Model Development Phase Template

Date	8 July 2024
Team ID	SWTID1720104754
Project Title	Cereal Analysis Based On Rating By Using Machine Learning Techniques
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include classification reports, accuracy, and confusion matrices for multiple models, presented through respective screenshots.

Initial Model Training Code:

```
# Train linear regression models
lr = LinearRegression()
r = Ridge(alpha=1.5)
l = Lasso(alpha=0.001)
lr.fit(x_train, y_train)
r.fit(x_train, y_train)
l.fit(x_train, y_train)
```

```
Lasso
Lasso(alpha=0.001)
```

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_percentage_error
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.2,random_state = 42)
#random_state acts as the seed for the random number generator during the split
```

```
# Train decision tree regressor
dt = DecisionTreeRegressor()
dt.fit(x_train, y_train)
```

```
DecisionTreeRegressor
DecisionTreeRegressor()
```

```
# Train random forest regressor
rf = RandomForestRegressor(n_estimators=100, random_state=42)
rf.fit(x_train, y_train)
```

```
C:\Users\prade\AppData\Local\Programs\Python\Python312\Lib\site-pa
a 1d array was expected. Please change the shape of y to (n_sampl
  return fit_method(estimator, *args, **kwargs)
```

```
RandomForestRegressor
RandomForestRegressor(random_state=42)
```

```
lr.score(x_test,y_test)
```

```
0.9999999999999994
```

```
r.score(x_test,y_test)
```

```
0.9940890789552553
```

```
l.score(x_test,y_test)
```

```
0.9999663757215922
```

```
dt.score(x_test,y_test)
```

```
0.7235434757748809
```

```
rf.score(x_test,y_test)
```

```
0.7840688820583703
```

```
# Make predictions on test set
y_pred_lr = lr.predict(x_test)
y_pred_r = r.predict(x_test)
y_pred_l = l.predict(x_test)
y_pred_dt = dt.predict(x_test)
y_pred_rf = rf.predict(x_test)
```

```
# Calculate evaluation metrics for each model
models = ["Linear Regression", "Ridge Regression", "Lasso Regression",
          "Decision Tree Regressor", "Random Forest Regressor"]
y_preds = [y_pred_lr, y_pred_r, y_pred_l, y_pred_dt, y_pred_rf]
```

```
# Evaluate models
print(f"Linear Regression score: {lr.score(x_test, y_test):.4f}")
print(f"Ridge Regression score: {r.score(x_test, y_test):.4f}")
print(f"Lasso Regression score: {l.score(x_test, y_test):.4f}")
print(f"Decision Tree Regressor score: {dt.score(x_test, y_test):.4f}")
print(f"Random Forest Regressor score: {rf.score(x_test, y_test):.4f}")
```

```
Linear Regression score: 1.0000
Ridge Regression score: 0.9941
Lasso Regression score: 1.0000
Decision Tree Regressor score: 0.7235
Random Forest Regressor score: 0.7841
```

```
# Save the models as pickle files
with open('linear_regression.pkl', 'wb') as f:
    pickle.dump(lr, f)

with open('ridge_regression.pkl', 'wb') as f:
    pickle.dump(r, f)

with open('lasso_regression.pkl', 'wb') as f:
    pickle.dump(l, f)

with open('decision_tree_regressor.pkl', 'wb') as f:
    pickle.dump(dt, f)

with open('random_forest_regressor.pkl', 'wb') as f:
    pickle.dump(rf, f)
```

```
for model, y_pred in zip(models, y_preds):
    r2 = r2_score(y_test, y_pred)
    rmse = mean_squared_error(y_test, y_pred, squared=False)  # Square root for interpretability
    mape = mean_absolute_percentage_error(y_test, y_pred) * 100  # Percentage error

    print(f"\nModel: {model}")
    print(f"R-squared: {r2:.4f}")
    print(f"Root Mean Squared Error (RMSE): {rmse:.4f}")
    print(f"Mean Absolute Percentage Error (MAPE): {mape:.4f}%")
```

```
Model: Linear Regression
R-squared: 1.0000
Root Mean Squared Error (RMSE): 0.0000
Mean Absolute Percentage Error (MAPE): 0.0000%

Model: Ridge Regression
R-squared: 0.9941
Root Mean Squared Error (RMSE): 1.1395
Mean Absolute Percentage Error (MAPE): 2.0762%

Model: Lasso Regression
R-squared: 1.0000
Root Mean Squared Error (RMSE): 0.0859
Mean Absolute Percentage Error (MAPE): 0.1717%

Model: Decision Tree Regressor
R-squared: 0.7235
Root Mean Squared Error (RMSE): 7.7926
Mean Absolute Percentage Error (MAPE): 16.3354%

Model: Random Forest Regressor
R-squared: 0.7841
Root Mean Squared Error (RMSE): 6.8870
Mean Absolute Percentage Error (MAPE): 16.9964%
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Decision Tree			