Abalones

Francisco Aramburu

Abalones

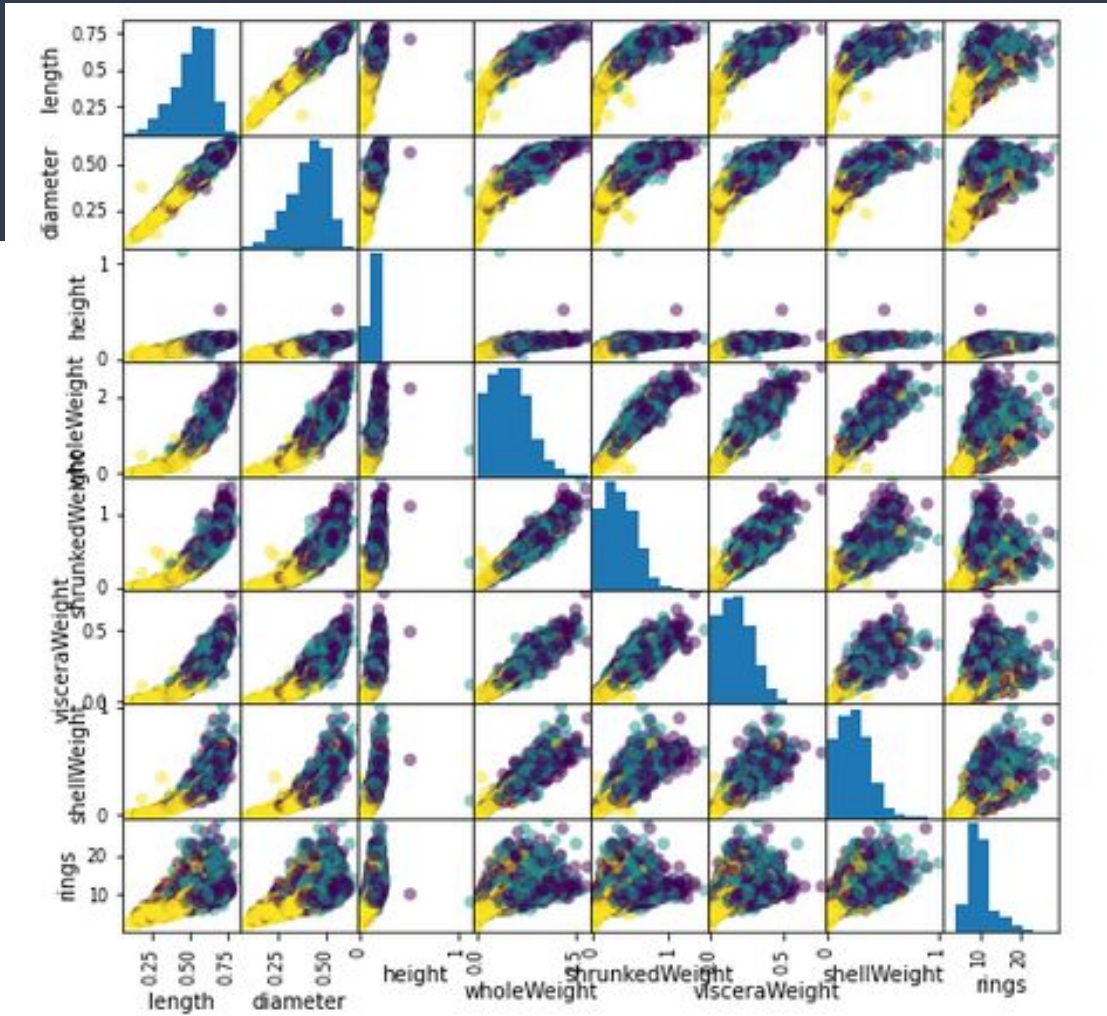
Abalones

Dataset

	sex	length	diameter	height	wholeWeight	shrunkedWeight	visceraWeight	shellWeight	rings
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

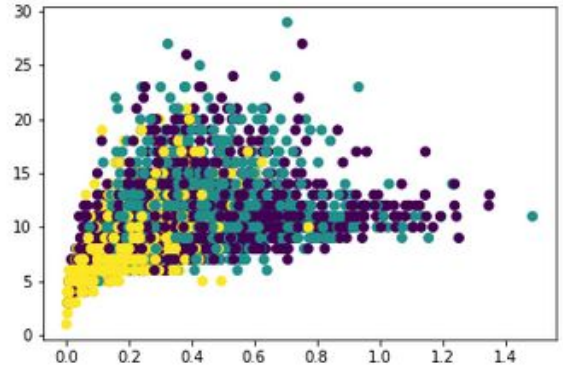
Stats

	sex	length	diameter	height	wholeWeight	shrunkedWeight	visceraWeight	shellWeight	rings
count	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000	4177.000000
mean	0.955470	0.523992	0.407881	0.139516	0.828742	0.359367	0.180594	0.238831	9.933684
std	0.827815	0.120093	0.099240	0.041827	0.490389	0.221963	0.109614	0.139203	3.224169
min	0.000000	0.075000	0.055000	0.000000	0.002000	0.001000	0.000500	0.001500	1.000000
25%	0.000000	0.450000	0.350000	0.115000	0.441500	0.186000	0.093500	0.130000	8.000000
50%	1.000000	0.545000	0.425000	0.140000	0.799500	0.336000	0.171000	0.234000	9.000000
75%	2.000000	0.615000	0.480000	0.165000	1.153000	0.502000	0.253000	0.329000	11.000000
max	2.000000	0.815000	0.650000	1.130000	2.825500	1.488000	0.760000	1.005000	29.000000


length
diameter
height
wholeWeight
shrunkedWeight
visceraWeight
shellWeight
rings

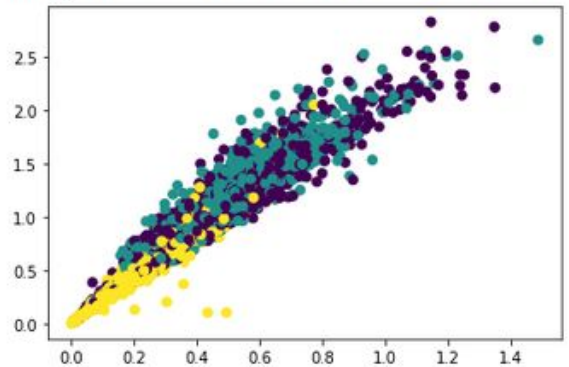
```
1 plt.scatter(db['shrunkedWeight'], db['rings'], c = db['sex'])
```

```
<matplotlib.collections.PathCollection at 0x7fb81d5dcc10>
```




```
[238] 1 plt.scatter(db['shrunkedWeight'], db['wholeWeight'], c = db['sex'])
```

```
<matplotlib.collections.PathCollection at 0x7fb81d57d9d0>
```



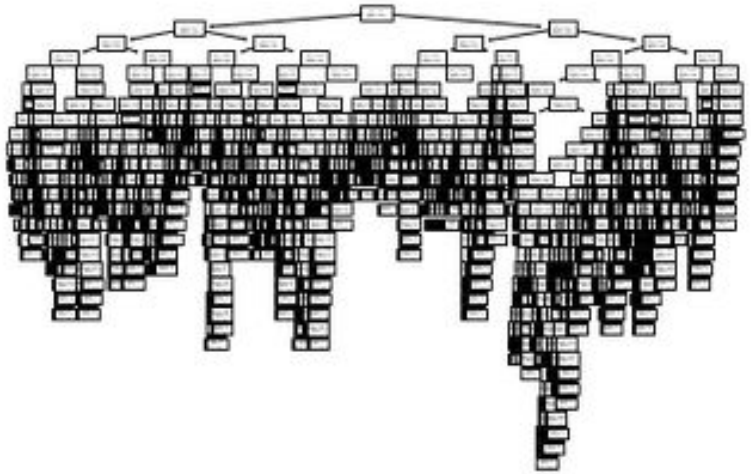
Intento 1

```
1 from sklearn.tree import DecisionTreeClassifier
2 from sklearn.model_selection import train_test_split
3 from sklearn.metrics import accuracy_score
4
5 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state=1)
6 dt = DecisionTreeClassifier(criterion='entropy', random_state=1)
7 dt.fit(X_train, y_train)
8 y_pred = dt.predict(X_test)
9 accuracy_score(y_test, y_pred)
```

```
0.4700956937799043
```

Árbol

```
1 tree.plot_tree(dt)
```



Voting Classifier

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, random_state=SEED)

lr = LogisticRegression(random_state=SEED)
knn = KNN(n_neighbors=27)
dt = DecisionTreeClassifier(min_samples_leaf=0.13, random_state=SEED)

classifiers = [('Logistic Regression', lr),
               ('K Nearest Neighbours', knn),
               ('Classification Tree', dt)]
```

```
lbfgs failed to converge
```

```
0.557
0.532
0.533
Voting Classifier: 0.547
```

Bagging

```
from sklearn.ensemble import BaggingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

SEED = 1
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3, stratify=y, random_state=SEED)

dt = DecisionTreeClassifier(max_depth =4, min_samples_leaf=0.16, random_state=SEED)
bc = BaggingClassifier(base_estimator=dt, n_estimators=300, n_jobs=-1, oob_score=True)
bc.fit(X_train, y_train)
y_pred = bc.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
print('Accuracy of Bagging Classifier: {:.3f}'.format(accuracy))
```

```
Accuracy of Bagging Classifier: 0.522
```

Random Forest

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error as MSE

SEED = 1

#400 Regression trees and each leaf has at least 12% of data
rf = RandomForestClassifier(n_estimators = 200,
                            max_features = 'sqrt',
                           random_state=SEED)

rf.fit(X_train, y_train)
#Predict the test set lables
y_pred = rf.predict(X_test)
#Evaluate the test set RMSE
accuracy = accuracy_score(y_test, y_pred)
#Print the test set RMSE
print('Accuracy: {:.2f}'.format(accuracy))
```

```
Accuracy: 0.56
```

Ada Boost

```python
#Usamos un AdaBoost
from sklearn.ensemble import AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import roc_auc_score
from sklearn.model_selection import train_test_split

SEED = 1

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.3,
                                                    stratify=y,
                                                    random_state=SEED)

dt = DecisionTreeClassifier(max_depth = 1, random_state=SEED)

adb_clf = AdaBoostClassifier(base_estimator=dt, n_estimators = 100)

adb_clf.fit(X_train, y_train)
#Predecir el test set of probabilities of positive cases.
y_pred_proba = adb_clf.predict_proba(X_test)[:,1]

#adb_clf_roc_auc_score = roc_auc_score(y_test, y_pred_proba)

#print('ROC AUC score:{:.2f}'.format(adb_clf_roc_auc_score))
print(accuracy_score(y_test, y_pred))
```

```
0.5645933014354066
```

Hyperparameter Tunning

```
params_dt = {
              'max_depth': [1,2,3,4,5,6,7,8,9,10,1000],
              'min_samples_leaf': [0.1,0.09,0.08,0.11],
              'max_features': ['log2','sqrt']
            }

grid_dt = GridSearchCV(estimator=dt,
                       param_grid= params_dt,
                       verbose = 1,
                       scoring= 'accuracy',
                       cv=20,
                       n_jobs=-1)
```

```
[259]  1 print(test_acc)

       0.55103668261563
```

```
       1 best_hyperparameters

{'max_depth': 4,
 'max_features': 'log2',
 'min_samples_leaf': 0.1,
 'n_estimators': 400}
```

Resumen

	Inicial	Voting	Bagging	RF	AdaBoost
Intento 1	0.47	0.54	0.531	0.56	0.5645

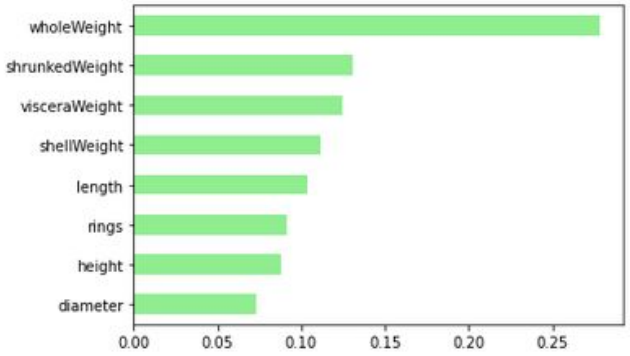
Intento 2

```python
## Importancias
import pandas as pd
import matplotlib.pyplot as plt

#Create Series fo features importances
importances_rf = pd.Series(dt.feature_importances_, index = X.columns)

#Sort importances_rf
sorted_importances_rf = importances_rf.sort_values()

#Make a horizontal bar plot
sorted_importances_rf.plot(kind='barh', color='lightgreen')
plt.show()
```

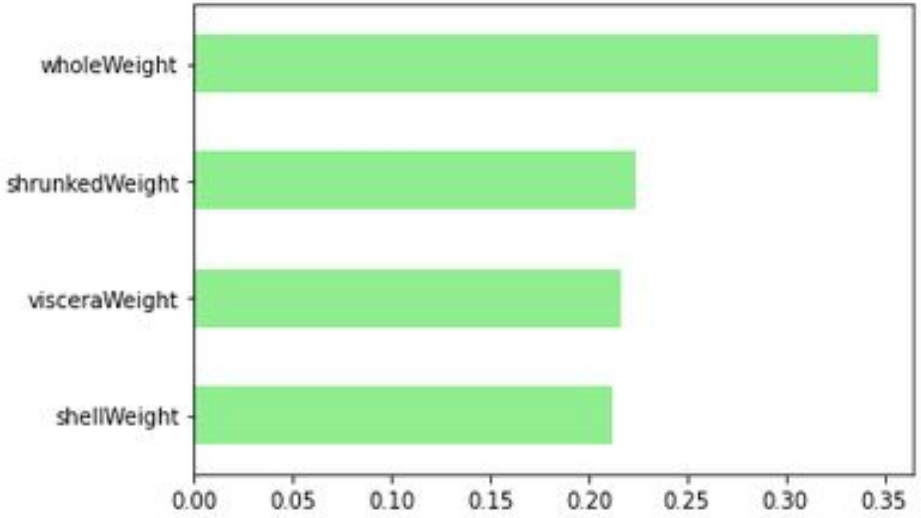



```python
y = db['sex']
X = db[['wholeWeight', 'shrunkedWeight', 'visceraWeight', 'shellWeight']]
```

Resumen

	Inicial	Voting	Bagging	RF	AdaBoost
Intento 1	0.47	0.54	0.531	0.56	0.5645
Intento 2	0.44	0.545	0.52	0.52	0.523
Intento 3					

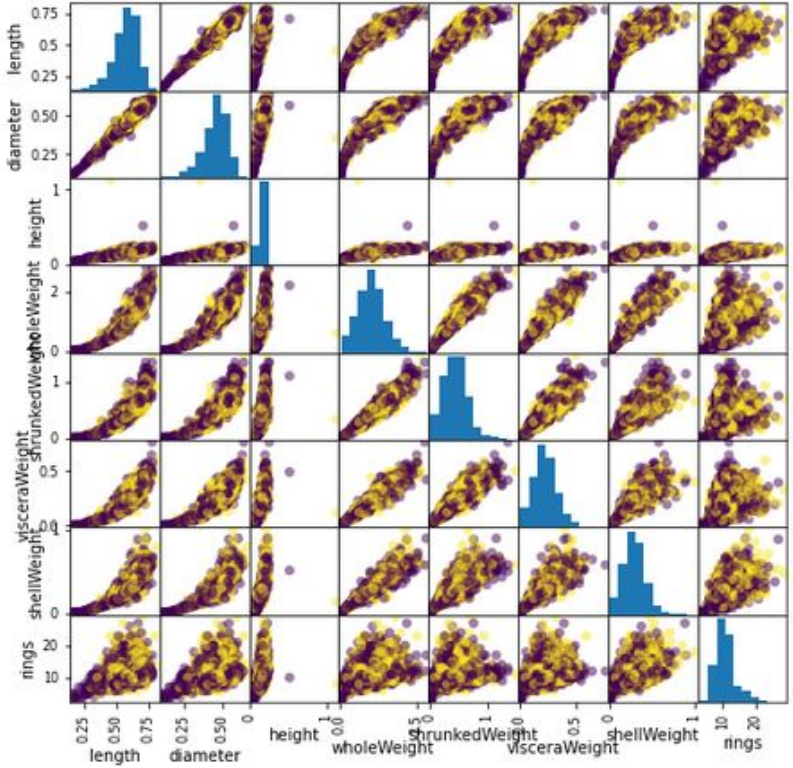
Otro intento



Resumen

	Inicial	Voting	Bagging	RF	AdaBoost
Intento 1	0.47	0.54	0.531	0.56	0.5645
Intento 2	0.44	0.545	0.52	0.52	0.523
Intento 3	0.47	0.532	0.51	0.48	0.4816

Una más


length
diameter
height
wholeWeight
shrunkedWeight
visceraWeight
shellWeight
rings

Resumen

	Inicial	Voting	Bagging	RF	AdaBoost
Intento 1	0.47	0.54	0.531	0.56	0.5645
Intento 2	0.44	0.545	0.52	0.52	0.523
Intento 3	0.47	0.532	0.51	0.48	0.4816
Final	0.52	0.524	0.531	0.56	0.559