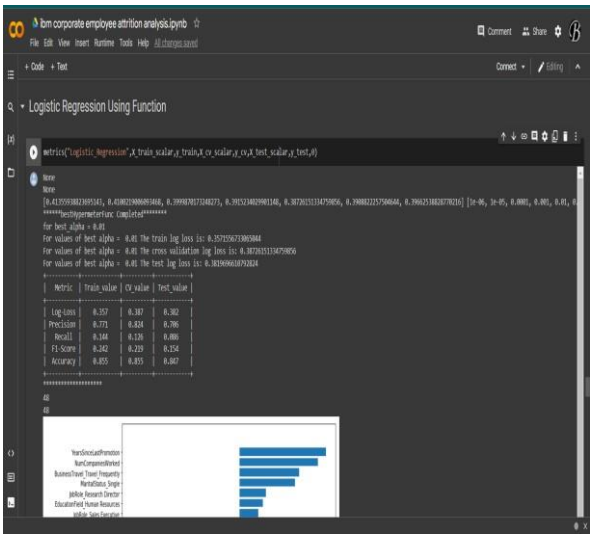
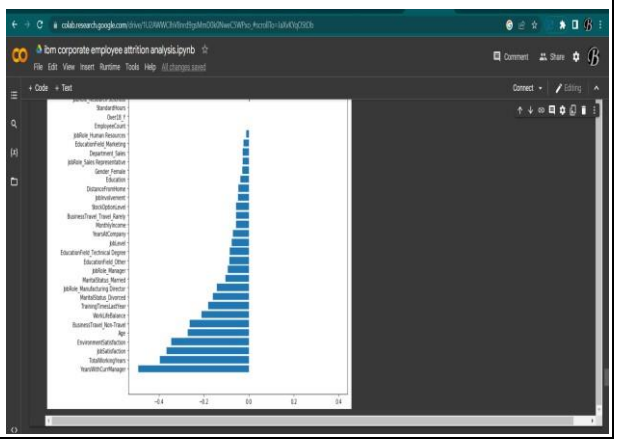
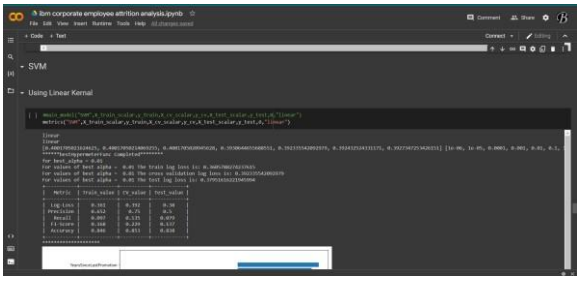
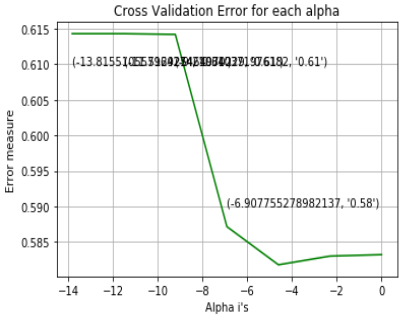
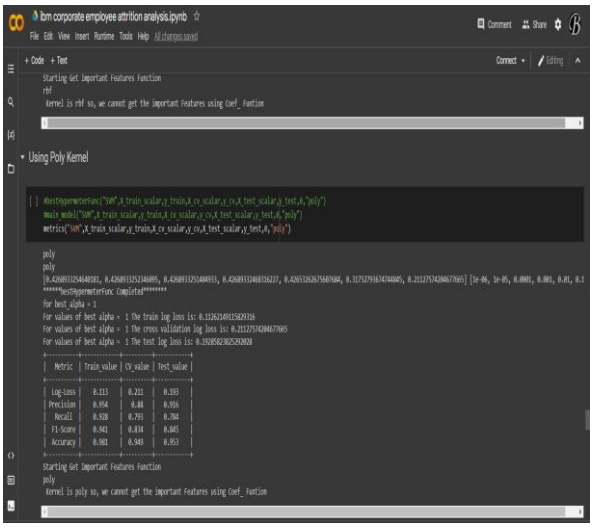
Project Development Phase
Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID06047
Project Name	Project - Corporate Employee Attrition Analytics
Maximum Marks	10 Marks

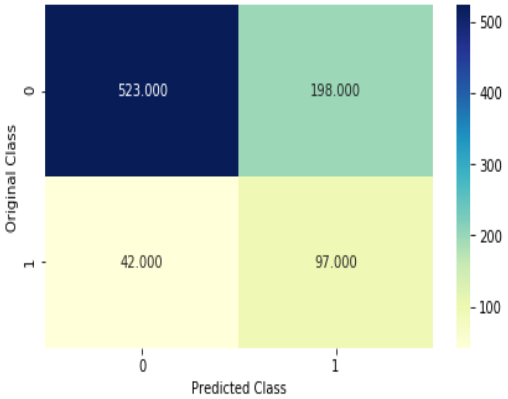
Model Performance Testing:

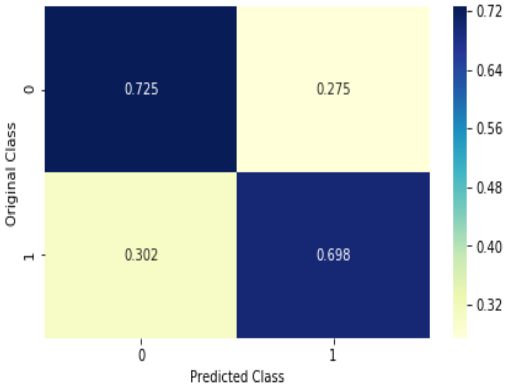
Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary Regression analysis	for best_alpha = 0.01 For values of best alpha = 0.01 The train log loss is: 0.3571556733065044 For values of best alpha = 0.01 The cross validation log loss is: 0.38726151334759856 For values of best alpha = 0.01 The test log loss is: 0.3819696610792824 +----------+----------+----------+----------+ | Metric | Train_value | CV_value | Test_value | +----------+----------+----------+----------+ | Log-Loss | 0.357 | 0.387 | 0.382 | | Precision | 0.771 | 0.824 | 0.706 | | Recall | 0.144 | 0.126 | 0.086 | | F1-Score | 0.242 | 0.219 | 0.154 | | Accuracy | 0.855 | 0.855 | 0.847 |	
	SVM Model	linear linear [0.4001705021624625, 0.40017050214069255, 0.4001705020945628, 0.3930644655688551, 0.392335542092979, 0.392432524331171, 0.3927347253426151] [1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1, 1] ******bestHypermeterFunc Completed******* for best_alpha = 0.01 For values of best alpha = 0.01 The train log loss is: 0.3605788274237615 For values of best alpha = 0.01 The cross validation log loss is: 0.392335542092979 For values of best alpha = 0.01 The test log loss is: 0.37951616221945994 +----------+----------+----------+----------+ | Metric | Train_value | CV_value | Test_value | +----------+----------+----------+----------+ | Log-Loss | 0.361 | 0.392 | 0.38 | | Precision | 0.652 | 0.75 | 0.5 | | Recall | 0.097 | 0.135 | 0.079 | | F1-Score | 0.168 | 0.229 | 0.137 | | Accuracy | 0.846 | 0.853 | 0.838 | +----------+----------+----------+----------+ **********	

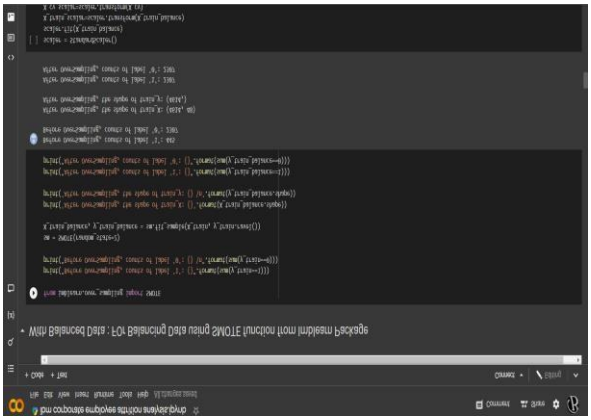
			
2.	Tune the Model Accuracy	Training Accuracy – 80% linear for alpha = 1e-06 Log Loss : 0.6143143363763562 for alpha = 1e-05 Log Loss : 0.6143143337374273 for alpha = 0.0001 Log Loss : 0.6142136356700172 for alpha = 0.001 Log Loss : 0.5871537845084024 for alpha = 0.01 Log Loss : 0.5817866567511131 for alpha = 0.1 Log Loss : 0.583009140383031 for alpha = 1  Log Loss : 0.5832164963645 linear [0.6143143363763562, 0.6143143337374273, 0.6142136356700172, 0.5871537845084024, 0.5817866567511131, 0.583009140383031, 0.5832164963645] [1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1, 1] ******bestHypermeterFunc Completed******* for best_alpha = 0.01 For values of best alpha = 0.01 The train log loss is: 0.5278361295624395 For values of best alpha = 0.01 The cross validation log loss is: 0.5817866567511131	 

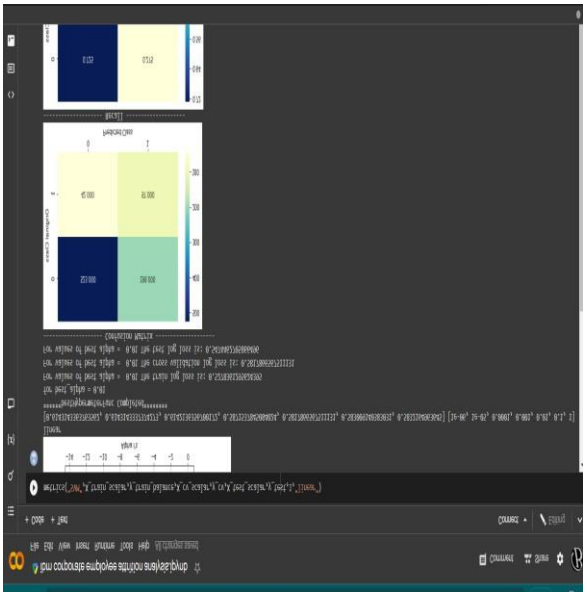
		For values of best alpha = 0.01 The test log loss is: 0.5474462705866496 ---------- Confusion Matrix ----------	

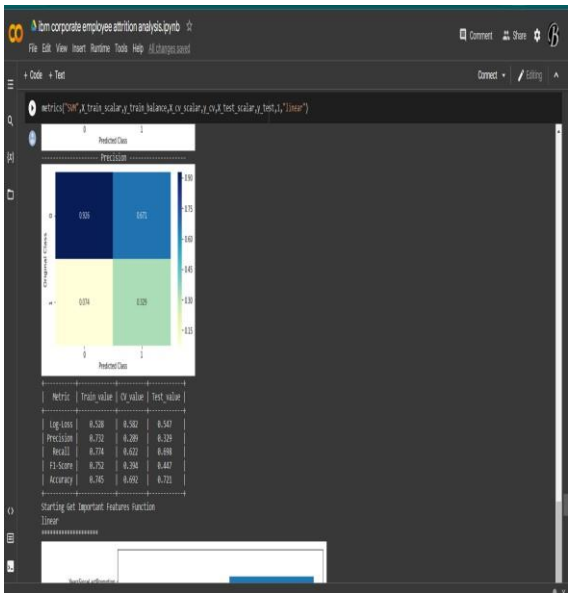

523.000
198.000
42.000
97.000
Original Class
Predicted Class
0
1
500
400
300
200
100

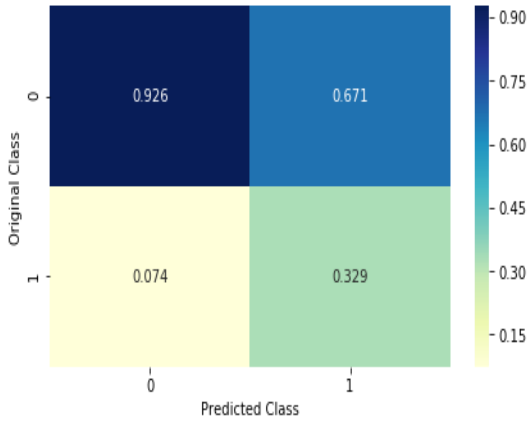

0.725
0.275
0.302
0.698
Original Class
Predicted Class
0
1
0.72
0.64
0.56
0.48
0.40
0.32

---------- Recall ----------





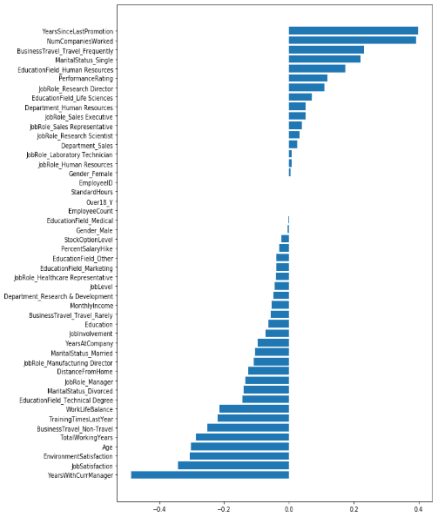

Precision
Predicted Class
Metric | Train value | CV value | Test value
Log-Loss | 0.528 | 0.582 | 0.547
Precision | 0.732 | 0.289 | 0.329
Recall | 0.774 | 0.622 | 0.698
F1-Score | 0.752 | 0.394 | 0.447
Accuracy | 0.745 | 0.692 | 0.721
Starting Get Important Features Function
Linear

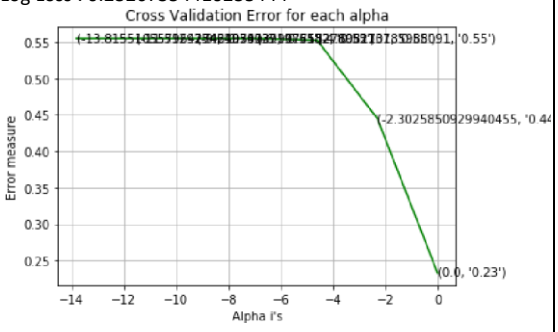
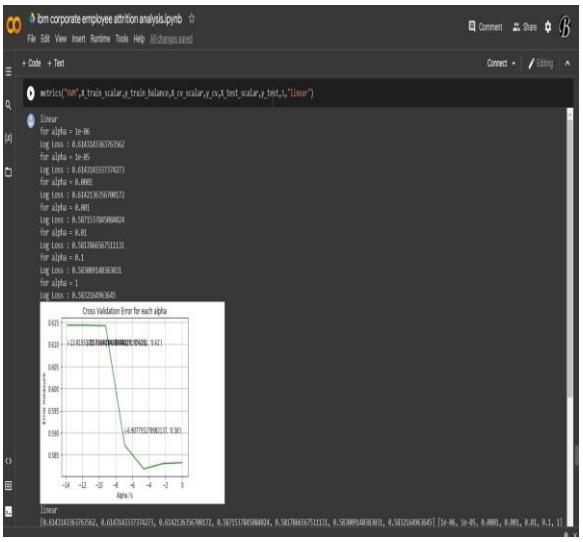
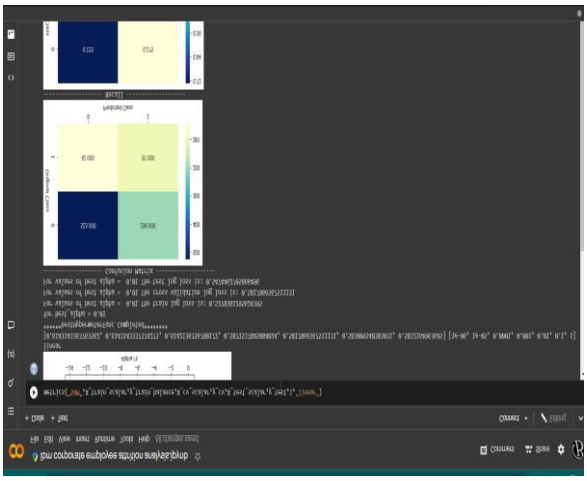


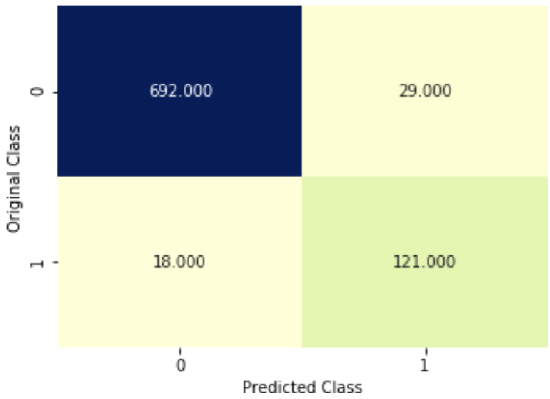
---------- Precision ----------

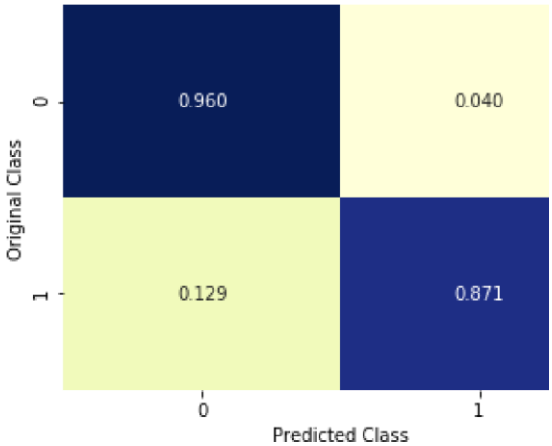
+----------+----------+----------+----------+
| Metric | Train_value | CV_value | Test_value |
+----------+----------+----------+----------+
Log-Loss	0.528	0.582	0.547
Precision	0.732	0.289	0.329
Recall	0.774	0.622	0.698
F1-Score	0.752	0.394	0.447
Accuracy	0.745	0.692	0.721
+----------+----------+----------+----------+

Starting Get Important Features Function
linear
****

		 Validation Accuracy – 76%	

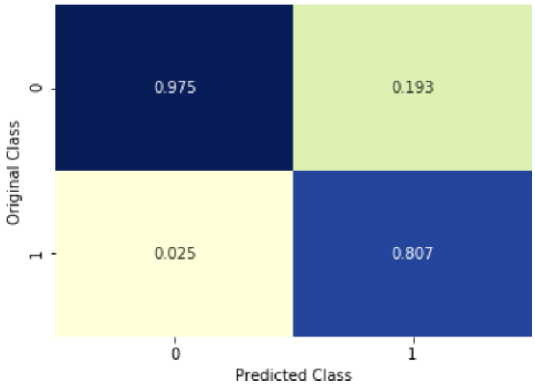
3.	Confidence Score (Only Yolo Projects)	Confidence Score – for alpha = 1e-06 Log Loss : 0.5544280465616621 for alpha = 1e-05 Log Loss : 0.5544279653425883 for alpha = 0.0001 Log Loss : 0.5544279626082605 for alpha = 0.001 Log Loss : 0.554427965596575 for alpha = 0.01 Log Loss : 0.5520345548670238 for alpha = 0.1 Log Loss : 0.4446536394141772 for alpha = 1 Log Loss : 0.23267354410235444  poly [0.5544280465616621, 0.5544279653425883, 0.5544279626082605, 0.554427965596575, 0.5520345548670238, 0.4446536394141772, 0.23267354410235444] [1e-06, 1e-05, 0.0001, 0.001, 0.01, 0.1, 1] ******bestHypermeterFunc Completed******** for best_alpha = 1 For values of best alpha = 1 The train log loss is: 0.05843126754768401 For values of best alpha = 1 The cross validation log loss is: 0.23267354410235444 For values of best alpha = 1 The test log loss is: 0.19790587725653494 ---------- Confusion Matrix ----------	 





Recall ----------

---------- Precision ----------



+----------+----------+----------+----------+

| Metric | Train_value | CV_value | Test_value |

+----------+----------+----------+----------+

| Log-Loss | 0.058 | 0.233 | 0.198 |

| Precision | 0.985 | 0.802 | 0.807 |

| Recall | 0.99 | 0.838 | 0.871 |

| F1-Score | 0.988 | 0.819 | 0.837 |

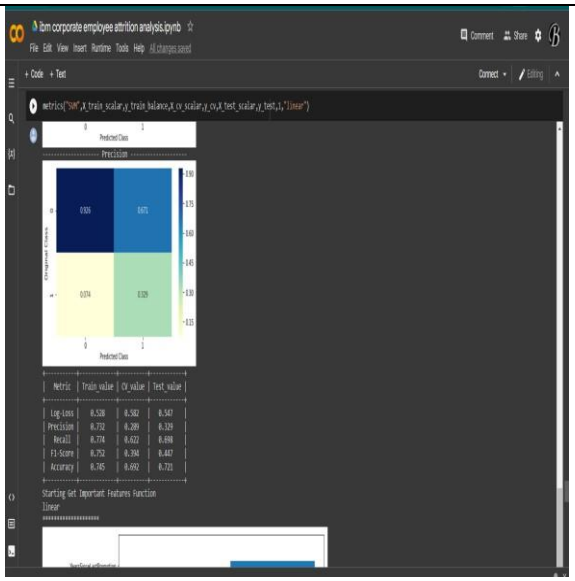
| Accuracy | 0.988 | 0.94 | 0.945 |

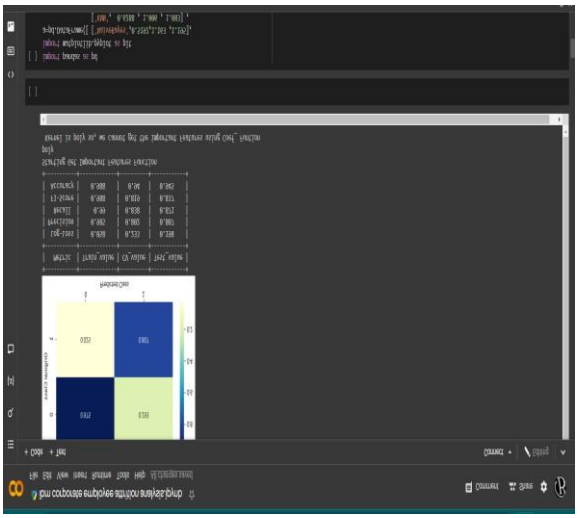
+----------+----------+----------+----------+

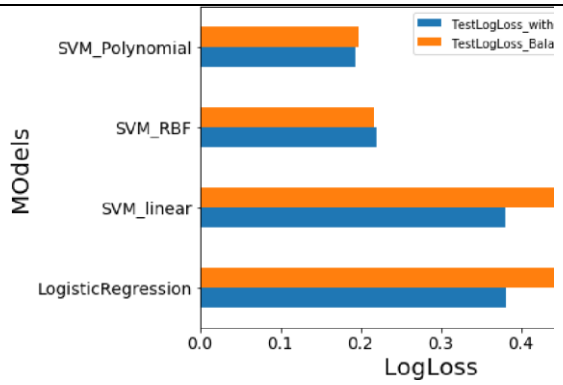
Starting Get Important Features Function

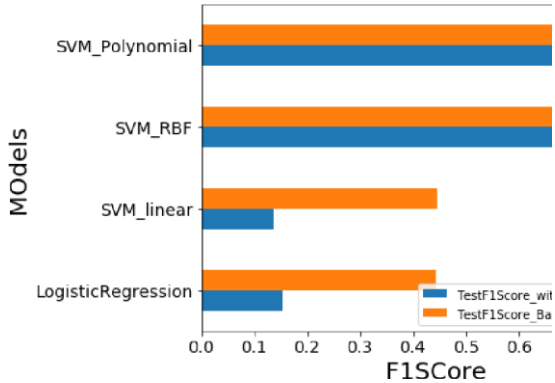
poly

Kernel is poly so, we cannot get the important Features using Coef_ Funtion









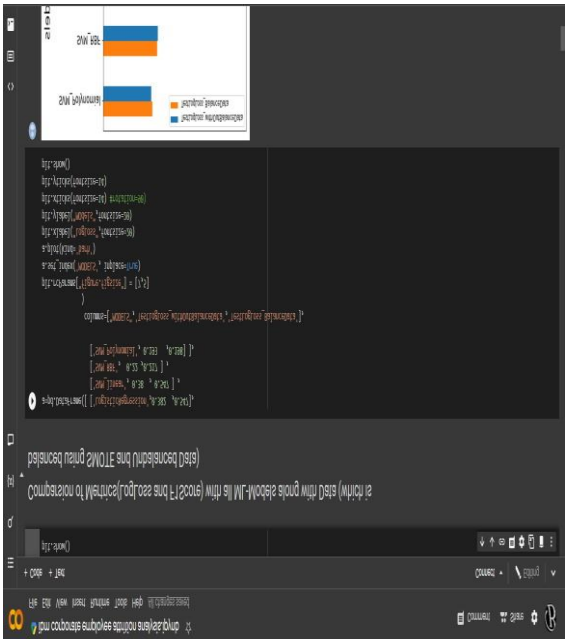
Regression Model:
R2 score -79%

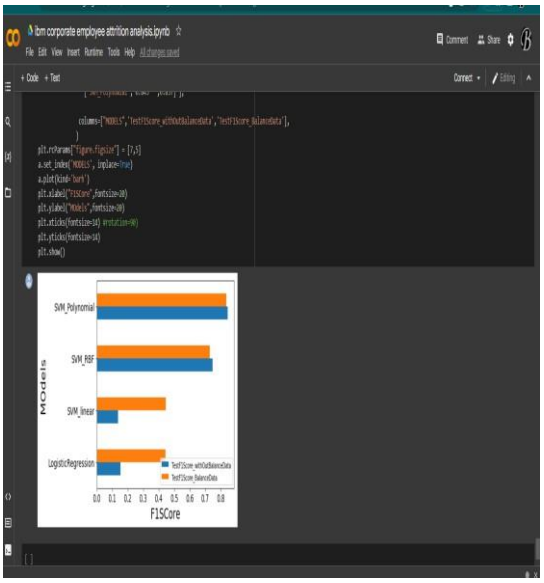
Classification Model:
Confusion Matrix - 79%
Accuracy Score- 76%

Hyperparameter Tuning – 80%

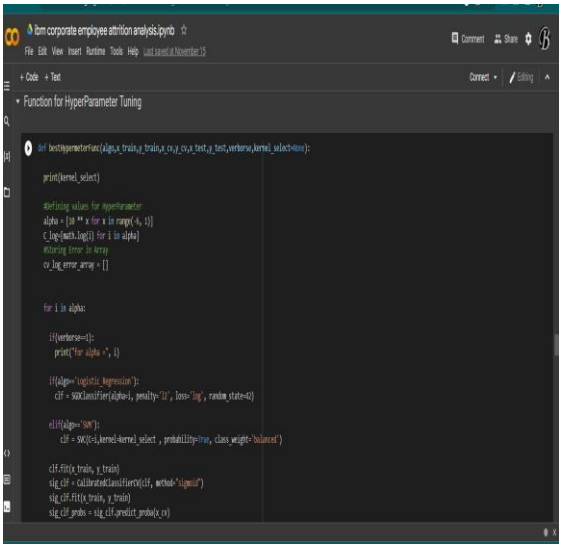
Validation Method –

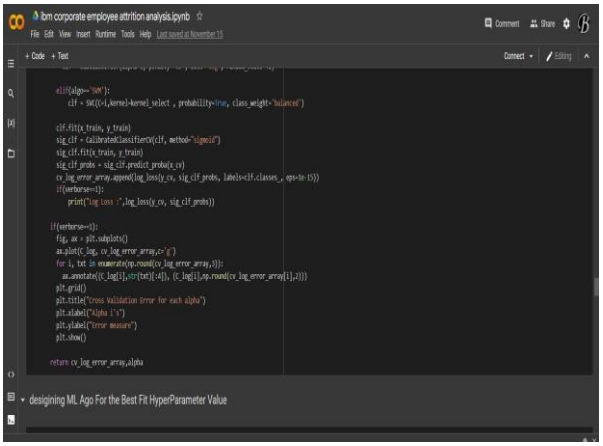
1.predictive analysis
2.svm model
3. poly kernel
4. RBF Kernel
5.regression analysis
6.ML Model
7. SMOTE Function

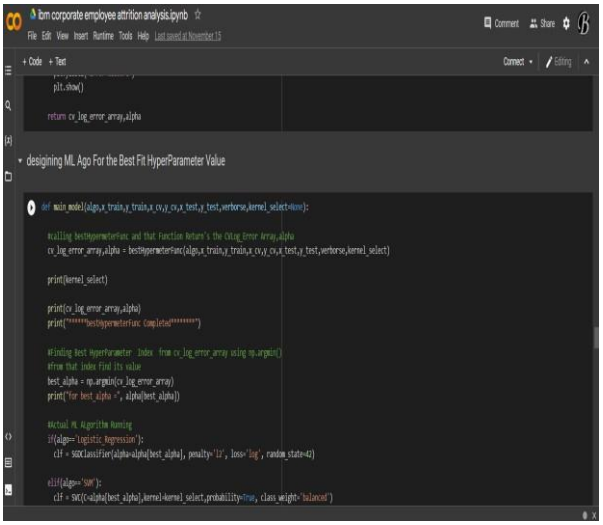


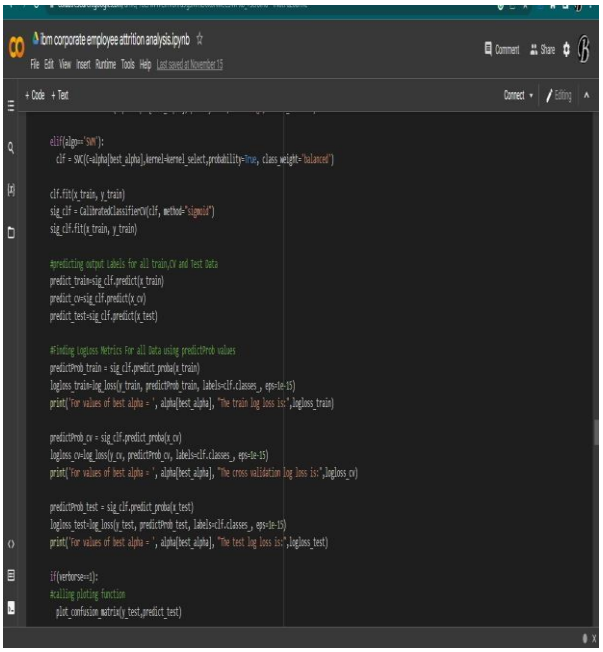


	Hyper Parameter Tuning 80%		


ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help Last saved at November 15
Comment Share
+ Code + Text
Connect Editing
Function for HyperParameter Tuning
def bestHyperparameterFunc(algo,x_train,y_train,x_cv,y_cv,x_test,y_test,verbose,kernel_select=None):
print(kernel_select)
#Defining values for HyperParameter
alpha = [10 ** x for x in range(-6, 1)]
C_log=[math.log(i) for i in alpha]
#Storing Error in Array
cv_log_error_array = []
for i in alpha:
if(verbose==1):
print("For alpha =", i)
if(algo=='Logistic_Regression'):
clf = SGDClassifier(alpha=i, penalty='l2', loss='log', random_state=42)
elif(algo=='SVM'):
clf = SVC(C=i,kernel=kernel_select , probability=True, class_weight='balanced')
clf.fit(x_train, y_train)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(x_train, y_train)
sig_clf_probs = sig_clf.predict_proba(x_cv)


ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help Last saved at November 15
Comment Share
+ Code + Text
Connect Editing
elif(algo=='SVM'):
clf = SVC(C=i,kernel=kernel_select , probability=True, class_weight='balanced')
clf.fit(x_train, y_train)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(x_train, y_train)
sig_clf_probs = sig_clf.predict_proba(x_cv)
cv_log_error_array.append(log_loss(y_cv, sig_clf_probs, labels=clf.classes_, eps=1e-15))
if(verbose==1):
print("Log Loss =",log_loss(y_cv, sig_clf_probs))
if(verbose==1):
fig, ax = plt.subplots()
ax.plot(C_log, cv_log_error_array,c='g')
for i, txt in enumerate(np.round(cv_log_error_array,3)):
ax.annotate((C_log[i],str(txt)[:4]), (C_log[i],np.round(cv_log_error_array[i],2)))
plt.grid()
plt.title("Cross Validation Error for each alpha")
plt.xlabel("Alpha i's")
plt.ylabel("Error measure")
plt.show()
return cv_log_error_array,alpha
designing ML Algo For the Best Fit HyperParameter Value


ibm corporate employee attrition analysis.ipynb
File Edit View Insert Runtime Tools Help Last saved at November 15
Comment Share
+ Code + Text
Connect Editing
plt.show()
return cv_log_error_array,alpha
designing ML Algo For the Best Fit HyperParameter Value
def main_model(algo,x_train,y_train,x_cv,y_cv,x_test,y_test,verbose,kernel_select=None):
#calling bestHyperparameterFunc and that function Return's the CVLog Error Array,alpha
cv_log_error_array,alpha = bestHyperparameterFunc(algo,x_train,y_train,x_cv,y_cv,x_test,y_test,verbose,kernel_select)
print(kernel_select)
print(cv_log_error_array,alpha)
print("******bestHyperparameterFunc Completed********")
#Finding Best HyperParameter Index from cv_log_error_array using np.argmin()
#from that index find its value
best_alpha = np.argmin(cv_log_error_array)
print("For best_alpha =", alpha[best_alpha])
#Actual ML Algorithm Running
if(algo=='Logistic_Regression'):
clf = SGDClassifier(alpha=alpha[best_alpha], penalty='l2', loss='log', random_state=42)
elif(algo=='SVM'):
clf = SVC(C=alpha[best_alpha],kernel=kernel_select,probability=True, class_weight='balanced')



ibm corporate employee attrition analysis.ipynb

File Edit View Insert Runtime Tools Help Last saved at November 15

Comment Share

+ Code + Text

Connect Editing

```
elif(algo='SVM'):
  clf = SVC(C=alpha[best_alpha],kernel=kernel_select,probability=True, class_weight='balanced')

clf.fit(x_train, y_train)
sig_clf = CalibratedClassifierCV(clf, method="sigmoid")
sig_clf.fit(x_train, y_train)

#predicting output Labels for all train,CV and Test Data
predict_train=sig_clf.predict(x_train)
predict_cv=sig_clf.predict(x_cv)
predict_test=sig_clf.predict(x_test)

#finding Loglosss Metrics For all Data using predictProb values
predictProb_train = sig_clf.predict_proba(x_train)
logloss_train=log_loss(y_train, predictProb_train, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The train log loss is:",logloss_train)

predictProb_cv = sig_clf.predict_proba(x_cv)
logloss_cv=log_loss(y_cv, predictProb_cv, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The cross validation log loss is:",logloss_cv)

predictProb_test = sig_clf.predict_proba(x_test)
logloss_test=log_loss(y_test, predictProb_test, labels=clf.classes_, eps=1e-15)
print('For values of best alpha = ', alpha[best_alpha], "The test log loss is:",logloss_test)

if(verbose==1):
#calling plotting function
  plot_confusion_matrix(y_test,predict_test)
```