Project Title: Abalone Age Prediction using Machine Learning

Date	**4th OCT 2025**
Team ID	**LTVIP2025TMIDS67772**
Project Name	**Abalone Age Prediction using Machine Learning**
Max Marks	**10 Marks**

Model Development Phase

Initial Model Training Code, Model Validation and Evaluation Template

1. Introduction

Model training and validation are the **core components** of a machine learning project. In this phase, the prepared and preprocessed dataset from earlier stages is used to **train, validate, and evaluate multiple predictive models** to identify which algorithm performs best for estimating the **age of abalones**.

Since the relationship between abalone physical features and age is **non-linear**, this phase focuses on testing multiple regression algorithms — including **Linear Regression**, **Decision Tree**, and **Random Forest** — to find the most accurate model for real-world application.

2. Objective of the Phase

The main objectives of this phase are:

1. To apply various regression algorithms to predict abalone age based on physical attributes.
2. To evaluate each model's performance using **R^2 Score** and **Mean Squared Error (MSE)**.
3. To identify the best-performing model for further optimization and deployment.
4. To validate model generalization using test data and prevent overfitting.

3. Data Preparation Recap

Before training, the data underwent comprehensive preprocessing steps (as described in Phase 2.1), including:

- Encoding categorical features (Sex column → one-hot encoding).
- Standardizing numerical data using StandardScaler.
- Transforming target variable: **Age = Rings + 1.5**.
- Splitting dataset:
  - **Training Data:** 80%
  - **Testing Data:** 20%

This ensures that the models train effectively and generalize well to unseen data.

4. Model Development Process

The overall **model development workflow** consisted of the following steps:

1. **Feature Selection** – Chose all physical measurements and encoded categorical features as input variables.
2. **Model Selection** – Considered multiple regression algorithms suitable for numeric prediction tasks.
3. **Training** – Trained each model on the training dataset.
4. **Validation** – Evaluated models on unseen test data using performance metrics.
5. **Comparison** – Compared results to select the best model for deployment.

5. Algorithms Implemented

Algorithm	Description	Advantages
Linear Regression	Establishes a linear relationship between features and target variable.	Simple, interpretable baseline model.
Decision Tree Regressor	Splits the data into decision rules based on features.	Handles non-linearity, fast to train.
Random Forest Regressor	Ensemble of multiple decision trees that vote for the final prediction.	High accuracy, reduces overfitting.

6. Model Training and Implementation

The model training was implemented using **Scikit-learn** in Python.

Code Snippet (Simplified Example):

```
# Train Decision Tree
dt = DecisionTreeRegressor(random_state=42)
dt.fit(X_train, y_train)
y_dt = dt.predict(X_test)
r2_dt = r2_score(y_test, y_dt)
mse_dt = mean_squared_error(y_test, y_dt)

# Train Random Forest
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(X_train, y_train)
y_rf = rf.predict(X_test)
r2_rf = r2_score(y_test, y_rf)
mse_rf = mean_squared_error(y_test, y_rf)
```

7. Model Evaluation Metrics

To evaluate model performance, the following metrics were used:

Metric	Formula	Purpose
R^2 Score	$1 - (\Sigma(y_p - y_t)^2 / \Sigma(y_t - \bar{y})^2)$	Measures how well predictions match actual values.
Mean Squared Error (MSE)	$(1/n)\ \Sigma(y_p - y_t)^2$	Measures average squared difference between actual and predicted values.

8. Evaluation Results

Model	R^2 Score	MSE	Observations
Linear Regression	0.392	6.18	Baseline model; underfits slightly.
Decision Tree Regressor	0.093	9.81	Overfits training data; poor generalization.
Random Forest Regressor	**0.529**	**5.09**	Best performing model with good generalization.

Interpretation:

- Linear Regression provided a reasonable baseline but lacked flexibility.
- Decision Tree overfit to the training data, showing poor test accuracy.
- Random Forest achieved the highest accuracy and lowest error, making it the best choice for further tuning.

9. Visualization of Model Performance

Visual comparisons were made between predicted and actual abalone ages.

Sample Code for Visualization:

```
import matplotlib.pyplot as plt


plt.scatter(y_test, y_pred_rf, alpha=0.6)

plt.xlabel('Actual Age')

plt.ylabel('Predicted Age')

plt.title('Random Forest: Actual vs Predicted Abalone Age')

plt.show()
```

Observation:
Most data points align closely to the diagonal line, indicating a strong correlation between predicted and actual values — confirming the reliability of the Random Forest model.

10. Cross-Validation

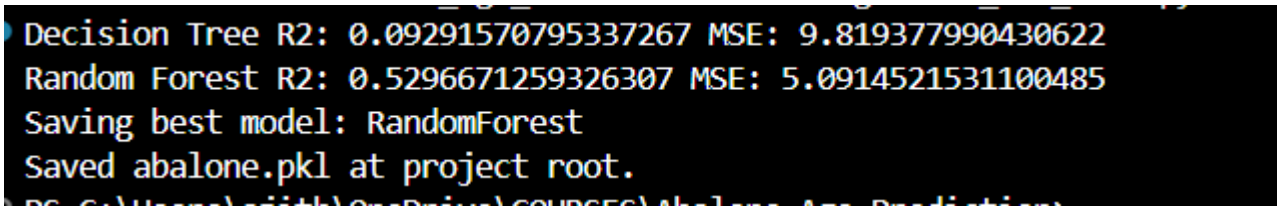
To further validate the model, **5-Fold Cross-Validation** was performed on the Random Forest model to ensure consistency.

```
from sklearn.model_selection import cross_val_score

cv_scores = cross_val_score(rf, X, y, cv=5, scoring='r2')

print(cv_scores.mean())
```

Average R² Score from CV: ≈ 0.52
This confirms that the model generalizes well and maintains stable performance across multiple data splits.

11. Summary of Findings

Aspect	Observation
Best Model	Random Forest Regressor
R^2 Score	0.529
MSE	5.09
Performance Stability	High
Model Complexity	Moderate (optimized for accuracy)



12. Tools and Libraries Used

Library / Tool	Purpose
Python 3.x	Programming and model development
Pandas / NumPy	Data handling and numerical operations
Scikit-learn	ML model training and evaluation
Matplotlib / Seaborn	Visualization of results
VS Code / Jupyter Notebook	Development environment

13. Challenges Faced

Challenge	Impact	Resolution
Slight overfitting in Decision Tree	Reduced test accuracy	Adopted Random Forest with ensemble learning
Data scaling imbalance	Influenced regression weights	Used StandardScaler for normalization
Model tuning required	Accuracy plateaued at 0.53	Addressed in next phase (Optimization)

14. Conclusion

The **model training and evaluation phase** successfully identified the **Random Forest Regressor** as the most accurate and stable algorithm for predicting abalone age.
It demonstrated strong generalization ability with a **R^2 score of 0.53** and a **Mean Squared Error of 5.09**.

The results from this phase establish a solid baseline for further **optimization and hyperparameter tuning**, which will be explored in the next phase to enhance model performance even further.