

Model Development Phase Template

Date	08 May 2024
Team ID	722312
Project Title	Walmart Sales Analysis For Retail Industry With Machine Learning
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:

Paste the screenshot of the model training code

Model Validation and Evaluation Report:

Model	Classification Report	Accuracy	Correlation Matrix
-------	-----------------------	----------	--------------------

Random Forest	<p>Screenshot of the classification report</p> <pre> from sklearn.ensemble import RandomForestRegressor rf=RandomForestRegressor(n_estimators=150,max_depth=30,min_samples_split=5,min_samples_leaf=1) rf.fit(x_train,y_train.ravel()) # RandomForestRegressor(max_depth=30, min_samples_split=5, n_estimators=150) Testing_Accuracy_rf=rf.score(x_test,y_test.ravel())*100 print('Testing Accuracy:',Testing_Accuracy_rf,'%') Testing Accuracy: 96.74488552124691 % y_pred=rf.predict(x_test) rms_rf=mean_squared_error(y_test,y_pred,squared=False) print('RMSE:',rms_rf) RMSE 4118.162627047973 MAE_rf=mean_absolute_error(y_test,y_pred) print('MAE:',MAE_rf) MAE 1627.843455788072 Training_Accuracy_rf=rf.score(x_train,y_train.ravel())*100 print('Training Accuracy:',Training_Accuracy_rf,'%') Training Accuracy 99.07801026679833 % </pre>	<p>Accuracy Value: 96.74488552124691 %</p>	<p>Screenshot of the confusion matrix</p> <pre> correlation = np.corrcoef(y_test.ravel(), y_pred)[0, 1] print("Correlation between y_train and y_pred:", correlation) Correlation between y_train and y_pred: 0.9835904953714786 </pre>
Decision Tree	<p>Screenshot of the classification report</p> <pre> # DecisionTreeRegressor DecisionTreeRegressor(random_state=0) y_pred=dtr.predict(x_test) Testing_Accuracy_dtr=dtr.score(x_test,y_test.ravel())*100 print('Testing Accuracy:',Testing_Accuracy_dtr,'%') Testing Accuracy: 94.55987947188075 % rms_dtr=mean_squared_error(y_test,y_pred,squared=False) print('RMSE:',rms_dtr) RMSE 5323.835409637849 MAE_dtr=mean_absolute_error(y_test,y_pred) print('MAE:',MAE_dtr) MAE: 2068.9108296751015 Training_Accuracy_dtr=rf.score(x_train,y_train.ravel())*100 print('Training Accuracy:',Training_Accuracy_dtr,'%') Training Accuracy 99.08201396703826 % </pre>	<p>Accuracy Value: 94.55987947188075 %</p>	<p>Screenshot of the confusion matrix</p> <pre> correlation = np.corrcoef(y_test.ravel(), y_pred)[0, 1] print("Correlation between y_train and y_pred:", correlation) Correlation between y_train and y_pred: 0.9727406145594049 </pre>
XgBoost	<p>Screenshot of the classification report</p> <pre> import xgboost as xgb import warnings xg_reg=xgb.XGBRegressor(objective='reg:squarederror',n_estimators=100,max_depth=4,learning_rate=0.3) xg_reg.fit(x_train,y_train) # XGBRegressor XGBRegressor(base_score=0.5, booster=None, callbacks=None, colsample_bynode=None, colsample_byrow=None, columnsample_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=0.3, max_bin=None, max_cat_threshold=None, max_cat_to_onehot=None, max_delta_step=None, max_depth=4, max_leaves=None, min_child_weight=None, missingnan, monotone_constraints=None, multi_valuedness=None, num_parallel_tree=0, n_jobs=None, nthread=None, num_parallel_tree=None, ...): preds=xg_reg.predict(x_test) y_pred=xg_reg.predict(x_test) Testing_Accuracy_xg_reg=xg_reg.score(x_test,y_test.ravel())*100 print('Testing Accuracy:',Testing_Accuracy_xg_reg,'%') Testing Accuracy: 94.12206327106377 % rms_xg_reg=mean_squared_error(y_test,y_pred,squared=False) print('RMSE:',rms_xg_reg) RMSE 5533.919190794416 MAE_xg_reg=mean_absolute_error(y_test,y_pred) print('MAE:',MAE_xg_reg) MAE: 3068.009812765289 Training_Accuracy_xg_reg=xg_reg.score(x_train,y_train.ravel())*100 print('Training Accuracy:',Training_Accuracy_xg_reg,'%') Training Accuracy 94.09053875232357 % </pre>	<p>94.12206327106377 %</p>	<p>Screenshot of the confusion matrix</p> <pre> correlation = np.corrcoef(y_test.ravel(), y_pred)[0, 1] print("Correlation between y_train and y_pred:", correlation) Correlation between y_train and y_pred: 0.970242865687869 </pre>

[illegible]