**The Abalone Machine Learning Project Report with model, front end and backend code;**

* Abalone
* About dataset
* Project report
* Notebook code
* Flask code
* Frontend code (html bootstrap)

**What is Abalone?**

Abalone is a common name for a group of small to very large sea snails, marine gastropod mollusks in the family Haliotidae. These snails have a large, flattened, ear-shaped shell with a row of holes along the outer edge. The inner surface of the shell is iridescent and is highly prized for its beauty and used in jewelry and decorative items.



Abalone is also a popular food source in many cultures, particularly in Asia and North America. The meat is considered a delicacy and is often used in sushi, salads, and other dishes. Because of its popularity as a food and the high demand for its shells, many species of abalone have been overfished and are now endangered.

**About Dataset:**

The Abalone dataset is a popular machine learning dataset that contains measurements of physical characteristics of abalone, a type of sea snail. The dataset is often used as a benchmark for regression tasks in machine learning.

The dataset includes the following features or variables for each abalone:

Sex: categorical variable (M for male, F for female, and I for infant)

Length: continuous variable representing the longest shell measurement in mm

Diameter: continuous variable representing the diameter of the shell in mm

Height: continuous variable representing the height of the shell in mm

Whole weight: continuous variable representing the weight of the whole abalone in grams

Shucked weight: continuous variable representing the weight of the meat in grams

Viscera weight: continuous variable representing the weight of the gut (after bleeding) in grams

Shell weight: continuous variable representing the weight of the shell in grams

Rings: integer variable representing the age of the abalone (the number of rings on the shell)

The goal of the dataset is to predict the age of the abalone (i.e., the number of rings) based on its physical characteristics. This is a regression task, as the target variable (age) is a continuous variable.

The dataset contains 4,177 instances and has been preprocessed to remove any missing values and to transform the categorical variable (sex) into a set of binary variables (one-hot encoding).

**Project Report:**

1. Introduction

Problem Statement

The objective of this project is to develop a machine learning model to predict the age of abalone based on its physical characteristics. The model will be trained using the Abalone dataset and deployed on a website using Flask and HTML.

Abalone Dataset

The Abalone dataset is a popular machine learning dataset that contains measurements of physical characteristics of abalone, a type of sea snail. The dataset is often used as a benchmark for regression tasks in machine learning.

Objectives

The objectives of this project are:

To preprocess and analyze the Abalone dataset

To develop a machine learning model to predict the age of abalone

To deploy the model on a website using Flask and HTML

2. Data Preprocessing

Loading and Exploring the Dataset

The first step in any machine learning project is to load and explore the dataset. In this project, we will load the Abalone dataset using Python's pandas library and explore its features using descriptive statistics.

Data Cleaning and Handling Missing Values

After exploring the dataset, we will check for any missing values and handle them appropriately. We will also check for any outliers or anomalies in the data and remove or correct them as necessary.

Feature Selection and Transformation

Once the dataset is cleaned, we will select the relevant features for our model and transform them as necessary. This may involve converting categorical variables into numerical variables, scaling the data, or applying other transformations.

Data Visualization to Gain Insights

We will use various data visualization techniques to gain insights into the dataset and understand the relationships between the different features. This will help us select the appropriate machine learning algorithm for our model.

3. Model Development

Splitting the Dataset into Training and Testing Sets

Before developing the machine learning model, we will split the dataset into training and testing sets. The training set will be used to train the model, while the testing set will be used to evaluate its performance.

Selection of a Suitable Machine Learning Algorithm

There are many machine learning algorithms that can be used for regression tasks such as this. We will evaluate the performance of several algorithms and select the one that gives the best results.

Hyperparameter Tuning and Cross-Validation

Once we have selected the machine learning algorithm, we will tune its hyperparameters to optimize its performance. We will also use cross-validation to ensure that our model is not overfitting to the training data.

Model Evaluation and Selection of Metrics

After training the model, we will evaluate its performance using appropriate metrics such as mean squared error or mean absolute error. We will also visualize the results to gain insights into the model's performance.

4. Model Deployment on a Website

Building a Flask Web Application

To deploy the machine learning model on a website, we will use the Flask web framework. We will build a simple web application that allows users to enter the physical characteristics of an abalone and get a prediction of its age.

Creating an HTML Front-End

We will create an HTML front-end for our web application using Bootstrap and JavaScript. The front-end will provide a user-friendly interface for entering data and displaying the results.

5. Conclusion

Summary of Results

In this project, we developed a machine learning model to predict the age of abalone based on its physical characteristics. We preprocessed and analyzed the Abalone dataset, selected a suitable machine learning algorithm, and optimized its hyperparameters using cross-validation. We deployed the model on a website using Flask and HTML, providing a user-friendly interface for users to interact with the model.

**Note Book Code:**

**import** numpy **as** np  
**import** pandas **as** pd

[63]

abalone = pd.read\_csv("abalone.csv")

[64]

abalone.head()

**Ask Six Questions Before moving forward**

[65]

abalone.shape

(4177, 9)

[66]

abalone.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 4177 entries, 0 to 4176  
Data columns (total 9 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 Sex 4177 non-null object   
 1 Length 4177 non-null float64  
 2 Diameter 4177 non-null float64  
 3 Height 4177 non-null float64  
 4 Whole weight 4177 non-null float64  
 5 Shucked weight 4177 non-null float64  
 6 Viscera weight 4177 non-null float64  
 7 Shell weight 4177 non-null float64  
 8 Rings 4177 non-null int64   
dtypes: float64(7), int64(1), object(1)  
memory usage: 293.8+ KB

[67]

abalone.isnull().sum()

Sex 0  
Length 0  
Diameter 0  
Height 0  
Whole weight 0  
Shucked weight 0  
Viscera weight 0  
Shell weight 0  
Rings 0  
dtype: int64

[68]

abalone.duplicated().sum()

0

[69]

abalone.describe()

**Encoding**

[70]

abalone['Sex'].value\_counts()

M 1528  
I 1342  
F 1307  
Name: Sex, dtype: int64

[71]

abalone['Sex'] = abalone['Sex'].map({"M":0,"F":1,"I":2})

[72]

abalone['Sex'].value\_counts()

0 1528  
2 1342  
1 1307  
Name: Sex, dtype: int64

**EDA ( Exploratory data Analysis)**

[76]

corr = abalone.corr()

[77]

**import** seaborn **as** sns  
sns.heatmap(corr,annot=True,cbar=True,cmap='coolwarm')

<AxesSubplot:>

**Distribution of target variable (age)**

[78]

sns.histplot(abalone['Rings'],bins=20)

<AxesSubplot:xlabel='Rings', ylabel='Count'>

[80]

abalone['Rings'].value\_counts()

9 689  
10 634  
8 567  
11 487  
7 391  
12 267  
6 258  
13 203  
14 126  
5 115  
15 103  
16 67  
17 58  
4 57  
18 42  
19 32  
20 26  
3 15  
21 14  
23 9  
22 6  
27 2  
24 2  
1 1  
26 1  
29 1  
2 1  
25 1  
Name: Rings, dtype: int64

**Scatter plot of length vs age**

[82]

sns.scatterplot(x='Length',y='Rings',data=abalone)

<AxesSubplot:xlabel='Length', ylabel='Rings'>

**Train Test Split**

[83]

X = abalone.drop('Rings',axis=1)  
y = abalone['Rings']

[84]

**from** sklearn.model\_selection **import** train\_test\_split

[85]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Standarization the data**

[86]

**from** sklearn.preprocessing **import** StandardScaler  
sc = StandardScaler()  
X\_train\_scaled = sc.fit\_transform(X\_train)  
X\_test\_scaled = sc.transform(X\_test)

[87]

X\_test\_scaled

array([[-1.16554476, 0.54762541, 0.5216659 , ..., -0.14814366,  
 -0.29539849, 0.69031287],  
 [-1.16554476, -1.80260763, -1.65723934, ..., -1.38934271,  
 -1.34175878, -1.37480684],  
 [ 1.25995026, -1.29898627, -1.30253383, ..., -1.21751916,  
 -1.15441916, -1.21650498],  
 ...,  
 [ 1.25995026, -0.92127024, -0.94782833, ..., -0.82865534,  
 -0.92595622, -1.0366165 ],  
 [-1.16554476, 0.84140454, 0.77502697, ..., 1.02070864,  
 1.02511735, 1.69768834],  
 [-1.16554476, -1.29898627, -1.2011894 , ..., -1.12934655,  
 -1.24123508, -1.22370052]])

[88]

X\_test\_scaled

array([[-1.16554476, 0.54762541, 0.5216659 , ..., -0.14814366,  
 -0.29539849, 0.69031287],  
 [-1.16554476, -1.80260763, -1.65723934, ..., -1.38934271,  
 -1.34175878, -1.37480684],  
 [ 1.25995026, -1.29898627, -1.30253383, ..., -1.21751916,  
 -1.15441916, -1.21650498],  
 ...,  
 [ 1.25995026, -0.92127024, -0.94782833, ..., -0.82865534,  
 -0.92595622, -1.0366165 ],  
 [-1.16554476, 0.84140454, 0.77502697, ..., 1.02070864,  
 1.02511735, 1.69768834],  
 [-1.16554476, -1.29898627, -1.2011894 , ..., -1.12934655,  
 -1.24123508, -1.22370052]])

**Training Models**

[89]

**from** sklearn.linear\_model **import** LinearRegression, Ridge, Lasso  
**from** sklearn.tree **import** DecisionTreeRegressor  
**from** sklearn.ensemble **import** RandomForestRegressor  
**from** sklearn.metrics **import** mean\_squared\_error, r2\_score

[90]

*# Define a list of models to train and compare*  
models = [  
    ('Linear Regression', LinearRegression()),  
    ('Ridge Regression', Ridge()),  
    ('Lasso Regression', Lasso()),  
    ('Decision Tree', DecisionTreeRegressor(random\_state=42)),  
    ('Random Forest', RandomForestRegressor(random\_state=42))  
]  
  
*# Train and evaluate each model*  
**for** name, model **in** models:  
    model.fit(X\_train, y\_train)  
    y\_pred = model.predict(X\_test)  
    mse = mean\_squared\_error(y\_test, y\_pred)  
    r2 = r2\_score(y\_test, y\_pred)  
    print(**f**'{name}: MSE = {mse**:.2f**}, R2 = {r2**:.2f**}')

Linear Regression: MSE = 4.96, R2 = 0.56  
Ridge Regression: MSE = 5.07, R2 = 0.56  
Lasso Regression: MSE = 11.41, R2 = -0.00  
Decision Tree: MSE = 9.19, R2 = 0.19  
Random Forest: MSE = 4.99, R2 = 0.56

*# The MSE represents the average squared difference between the predicted and actual values, and a lower MSE indicates better performance.*  
*# The R2 score represents the proportion of variance in the target variable that is predictable from the independent variables, and a higher R2 score indicates better performance.*

**Chosen Model**

[95]

dtr = DecisionTreeRegressor()  
dtr.fit(X\_train, y\_train)  
y\_pred = dtr.predict(X\_test)  
print(mean\_squared\_error(y\_test, y\_pred))  
print(r2\_score(y\_test,y\_pred))

9.026347305389221  
0.2089537099162797

**Prediction System**

[101]

**def** prediction\_age(Sex,Length,Diameter,Height,Whole\_weght,shucked\_weght,visc\_wet,shell\_weight):  
    features = np.array([[Sex,Length,Diameter,Height,Whole\_weght,shucked\_weght,visc\_wet,shell\_weight]])  
      
    pred = dtr.predict(features).reshape(1,-1)  
      
    **return** pred[0]  
  
  
Sex = 2  
Length = 8.0  
Diameter = 4.0  
Height = 6.0  
Whole\_weght = 10.0  
shucked\_weght = 20.0  
visc\_wet = 20.0  
shell\_weight = 15.0  
  
  
prediciton = prediction\_age(Sex,Length,Diameter,Height,Whole\_weght,shucked\_weght,visc\_wet,shell\_weight)  
  
**if** prediciton[0] == 0:  
    print("{} is a Male".format(prediciton))  
**elif** prediciton[0] == 1:  
    print("{} is a Female".format(prediciton))  
**else**:  
    print("{} is a Ifant".format(prediciton))

[14.] is a Ifant

C:\Users\Noor Saeed\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but DecisionTreeRegressor was fitted with feature names  
 warnings.warn(

[102]

**import** pickle  
pickle.dump(dtr,open('model.pkl','wb'))

**Flask Code**

from flask import Flask**,**request**,** render\_template  
import numpy as np  
import pandas as pd  
import pickle  
  
  
# load modle  
model = pickle.load(open('model.pkl'**,**'rb'))  
  
#create app  
app = Flask(\_\_name\_\_)  
  
@app.route('/')  
def index():  
 return render\_template('index.html')  
  
@app.route('/predict'**,**methods=['POST'])  
def predict():  
 # sex, length, diameter, height, wholeWeight, Shuckedweight, Visceraweight, Shellweight  
 sex = int(request.form['sex'])  
 length = float(request.form['length'])  
 diameter = float(request.form['diameter'])  
 height = float(request.form['height'])  
 wholeWeight = float(request.form['wholeWeight'])  
 Shuckedweight = float(request.form['Shuckedweight'])  
 Visceraweight = float(request.form['Visceraweight'])  
 Shellweight = float(request.form['Shellweight'])  
  
 features = np.array([[sex**,** length**,** diameter**,** height**,** wholeWeight**,** Shuckedweight**,** Visceraweight**,** Shellweight]])  
  
 age = model.predict(features).reshape(**1,**-**1**)[**0**]  
 return render\_template('index.html'**,**age = age)  
  
# python main  
if \_\_name\_\_ == "\_\_main\_\_":  
 app.run(debug=True)

**Front End (HTML Bootstrap)**

<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <meta name="viewport" content="width=device-width, initial-scale=1.0">  
 <meta http-equiv="X-UA-Compatible" content="ie=edge">  
 <title>Abalone Age Prediction</title>  
  
 <!-- Bootstrap CSS -->  
 <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css"  
 integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T"  
 crossorigin="anonymous">  
  
</head>  
  
<style>  
 body{  
 background:black;  
 color:white;  
 }  
 .container{  
 background: yellow;  
 color: black;  
 border-radius:15px;  
 }  
 .container h1{  
 text-align:center;  
 }  
 .btn{  
 margin-bottom:10px;  
 margin-left:38%;  
 }  
 .img{  
 width:50%;  
 height:150px;  
 border-radius:30px;  
 margin-left:22%;  
 }  
 .card{  
 background: black;  
 color: white;  
 margin-left:33%;  
 margin-bottom:10px  
 }  
</style>  
  
<body>  
<nav class="navbar navbar-expand-lg navbar-light bg-light">  
 <div class="container-fluid">  
 <a class="navbar-brand" href="#">Abalone Age</a>  
 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">  
 <span class="navbar-toggler-icon"></span>  
 </button>  
 <div class="collapse navbar-collapse" id="navbarSupportedContent">  
 <ul class="navbar-nav me-auto mb-2 mb-lg-0">  
 <li class="nav-item">  
 <a class="nav-link active" aria-current="page" href="#">Home</a>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link" href="#">Link</a>  
 </li>  
 <li class="nav-item dropdown">  
 <a class="nav-link dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown" aria-expanded="false">  
 Dropdown  
 </a>  
 <ul class="dropdown-menu">  
 <li><a class="dropdown-item" href="#">Action</a></li>  
 <li><a class="dropdown-item" href="#">Another action</a></li>  
 <li><hr class="dropdown-divider"></li>  
 <li><a class="dropdown-item" href="#">Something else here</a></li>  
 </ul>  
 </li>  
 <li class="nav-item">  
 <a class="nav-link disabled">Disabled</a>  
 </li>  
 </ul>  
 <form class="d-flex" role="search">  
 <input class="form-control me-2" type="search" placeholder="Search" aria-label="Search">  
 <button class="btn btn-outline-success" type="submit">Search</button>  
 </form>  
 </div>  
 </div>  
</nav>  
  
<!--==========================================================-->  
  
<div class="container my-3 mt-3">  
 <h1 class="text-success">Abalone Age Prediction Model</h1>  
 <img class='img' src="{{url\_for('static',filename='img.jpg')}}" alt="">  
  
  
 <form action="/predict" method="POST">  
 <div class="row">  
 <div class="form-group col-md-4">  
 <label for="sex">Sex</label>  
 <select class="form-control" id="sex" name="sex">  
 <option value="0">Male</option>  
 <option value="1">Female</option>  
 <option value="2" selected>Infant</option>  
 </select>  
 </div>  
 <div class="form-group col-md-4">  
 <label for="length">Length</label>  
 <input type="number" class="form-control" id="length" name="length" step="0.01" value="0">  
 </div>  
 <div class="form-group col-md-4">  
 <label for="diameter">Diameter</label>  
 <input type="number" class="form-control" id="diameter" name="diameter" step="0.01" value="0">  
 </div>  
 </div>  
  
 <div class="row">  
 <div class="form-group col-md-4">  
 <label for="height">Height</label>  
 <input type="number" class="form-control" id="height" name="height" step="0.01" value="0">  
 </div>  
 <div class="form-group col-md-4">  
 <label for="whole Weight">Whole Weight</label>  
 <input type="number" class="form-control" id="wholeWeight" name="wholeWeight" step="0.01" value="0">  
 </div>  
 <div class="form-group col-md-4">  
 <label for="Shucked weight">Shucked weight</label>  
 <input type="number" class="form-control" id="Shuckedweight" name="Shuckedweight" step="0.01" value="0">  
 </div>  
 </div>  
 <div class="row">  
 <div class="form-group col-md-4">  
 <label for="Viscera weight">Viscera weight</label>  
 <input type="number" class="form-control" id="Visceraweight" name="Visceraweight" step="0.01" value="0">  
 </div>  
 <div class="form-group col-md-4">  
 <label for="Shell weight">Shell weight</label>  
 <input type="number" class="form-control" id="Shellweight" name="Shellweight" step="0.01" value="0">  
 </div>  
 </div>  
  
 <button type="submit" class="btn btn-primary btn-lg">Get Abalone Age</button>  
 </form>  
  
  
 {% if age %}  
 <div class="card" style="width: 18rem;">  
 <img src="{{url\_for('static', filename='img.jpg')}}" class="card-img-top" alt="...">  
 <div class="card-body">  
 <h5 class="card-title">Abalone Age: {{age}}</h5>  
 <p class="card-text">On the Basis of your input we recommend you the best possible age of this abalone</p>  
 </div>  
 </div>  
  
 {% else %}  
 <p>sorry there was an error</p>  
 {% endif %}  
</div>  
  
  
</body>  
</html>