Report

ARTIFICIAL INTELLIGENCE

Problem: Abalone Age Prediction Using Various Machine Learning Algorithms

Introduction

The task of predicting the age of abalone through their physical measurements is an important problem in marine biology and aquaculture. Accurate age prediction can assist in better understanding abalone growth patterns, managing abalone populations, and optimizing aquaculture practices. In this report, we evaluate the performance of four different machine learning algorithms in predicting the age of abalone: Linear Regression, Decision Tree Regressor, Random Forest Regressor, and Support Vector Regressor (SVR).

Methodology:

Dataset:

The dataset used in this analysis is the Abalone.data.csv taken from the Kaggle, which contains physical measurements of abalones and their corresponding ages (measured by the number of rings). The dataset includes the following attributes:

- Sex (M, F, I)
- Length
- Diameter
- Height
- Whole weight
- Shucked weight
- Viscera weight
- Shell weight
- Rings (target variable)

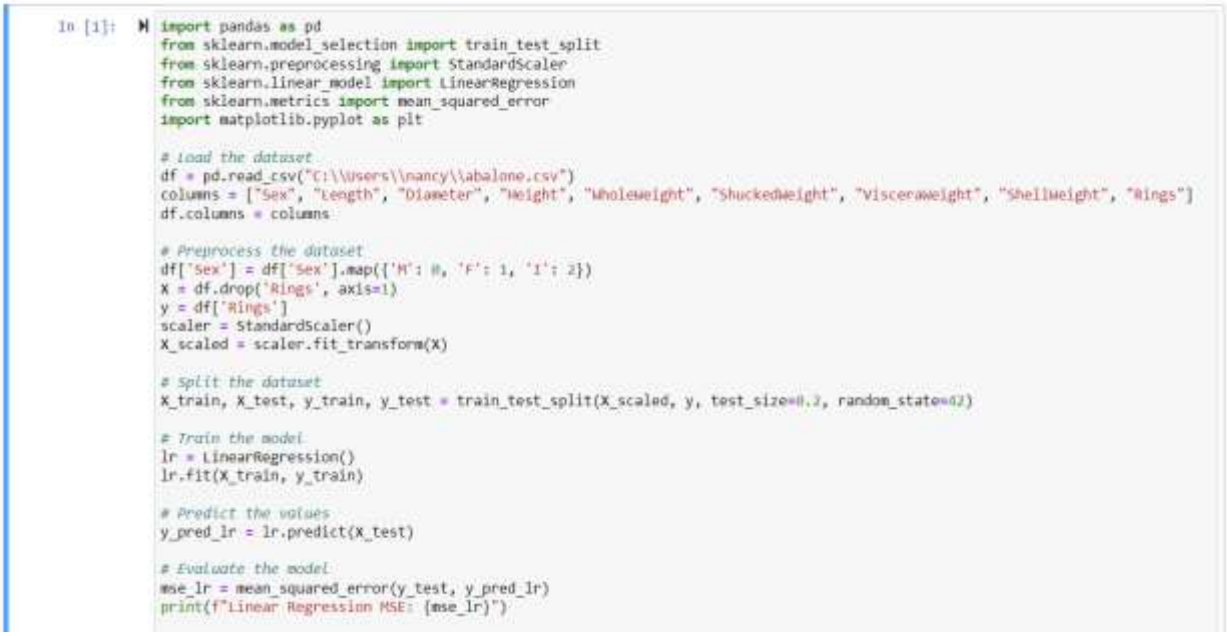
Data Preprocessing:

1. **Loading Data**: The dataset was loaded into a pandas DataFrame.
2. **Encoding Categorical Features**: The categorical feature 'Sex' was encoded into numerical values (M: 0, F: 1, I: 2).
3. **Scaling Features**: All features were scaled using StandardScaler to normalize the data.
4. **Splitting Data**: The dataset was split into training (80%) and testing (20%) sets.

Machine Learning Algorithms:

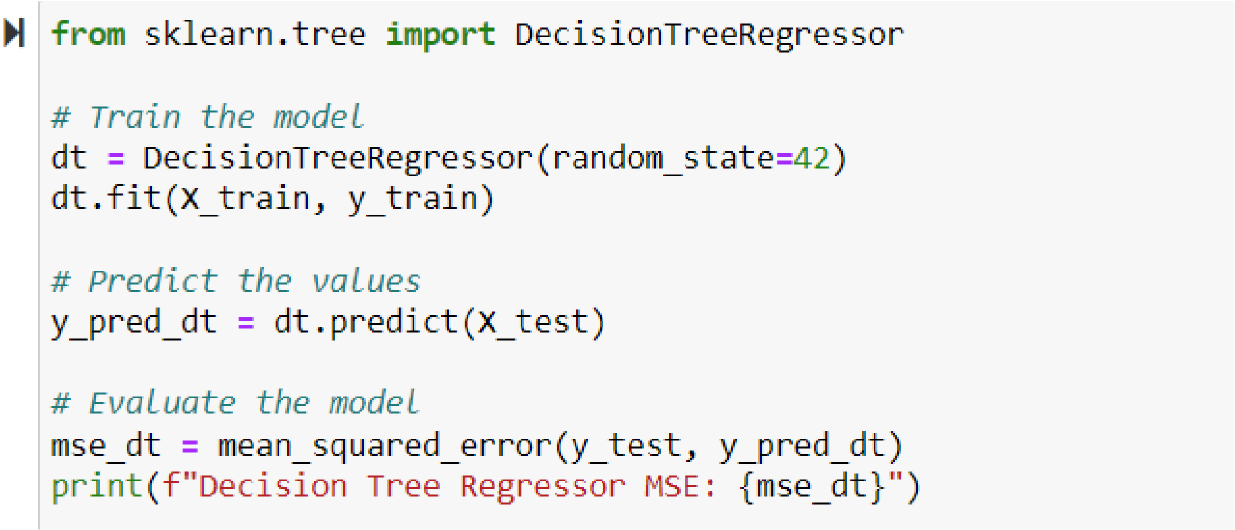
The following machine learning algorithms were applied to predict the age of abalone:

1. **Linear Regression**



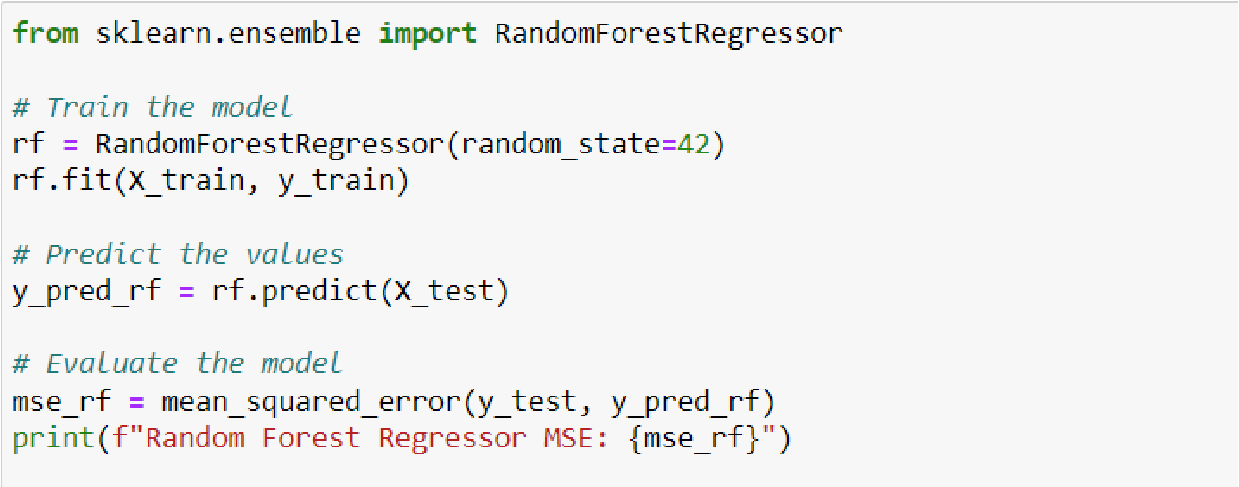
```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

# Load the dataset
df = pd.read_csv("C:\\Users\\nancy\\abalone.csv")
columns = ["Sex", "Length", "Diameter", "Height", "WholeWeight", "ShuckedWeight", "VisceraWeight", "ShellWeight", "Rings"]
df.columns = columns

# Preprocess the dataset
df['Sex'] = df['Sex'].map({'M': 0, 'F': 1, 'I': 2})
X = df.drop('Rings', axis=1)
y = df['Rings']
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Train the model
lr = LinearRegression()
lr.fit(X_train, y_train)

# Predict the values
y_pred_lr = lr.predict(X_test)

# Evaluate the model
mse_lr = mean_squared_error(y_test, y_pred_lr)
print(f"Linear Regression MSE: {mse_lr}")
```

2. **Decision Tree Regressor**



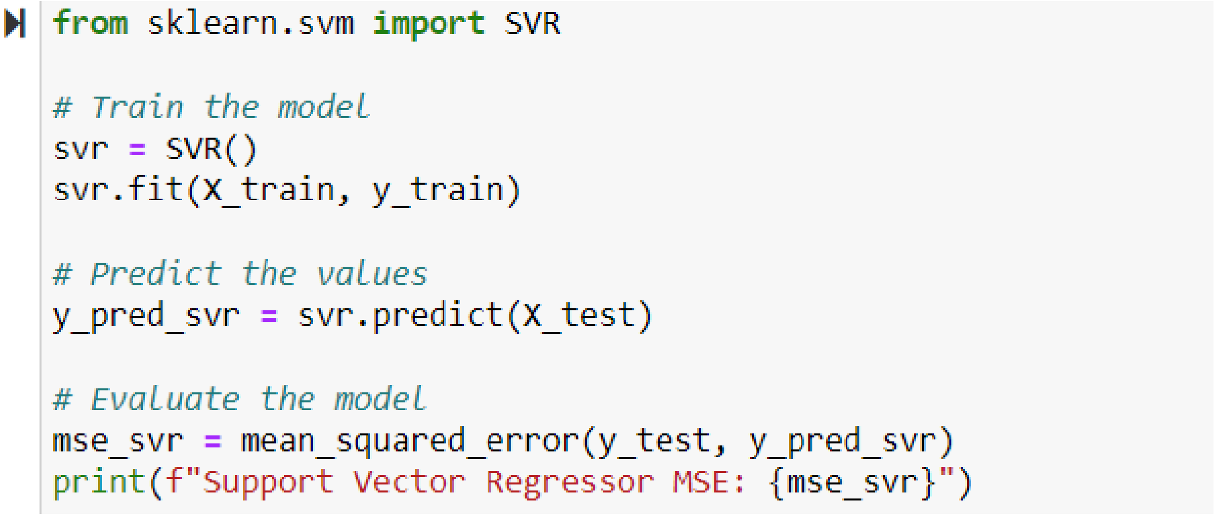
```python
from sklearn.tree import DecisionTreeRegressor

# Train the model
dt = DecisionTreeRegressor(random_state=42)
dt.fit(X_train, y_train)

# Predict the values
y_pred_dt = dt.predict(X_test)

# Evaluate the model
mse_dt = mean_squared_error(y_test, y_pred_dt)
print(f"Decision Tree Regressor MSE: {mse_dt}")
```

3. **Random Forest Regressor**



```
from sklearn.ensemble import RandomForestRegressor

# Train the model
rf = RandomForestRegressor(random_state=42)
rf.fit(X_train, y_train)

# Predict the values
y_pred_rf = rf.predict(X_test)

# Evaluate the model
mse_rf = mean_squared_error(y_test, y_pred_rf)
print(f"Random Forest Regressor MSE: {mse_rf}")
```

4. **Support Vector Regressor (SVR)**



```
from sklearn.svm import SVR

# Train the model
svr = SVR()
svr.fit(X_train, y_train)

# Predict the values
y_pred_svr = svr.predict(X_test)

# Evaluate the model
mse_svr = mean_squared_error(y_test, y_pred_svr)
print(f"Support Vector Regressor MSE: {mse_svr}")
```

Performance Metric:

The performance of each algorithm was evaluated using Mean Squared Error (MSE). Lower MSE values indicate better model performance.

Results:

The MSE values for each model are as follows:

- **Linear Regression**: 4.950
- **Decision Tree Regressor**: 9.049
- **Random Forest Regressor**: 5.067
- **Support Vector Regressor (SVR)**: 4.883

Based on the Mean Squared Error (MSE) values, we can conclude the following:

- The **Support Vector Regressor (SVR)** performed the best with the lowest MSE of 4.883.
- The **Linear Regression** model also performed well with an MSE of 4.950.
- The **Random Forest Regressor** had a slightly higher MSE of 5.067.
- The **Decision Tree Regressor** had the highest MSE of 9.049, indicating lower performance compared to the other models.

Conclusion:

In conclusion, the Support Vector Regressor (SVR) is the most effective algorithm for predicting the age of abalone in this dataset, followed closely by Linear Regression. The Decision Tree Regressor, while useful, showed the least accuracy among the evaluated models. Future work could explore further tuning of these models or try additional algorithms to improve prediction accuracy.