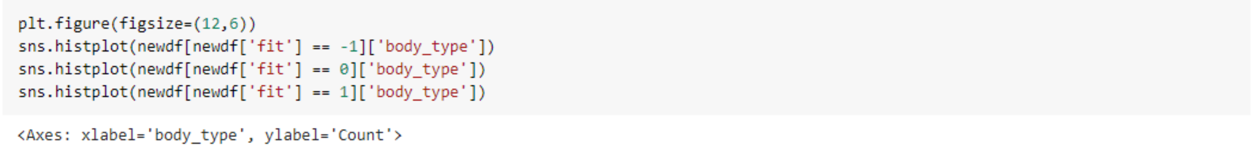
2020-2021 (Monsoon Semester)

For this week our goal was to try understand our data better by doing field specific EDA. The correlation matrix showed a better relation between the fit and the product size. Hence, we plotted the box and whisker's plot to understand this relationship better.



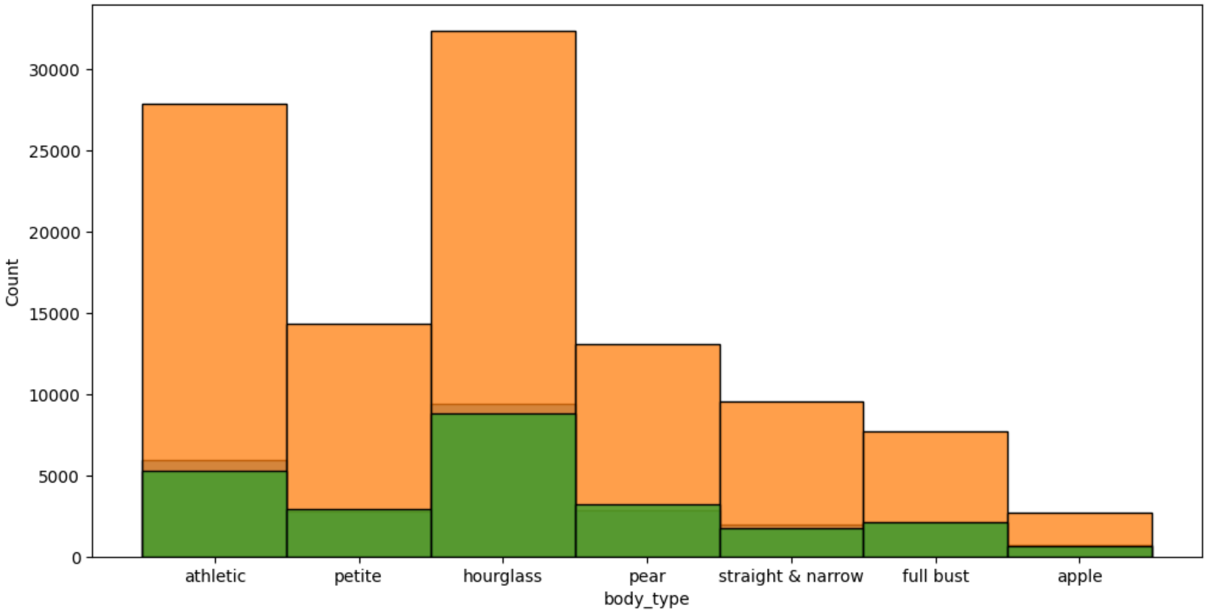
```
import matplotlib.pyplot as plt
plt.subplots(figsize=(15,7))
ax=sns.boxplot(x='fit',y='product_size',data=newdf)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```

We also found that body type plays a crucial role in deciding the fit of the product. Thus, we plotted histogram to understand how each body type corresponds to the fit of the product.

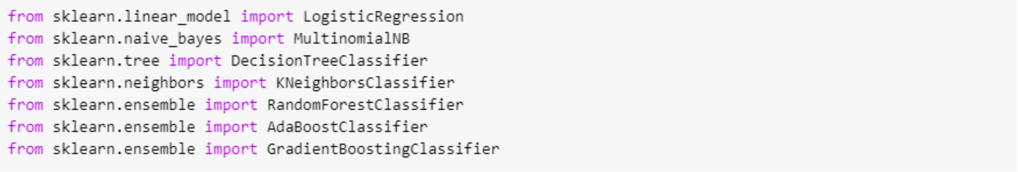


```
plt.figure(figsize=(12,6))
sns.histplot(newdf[newdf['fit'] == -1]['body_type'])
sns.histplot(newdf[newdf['fit'] == 0]['body_type'])
sns.histplot(newdf[newdf['fit'] == 1]['body_type'])
```

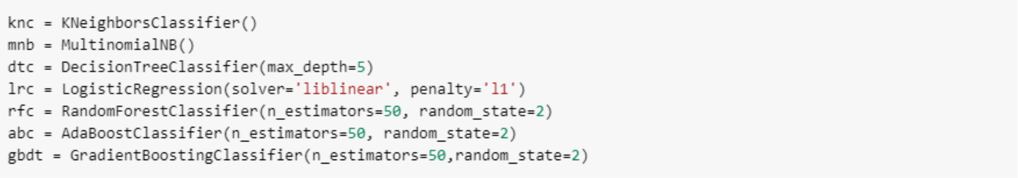
```
<Axes: xlabel='body_type', ylabel='Count'>
```



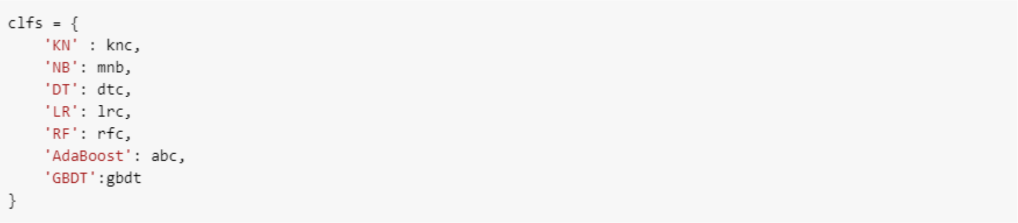
Next our goal was to check the accuracy of our model by applying various machine learning algorithms which we did. We used KNeighbors classifier, Multinomial Naive Baiyes' approach, Decision Tree, Random Forest, Ada Boost classifier and Gradient Boosting Classifier. Following are the accuracy results for each algorithm.



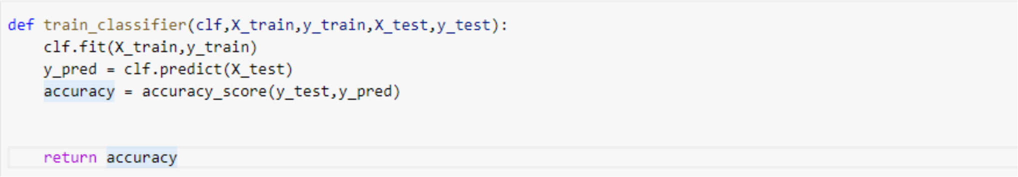
```
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import MultinomialNB
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
```



```
knc = KNeighborsClassifier()
mnb = MultinomialNB()
dtc = DecisionTreeClassifier(max_depth=5)
lrc = LogisticRegression(solver='liblinear', penalty='l1')
rfc = RandomForestClassifier(n_estimators=50, random_state=2)
abc = AdaBoostClassifier(n_estimators=50, random_state=2)
gbdt = GradientBoostingClassifier(n_estimators=50,random_state=2)
```

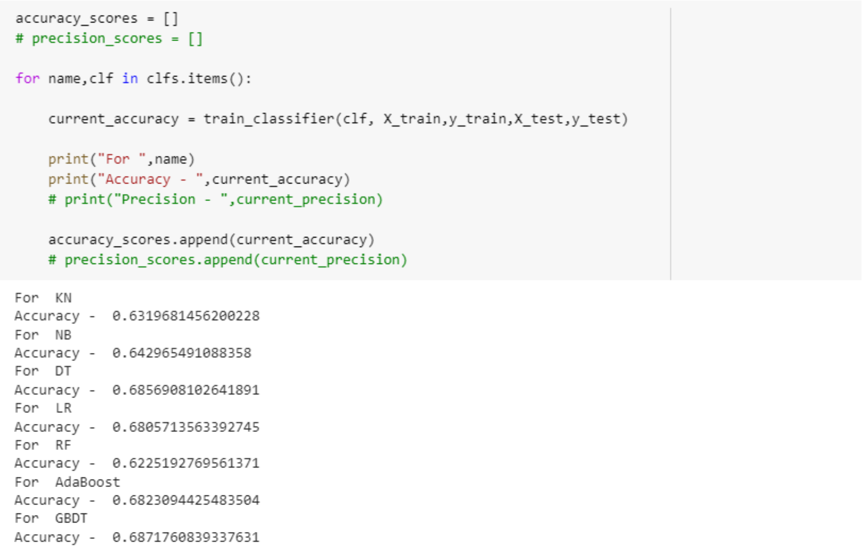


```
clfs = {
    'KN' : knc,
    'NB': mnb,
    'DT': dtc,
    'LR': lrc,
    'RF': rfc,
    'AdaBoost': abc,
    'GBDT':gbdt
}
```



```
def train_classifier(clf,X_train,y_train,X_test,y_test):
    clf.fit(X_train,y_train)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)


    return accuracy
```



```
accuracy_scores = []
# precision_scores = []

for name,clf in clfs.items():

    current_accuracy = train_classifier(clf, X_train,y_train,X_test,y_test)

    print("For ",name)
    print("Accuracy - ",current_accuracy)
    # print("Precision - ",current_precision)

    accuracy_scores.append(current_accuracy)
    # precision_scores.append(current_precision)
```

```
For  KN
Accuracy -  0.6319681456200228
For  NB
Accuracy -  0.642965491088358
For  DT
Accuracy -  0.6856908102641891
For  LR
Accuracy -  0.6805713563392745
For  RF
Accuracy -  0.6225192769561371
For  AdaBoost
Accuracy -  0.6823094425483504
For  GBDT
Accuracy -  0.6871760839337631
```

Goals for next week:-

- Define quadratic decision boundaries.
- Understanding where we can improve our accuracy and find the optimal algorithm