



Model Development Phase Template

Date	15 March 2024
Team ID	738303
Project Title	Machine Learning Approach For Employee Performance Prediction
Maximum Marks	4 Marks

Initial Model Training Code, Model Validation and Evaluation Report

Random Forest Regression model is the best fit for the employee performance prediction model.

Initial Model Training Code:





```
#splitting data into features and target column
x = data.drop(columns=['actual_productivity', 'wip'], axis=1)
y = data['actual_productivity']

print(x)

print(y)

#splitting data into train test split
#import train_test_split dependency
from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=42)

x.shape,x_train.shape,x_test.shape
```

Model Validation and Evaluation Report:

Note: Machine Learning Approach For Emplyee Performance Prediction is a regression model . It is not possible to create or fit a confusion matrix for regression models.

So we cannot create confusion matrix for this project.

Model	Regression Report	MSE	Confusion Matrix





Linear regression model	<pre>#model building #importing linear regression dependency from sklearn.linear.model import LinearRegression linear=linearRegression() linear.fit(x_train,y_train) #linear model mean squared error score_train=linear.predict(x_train) mse_train=mean_squared_error(y_train,score_train) print("mean squared error in training data in linear regre score_test=linear.predict(x_test)</pre>	ession is:",mse_train)	0.021			
	<pre>mse_test=mean_squared_error(y_test,score_test) print("mean squared error in testing data in linear regres</pre>	ssion is:",mse_test)			Actual Value	predicted_value
	#linear model r2_score			921	0.268214	0.432858
	<pre>score_train=linear.predict(x_train) mse_train=r2_score(y_train,score_train)</pre>	(variable) mse_train: Float		321	0.800359	0.799398
	<pre>print("r2_score in training data in linear regression is:"</pre>	',mse_train)		101	0.681061	0.671121
	<pre>score_test=linear.predict(x_test) mse_test=r2_score(y_test,score_test)</pre>			920	0.325000	0.591028
	<pre>print("r2_score in test data in linear regression is:",mse</pre>	e_test)		58	0.667604	0.593638
	<pre>#linear model mean_absolute_error score_train=linear.predict(x_train)</pre>			790	0.800980	0.735931
	<pre>mse_train=mean_absolute_error(y_train,score_train) print("mean_absolute_error in training data in linear regr</pre>	ression is:",mse_train)		948	0.768847	0.549655
	<pre>score_test=linear.predict(x_test)</pre>			969	0.768847	0.526311
	<pre>mse_test=mean_absolute_error(y_test,score_test) print("mean_absolute_error in testing data in linear regre</pre>	ession is:",mse_test)		410	0.650417	0.631047
				1079	0.750396	0.750391
Random forest model	#Random Forest Regressor from sklearn.ensemble import RandomForestRegressor RandomForest = RandomForestRegressor() RandomForest.fit(x_train, y_train)		0.0120			
	#Random Forest Regressor mean squared error score_train=RandomForest.predict(x_train) mse_train=mean_squared_error(y_train,score_train)				Actual Value	predicted_value
	<pre>print("mean squared error in training data in Random Fore score_test=RandomForest.predict(x_test)</pre>	est Regressor is:",mse_train)		921	0.268214	0.432858
	mse_test=mean_squared_error(y_test,score_test) print("mean squared error in testing data in Random Fores	it Regressor is:",mse_test)		321	0.800359	0.799398
	#Random Forest Regressor r2_score			101	0.681061	0.671121
	<pre>score_train=RandomForest.predict(x_train) mse_train=r2_score(y_train,score_train) print("r2_score in training data in Random Forest Regress</pre>	or is:".mse train)		920	0.325000	0.591028
	score_test=RandomForest.predict(x_test)			58	0.667604	0.593638
	<pre>mse_test=r2_score(y_test,score_test) print("r2_score in test data in Random Forest Regressor i</pre>	is:",mse_test)		790	0.800980	0.735931
	<pre>#Random Forest Regressor mean_absolute_error score_train=linear.predict(x_train)</pre>			948	0.768847	0.549655
	mse_train=mean_absolute_error(y_train,score_train) print("mean_absolute_error in training data in Random For	rest Regressor is:",mse_train)		969	0.768847	0.526311
	score_test=linear.predict(x_test)			410	0.650417	0.631047
	<pre>mse_test=mean_absolute_error(y_test,score_test) print("mean_absolute_error in testing data in Random Fore</pre>	est Regressor is:".mse test)		1079	0.750396	0.750391





Xgboost model

```
#Xgboost regression
import xgboost as xgb
model_xgb=xgb.XGBRegressor(n_estimators=200,max_depth=5,learning_rate=0.1)
model_xgb.fit(x_train,y_train)

#Xgboost mean squared error
score_train=model_xgb.predict(x_train)
mse_train=mean_squared_error(y_train,score_train)
print("mean squared_error in training data in Xgboost regression is:",mse_train)

score_test=model_xgb.predict(x_test)
mse_test=model_xgb.predict(x_test)
mse_test=model_xgb.predict(x_test)
print("mean squared_error in testing data in Xgboost regressionr is:",mse_test)

#Xgboost Regressor r2_score
score_train=model_xgb.predict(x_train)
mse_train=r2_score(y_train,score_train)
print("r2_score in training data in Xgboost regression is:",mse_train)

score_test=model_xgb.predict(x_test)
mse_test=r2_score(y_test,score_test)
print("r2_score in test data in Random Xgboost regression is:",mse_test)

#Xgboost regression mean_absolute_error
score_train=linear.predict(x_train)
mse_train=mean_absolute_error(y_train,score_train)
print("mean_absolute_error in training data in Xgboost regression is:",mse_train)
score_test=linear.predict(x_test)
mse_test=mean_absolute_error in training data in Xgboost regression is:",mse_train)
print("mean_absolute_error in testing data in Xgboost regression is:",mse_test)
print("mean_absolute_error in testing data in Xgboost regression is:",mse_test)
```

0.0133

	Actual Value	predicted_value
921	0.268214	0.432858
321	0.800359	0.799398
101	0.681061	0.671121
920	0.325000	0.591028
58	0.667604	0.593638
790	0.800980	0.735931
948	0.768847	0.549655
969	0.768847	0.526311
410	0.650417	0.631047
1079	0.750396	0.750391