Model Development Phase Template

Date	8 July 2024
Team ID	740138
Project Title	Identification Of Methodology Used In Real Estate Property Valuation
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:

REGRESSION MODELS

```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
lr = LinearRegression()
lr.fit(x_train_scaled, y_train)
y_pred_lr = lr.predict(x_test_scaled)
```

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=20)

mlr = LinearRegression()

mlr.fit(x_train, y_train)

y_pred_mlr = mlr.predict(x_test)
```

```
from sklearn.tree import DecisionTreeRegressor
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=20)
```

```
dtr=DecisionTreeRegressor(random_state=20)
dtr.fit(x_train,y_train)
```

```
DecisionTreeRegressor(random_state=20)
```

```
y_pred_dtr=dtr.predict(x_test)
```

```
from sklearn.ensemble import RandomForestRegressor
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=20)
rf=RandomForestRegressor(n_estimators=20,random_state=80)
rf.fit(x_train,y_train)
```

```
RandomForestRegressor(n_estimators=20, random_state=80)
```

```
y_pred_rf=rf.predict(x_test)
```

```
import xgboost as xg
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=20)
xgreg.XGBRegressor(objective='reg:linear',n_estimators=50,seed=23)
```

```
xg.fit(x_train,y_train)
```

```
xg.fit(x_train,y_train)
```

```
XGBRegressor(base_score=None, booster=None, callbacks=None,
             colsample_bylevel=None, colsample_bynode=None,
             colsample_bytree=None, device=None, early_stopping_rounds=None,
             enable_categorical=False, eval_metric=None, feature_types=None,
             gamma=None, grow_policy=None, importance_type=None,
             interaction_constraints=None, learning_rate=None, max_bin=None,
             max_cat_threshold=None, max_cat_to_onehot=None,
             max_delta_step=None, max_depth=None, max_leaves=None,
             min_child_weight=None, missing=nan, monotone_constraints=None,
             multi_strategy=None, n_estimators=50, n_jobs=None,
             num_parallel_tree=None, objective='reg:linear', ...)
```

```
[ ] y_pred_xg=xg.predict(x_test)
```

```
[ ] from sklearn.ensemble import GradientBoostingRegressor
```

```
[ ] gbr=GradientBoostingRegressor(n_estimators=10,max_depth=1,learning_rate=1)
```

```
[ ] x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=20)
```

```
[ ] gbr.fit(x_train,y_train)
```

```
GradientBoostingRegressor(learning_rate=1, n_estimators=10)
```

```
[ ] y_pred_gbr=gbr.predict(x_test)
```

```
[ ] from sklearn.ensemble import AdaBoostRegressor
```

```
[ ] adr=AdaBoostRegressor(n_estimators=10,learning_rate=1,random_state=20)
    x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=20)
```

```
adr.fit(x_train,y_train)
```

```
AdaBoostRegressor(learning_rate=1, n_estimators=10, random_state=20)
```

```
[ ] y_pred_adr=adr.predict(x_test)
```

COMPARE ALL MODELS

```
[ ] print("The accuracy of Linear Regression:",r2_score(y_pred_lr,y_test))
    print("The accuracy of multilinear regression:",r2_score(y_pred_mlr,y_test))
    print("The accuracy of Decision Tree Regression:",r2_score(y_pred_dtr,y_test))
    print("The accuracy of Random Forest Regression",r2_score(y_pred_rf,y_test))
    print("The accuracy of XGBoost Regression",r2_score(y_pred_xg,y_test))
    print("The accuracy of Gradient Boosting Regression",r2_score(y_pred_gbr,y_test))
    print("The accuracy of Adaboost Regression",r2_score(y_pred_adr,y_test))
```

```
The accuracy of Linear Regression: 0.4188466128748409
The accuracy of multilinear regression: 0.4188466128747596
The accuracy of Decision Tree Regression: 0.6978919103505941
The accuracy of Random Forest Regression 0.7082737310455957
The accuracy of XGBoost Regression 0.7337036671657219
The accuracy of Gradient Boosting Regression 0.5931610408263717
The accuracy of Adaboost Regression 0.6891654737766852
```

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Confusion Matrix
Decision Tree Regression	The accuracy of Decision Tree Regression: 0.69709	69%	-
Random forest regression	The accuracy of Random Forest Regression 0.78827373	78%	-
Linear regression	The accuracy of Linear Regression: 0.41804661	41%	-
Ada boost regression	The accuracy of Adaboost Regression 0.68936347	68%	-
Xgboost regression	The accuracy of XGBoost Regression 0.733703667165721	73%	-
Multi linear regression	The accuracy of multilinear regression: 0.418046	41%	