ML for Bank Claim Management Prediction

Mayara Cordeiro

Project overview

PD — DP — PML — AM — EM

Problem Definition

Data Preprocessing

Data cleaning
Nan values
Encode categorical values

Prepare Data for ML

Vector Assemble, split data
(test, train)

Apply models

Logistic Regression
Decision Tree Classifier,
Random Forest Classifier,
Gradient-boosted Tree
Classifier

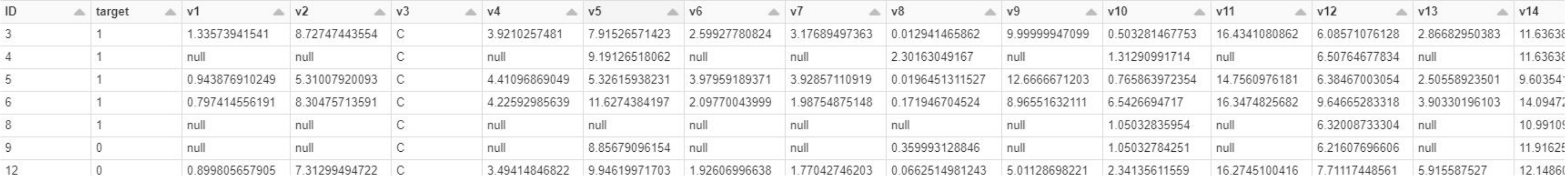
Evaluate Models

Binary Classification
Evaluator;
Multi Classification Evaluator

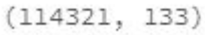
Run the project

“What is the best ML algorithm to improve a claim management process in a bank?”

The data



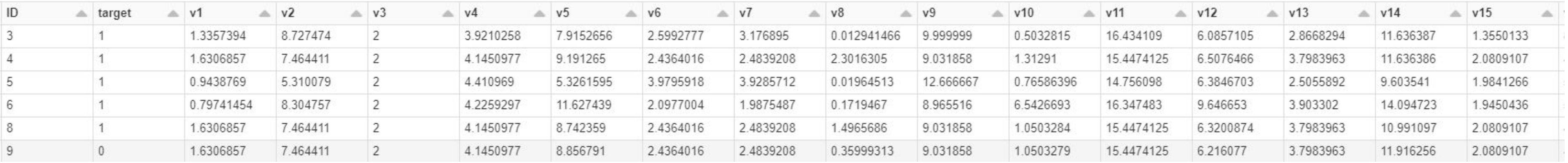
ID	target	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14
3	1	1.33573941541	8.72747443554	C	3.9210257481	7.91526571423	2.59927780824	3.17689497363	0.012941465862	9.99999947099	0.503281467753	16.4341080862	6.08571076128	2.86682950383	11.63638
4	1	null	null	C	null	9.19126518062	null	null	2.30163049167	null	1.31290991714	null	6.50764677834	null	11.63638
5	1	0.943876910249	5.31007920093	C	4.41096869049	5.32615938231	3.97959189371	3.92857110919	0.0196451311527	12.6666671203	0.765863972354	14.7560976181	6.38467003054	2.50558923501	9.60354
6	1	0.797414556191	8.30475713591	C	4.22592985639	11.6274384197	2.09770043999	1.98754875148	0.171946704524	8.96551632111	6.5426694717	16.3474825682	9.64665283318	3.90330196103	14.0947
8	1	null	null	C	null	null	null	null	null	null	1.05032835954	null	6.32008733304	null	10.9910
9	0	null	null	C	null	8.85679096154	null	null	0.359993128846	null	1.05032784251	null	6.21607696606	null	11.9162
12	0	0.899805657905	7.31299494722	C	3.49414846822	9.94619971703	1.92606996638	1.77042746203	0.0662514981243	5.01128698221	2.34135611559	16.2745100416	7.71117448561	5.915587527	12.1486



```
(114321, 133)
```

Data Preprocessing

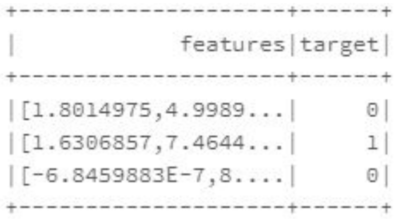
- ❏ CSV data
- ❏ Convert data types
- ❏ Replace nan values
- ❏ Encode categorical to numerical labels
- ❏ Result: New csv data



ID	target	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12	v13	v14	v15
3	1	1.3357394	8.727474	2	3.9210258	7.9152656	2.5992777	3.176895	0.012941466	9.999999	0.5032815	16.434109	6.0857105	2.8668294	11.636387	1.3550133
4	1	1.6306857	7.464411	2	4.1450977	9.191265	2.4364016	2.4839208	2.3016305	9.031858	1.31291	15.4474125	6.5076466	3.7983963	11.636386	2.0809107
5	1	0.9438769	5.310079	2	4.410969	5.3261595	3.9795918	3.9285712	0.01964513	12.666667	0.76586396	14.756098	6.3846703	2.5055892	9.603541	1.9841266
6	1	0.79741454	8.304757	2	4.2259297	11.627439	2.0977004	1.9875487	0.1719467	8.965516	6.5426693	16.347483	9.646653	3.903302	14.094723	1.9450436
8	1	1.6306857	7.464411	2	4.1450977	8.742359	2.4364016	2.4839208	1.4965686	9.031858	1.0503284	15.4474125	6.3200874	3.7983963	10.991097	2.0809107
9	0	1.6306857	7.464411	2	4.1450977	8.856791	2.4364016	2.4839208	0.35999313	9.031858	1.0503279	15.4474125	6.216077	3.7983963	11.916256	2.0809107

Prepare data for ML

- ❏ Vector Assembler
- ❏ Split data (train, test) - randomSplit([0.7, 0.3])



```
+--------------------+------+
|            features|target|
+--------------------+------+
|[1.8014975,4.9989...|     0|
|[1.6306857,7.4644...|     1|
|[-6.8459883E-7,8....|     0|
+--------------------+------+
```

Apply models

- ❑ Logistic Regression
- ❑ Decision Tree Classifier
- ❑ Random Forest Classifier
- ❑ Gradient Boost Classifier

Evaluation

ROC

Logistic Regression: 0.72

Decision Tree Classifier: 0.63

Random Forest Classifier: 0.71

Gradient Boosting Classifier: 0.74

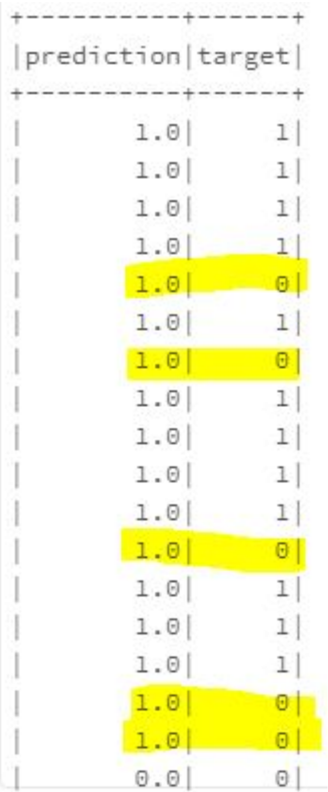
Accuracy

Logistic Regression: 0.767

Decision Tree Classifier: 0.776

Random Forest Classifier: 0.760

Gradient Boosting Classifier: 0.779



```
+----------+------+
|prediction|target|
+----------+------+
|       1.0|     1|
|       1.0|     1|
|       1.0|     1|
|       1.0|     1|
|       1.0|     0|
|       1.0|     1|
|       1.0|     0|
|       1.0|     1|
|       1.0|     1|
|       1.0|     1|
|       1.0|     1|
|       1.0|     0|
|       1.0|     1|
|       1.0|     1|
|       1.0|     1|
|       1.0|     0|
|       1.0|     0|
|       0.0|     0|
```

https://github.com/mayaraalvesc/CEBD1261_final_project

Conclusion

The ML algorithms allows the model to predict eligible process to be solved in priority. Improving the bank services and contribute to deliver a quality service for its clients, by decreasing the processing time.

Accuracy results are closer between the methods, gradient boosting tree presented better results in ROC and accuracy.