HA10

October 29, 2023

[ ]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import recall_score, precision_score, accuracy_score
import pandas as pd
import sqlite3
from pathlib import Path
```

[ ]:
```
dbPath = Path("../../data/data.sqlite")
if not dbPath.exists():
    # Avoid creating an empty database
    raise Exception("Database file does not exist")
con = sqlite3.connect(dbPath)
```

[ ]:
```
customerTypeMap = ["Student", "Teenager", "Adult", "Senior"]
customerLocationMap = ["Munich District One","Munich District Three","Munich␣
 ↪District Two","Munich District Four","Munich District Five"]
distributionChannelMap =  ["Feedera SE", "Deliver Now Holding", "Deliveruu Inc.
 ↪", "Orderly SE", "BestOrder Inc.", "TownExpress Inc.", "Heropizza Lmtd."]
weekdayMap = ["Sunday", "Tuesday", "Friday", "Saturday", "Wednesday", "Monday",␣
 ↪"Thursday"]
costfactorMap = ["","Chef 2","Chef 1","Ingredients","Delivery Scooters","Phone␣
 ↪Bill","Delivery Guy 2","Distribution channel fees","Waiter","Delivery Guy 1"]
sizeMap = ["Medium", "Small", "Large"]
typeMap = ["Funghi", "Salami", "Calzone", "Speciale", "Magherita", "Paprika",␣
 ↪"Veggie"]
```

[ ]:
```
df = pd.read_sql("SELECT * FROM Pizza_Case WHERE Variant != 5", con)
# df.drop(["_CASE_KEY"], axis=1, inplace=True)
df.drop("Customer_ID", axis=1, inplace=True)

# df["CustomerType"] = df["CustomerType"].map(customerTypeMap.index)
# df["CustomerLocation"] = df["CustomerLocation"].map(customerLocationMap.index)
# df["DistributionChannel"] = df["DistributionChannel"].
 ↪map(distributionChannelMap.index)
# df["Weekday"] = df["Weekday"].map(weekdayMap.index)
# df["CostFactor"] = df["CostFactor"].map(costfactorMap.index)
```

```
# df["PizzaSize"] = df["PizzaSize"].map(sizeMap.index)
# df["PizzaType"] = df["PizzaType"].map(typeMap.index)

df = pd.get_dummies(df, columns=["CustomerType", "CustomerLocation",␣
 ↪"DistributionChannel", "Weekday", "CostFactor", "PizzaSize", "PizzaType"])
df = pd.get_dummies(df, columns=["Variant"])

df["Profit"] = df["Revenue"] - df["Costs"]
df["IsOrderProfitable"] = df["Profit"] > 0

df
```

[ ]:

	_CASE_KEY	Revenue	Costs	CustomerSatisfaction	Daytime	\
0	895	17	13	3	14	
1	426	19	10	3	13	
2	690	9	16	0	12	
3	185	36	56	1	18	
4	1721	5	35	5	14	
…	…	…	…	…	…	
1821	1958	11	2	5	18	
1822	792	54	23	3	11	
1823	1500	29	2	4	18	
1824	605	12	12	3	21	
1825	178	8	44	1	21	

	CustomerType_Adult	CustomerType_Senior	CustomerType_Student	\
0	False	False	True	
1	False	False	True	
2	False	False	False	
3	True	False	False	
4	False	False	True	
…	…	…	…	
1821	False	False	True	
1822	False	False	True	
1823	False	False	False	
1824	False	False	True	
1825	False	False	True	

	CustomerType_Teenager	CustomerLocation_Munich District Five	…	\
0	False	False	…	
1	False	False	…	
2	True	False	…	
3	False	False	…	
4	False	False	…	
…	…	…	…	
1821	False	True	…	
1822	False	False	…	

```
1823                  True                                    False  …
1824                 False                                    False  …
1825                 False                                     True  …

      Variant_1  Variant_2  Variant_3  Variant_4  Variant_6  Variant_7  \
0          True      False      False      False      False      False
1          True      False      False      False      False      False
2          True      False      False      False      False      False
3          True      False      False      False      False      False
4          True      False      False      False      False      False
…           …          …          …          …          …          …
1821      False      False      False      False      False      False
1822      False      False      False      False      False      False
1823      False      False      False      False      False      False
1824      False      False      False      False      False      False
1825      False      False      False      False      False      False

      Variant_8  Variant_9  Profit  IsOrderProfitable
0         False      False       4               True
1         False      False       9               True
2         False      False      -7              False
3         False      False     -20              False
4         False      False     -30              False
…           …          …       …                  …
1821      False       True       9               True
1822      False       True      31               True
1823      False       True      27               True
1824      False       True       0              False
1825      False       True     -36              False

[1826 rows x 57 columns]
```

[ ]:
```python
# Extract the features and target variable
# X = df.drop(['CustomerSatisfaction'], axis=1)
# y = df['CustomerSatisfaction']

def algo(df, column: str):
    df = df.copy()
    X = df.drop([column], axis=1)
    y = df[column]


    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

    # Train a decision tree classifier on the training data
```

```
    clf = RandomForestClassifier(random_state=42) # Going with RandomForest␣
↪because it is 0.3324 accurate (DecisionTree is 0.24)
    clf.fit(X_train, y_train)

    # Evaluate the performance of the classifier on the testing data
    score = clf.score(X_test, y_test)
    print('Accuracy:', score)
    # recall

    y_pred = clf.predict(X_test)
    recall = recall_score(y_test, y_pred, average='macro')
    print('Recall:', recall)

    # precision
    precision = precision_score(y_test, y_pred, average='macro')
    print('Precision:', precision)



    print()
    # Print the feature importances
    importances = clf.feature_importances_
    for feature, importance in zip(X.columns, importances):
        print(feature, importance)
```

[ ]:
```
algo(df, "CustomerSatisfaction")
```

```
Accuracy: 0.3005464480874317
Recall: 0.28077330998970856
Precision: 0.2722970248705543

_CASE_KEY 0.08432939941360285
Revenue 0.06680530553936338
Costs 0.0699269689840014
Daytime 0.059033206857943214
CustomerType_Adult 0.011751377414448571
CustomerType_Senior 0.006374834464947382
CustomerType_Student 0.017370355236032198
CustomerType_Teenager 0.014280919448483033
CustomerLocation_Munich District Five 0.008503781927128418
CustomerLocation_Munich District Four 0.01118676119507871
CustomerLocation_Munich District One 0.019942859034068117
CustomerLocation_Munich District Three 0.015385742900750057
CustomerLocation_Munich District Two 0.013811181439244473
DistributionChannel_BestOrder Inc. 0.015869111353008433
DistributionChannel_Deliver Now Holding 0.017438999895611415
DistributionChannel_Deliveruu Inc. 0.009536309153017218
DistributionChannel_Feedera SE 0.018034431720154964
DistributionChannel_Heropizza Lmtd. 0.004614880127239662
```

```
DistributionChannel_Orderly SE 0.007458269900998698
DistributionChannel_TownExpress Inc. 0.007378176677216396
Weekday_Friday 0.01649637573282321
Weekday_Monday 0.0052289675391236955
Weekday_Saturday 0.016280387390048997
Weekday_Sunday 0.01733903863419077
Weekday_Thursday 0.005095950017712697
Weekday_Tuesday 0.004597368540836913
Weekday_Wednesday 0.008732589524701464
CostFactor_Chef 1 0.013867242485891062
CostFactor_Chef 2 0.015142105903718579
CostFactor_Delivery Guy 1 0.006307073563360592
CostFactor_Delivery Guy 2 0.008474325204799094
CostFactor_Delivery Scooters 0.01267673579478159
CostFactor_Distribution channel fees 0.0029112596973431314
CostFactor_Ingredients 0.014139001897748365
CostFactor_Phone Bill 0.010541099122699434
CostFactor_Waiter 0.011196236290614775
PizzaSize_Large 0.016933972125587792
PizzaSize_Medium 0.015884134753022673
PizzaSize_Small 0.01743216622630462
PizzaType_Calzone 0.012136298832956914
PizzaType_Funghi 0.015306941141092404
PizzaType_Magherita 0.014858791263325484
PizzaType_Paprika 0.005980891632132683
PizzaType_Salami 0.016808104886964024
PizzaType_Speciale 0.010385201552359153
PizzaType_Veggie 0.005960774684904636
Variant_1 0.013088340383255967
Variant_2 0.04758417881229145
Variant_3 0.023725639722316817
Variant_4 0.025120077640845718
Variant_6 0.006826467972036125
Variant_7 0.005664086211080998
Variant_8 0.004591128776274298
Variant_9 0.004163376778833921
Profit 0.07639521010400838
IsOrderProfitable 0.01309558647767299
```

[ ]:
```
algo(df, "Profit")
```

```
Accuracy: 0.11202185792349727
Recall: 0.06746672525219237
Precision: 0.06161092172648633

_CASE_KEY 0.07667673421596938
Revenue 0.0980763189888803
Costs 0.0906531295166075
```

CustomerSatisfaction 0.04904306825886533
Daytime 0.05516786951197574
CustomerType_Adult 0.012944370704456244
CustomerType_Senior 0.006928865814276364
CustomerType_Student 0.019396299118387278
CustomerType_Teenager 0.016154295482649784
CustomerLocation_Munich District Five 0.009596567662954315
CustomerLocation_Munich District Four 0.011241616633934106
CustomerLocation_Munich District One 0.02037368905591105
CustomerLocation_Munich District Three 0.014387026020183472
CustomerLocation_Munich District Two 0.014096648252217214
DistributionChannel_BestOrder Inc. 0.016869377061673747
DistributionChannel_Deliver Now Holding 0.01755390163486484
DistributionChannel_Deliveruu Inc. 0.010950153898415846
DistributionChannel_Feedera SE 0.01848192439315128
DistributionChannel_Heropizza Lmtd. 0.00567223599082628
DistributionChannel_Orderly SE 0.006929560619104183
DistributionChannel_TownExpress Inc. 0.0066633138321960045
Weekday_Friday 0.017874071480408378
Weekday_Monday 0.004791954229363332
Weekday_Saturday 0.01691205350394514
Weekday_Sunday 0.01873298268424433
Weekday_Thursday 0.006612844320026351
Weekday_Tuesday 0.005928228921467212
Weekday_Wednesday 0.009823332464708621
CostFactor_Chef 1 0.015384356144461864
CostFactor_Chef 2 0.01534635631273067
CostFactor_Delivery Guy 1 0.006372984210913514
CostFactor_Delivery Guy 2 0.00925856689198863
CostFactor_Delivery Scooters 0.012339553443674777
CostFactor_Distribution channel fees 0.0034582465902310016
CostFactor_Ingredients 0.014172517459475047
CostFactor_Phone Bill 0.010480577867903677
CostFactor_Waiter 0.011803105920097623
PizzaSize_Large 0.017373463287737933
PizzaSize_Medium 0.016578120462462088
PizzaSize_Small 0.019213993055673943
PizzaType_Calzone 0.011581161459809851
PizzaType_Funghi 0.01637655069882076
PizzaType_Magherita 0.014819568298546377
PizzaType_Paprika 0.006365374516114036
PizzaType_Salami 0.018654526297923277
PizzaType_Speciale 0.01064050298511402
PizzaType_Veggie 0.006866659344862168
Variant_1 0.0164792709133166
Variant_2 0.014653889351974866
Variant_3 0.013849384544456937
Variant_4 0.011843065916079818

```
Variant_6 0.008314021680770757
Variant_7 0.006711017357510553
Variant_8 0.004962948335996716
Variant_9 0.0037036502555398516
IsOrderProfitable 0.02386413212414914

/Users/passion/IIS/IIS Semester
5/FWP_ProcessInt/RootCauseAnalysis/.venv/lib/python3.11/site-
packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning: Recall
is ill-defined and being set to 0.0 in labels with no true samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
/Users/passion/IIS/IIS Semester
5/FWP_ProcessInt/RootCauseAnalysis/.venv/lib/python3.11/site-
packages/sklearn/metrics/_classification.py:1471: UndefinedMetricWarning:
Precision is ill-defined and being set to 0.0 in labels with no predicted
samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

[ ]:
```
# Drop Revenue and Costs because that would train the model to give us a non
↪useful result
dfPredictPrfitablity = df.copy()
dfPredictPrfitablity.drop("Profit", axis=1, inplace=True)
dfPredictPrfitablity.drop("Costs", axis=1, inplace=True)
dfPredictPrfitablity.drop("Revenue", axis=1, inplace=True)
algo(dfPredictPrfitablity, "IsOrderProfitable")
```

```
Accuracy: 0.6311475409836066
Recall: 0.49443413729128016
Precision: 0.4861111111111111

_CASE_KEY 0.1355577282356561
CustomerSatisfaction 0.06490618528825423
Daytime 0.08668795264961186
CustomerType_Adult 0.01438704742627468
CustomerType_Senior 0.00825754521837692
CustomerType_Student 0.02053394711487365
CustomerType_Teenager 0.017875116023892468
CustomerLocation_Munich District Five 0.011569976256827859
CustomerLocation_Munich District Four 0.015074795140506447
CustomerLocation_Munich District One 0.021129041135906874
CustomerLocation_Munich District Three 0.017468443640893384
CustomerLocation_Munich District Two 0.014526125286707168
DistributionChannel_BestOrder Inc. 0.019826051315630214
DistributionChannel_Deliver Now Holding 0.020717926380390 42
DistributionChannel_Deliveruu Inc. 0.01266200112284679
DistributionChannel_Feedera SE 0.020173001160956015
DistributionChannel_Heropizza Lmtd. 0.006970105178169404
```

```
DistributionChannel_Orderly SE 0.009446723808181774
DistributionChannel_TownExpress Inc. 0.007938415659395123
Weekday_Friday 0.018686477798084696
Weekday_Monday 0.007494759806653239
Weekday_Saturday 0.01862467419971576
Weekday_Sunday 0.020086748352224633
Weekday_Thursday 0.008699878462456996
Weekday_Tuesday 0.007867195930751431
Weekday_Wednesday 0.011294024550451795
CostFactor_Chef 1 0.019009482974569936
CostFactor_Chef 2 0.01644567856283895
CostFactor_Delivery Guy 1 0.010562878367192836
CostFactor_Delivery Guy 2 0.012363692731074225
CostFactor_Delivery Scooters 0.014682190363752016
CostFactor_Distribution channel fees 0.005679345880694441
CostFactor_Ingredients 0.016446181173428425
CostFactor_Phone Bill 0.013710982192719571
CostFactor_Waiter 0.013673371318568103
PizzaSize_Large 0.018976361783422366
PizzaSize_Medium 0.019835349184251998
PizzaSize_Small 0.022345190477971236
PizzaType_Calzone 0.013222049492127555
PizzaType_Funghi 0.019409489093381225
PizzaType_Magherita 0.016578899747356827
PizzaType_Paprika 0.009261715603271055
PizzaType_Salami 0.020469563462756624
PizzaType_Speciale 0.012972650999359672
PizzaType_Veggie 0.009285819215673265
Variant_1 0.019988528650726794
Variant_2 0.01568675470993108
Variant_3 0.01771548743735458
Variant_4 0.014057588514645775
Variant_6 0.009469876640160017
Variant_7 0.007317235864413748
Variant_8 0.006944171668200515
Variant_9 0.005427576746467377
```

[ ]:
```
dfPredictCustomerSatisfaction = df.copy()
dfPredictCustomerSatisfaction["IsCustomerSatisfied"] =␣
  ↪dfPredictCustomerSatisfaction["CustomerSatisfaction"] >= 3
algo(dfPredictCustomerSatisfaction, "CustomerSatisfaction")
```

```
Accuracy: 0.4972677595628415
Recall: 0.42287973140853263
Precision: 0.44346050415002286

_CASE_KEY 0.07697856298428007
Revenue 0.059991707839822196
```

Costs 0.05981558111689472
Daytime 0.0521370225357002
CustomerType_Adult 0.01041247827957797
CustomerType_Senior 0.005599535382377352
CustomerType_Student 0.01413190878039284
CustomerType_Teenager 0.012360345015669325
CustomerLocation_Munich District Five 0.007090703791759003
CustomerLocation_Munich District Four 0.009428871905294383
CustomerLocation_Munich District One 0.015413274199954181
CustomerLocation_Munich District Three 0.01211883250656866
CustomerLocation_Munich District Two 0.011811639112325792
DistributionChannel_BestOrder Inc. 0.013736451489523642
DistributionChannel_Deliver Now Holding 0.014088032123574859
DistributionChannel_Deliveruu Inc. 0.007956596769172508
DistributionChannel_Feedera SE 0.014327589162892264
DistributionChannel_Heropizza Lmtd. 0.004032360730129078
DistributionChannel_Orderly SE 0.005915159088743882
DistributionChannel_TownExpress Inc. 0.006361812830130432
Weekday_Friday 0.013335644351227402
Weekday_Monday 0.0036823315292144328
Weekday_Saturday 0.013107291411672528
Weekday_Sunday 0.014418836728531005
Weekday_Thursday 0.005004841544649436
Weekday_Tuesday 0.004140456395908661
Weekday_Wednesday 0.008447433886670573
CostFactor_Chef 1 0.012406248025027801
CostFactor_Chef 2 0.012509593082734376
CostFactor_Delivery Guy 1 0.005430035405000689
CostFactor_Delivery Guy 2 0.007613508335943312
CostFactor_Delivery Scooters 0.009868436199192008
CostFactor_Distribution channel fees 0.002799831527041059
CostFactor_Ingredients 0.010186104459078325
CostFactor_Phone Bill 0.008722771620197432
CostFactor_Waiter 0.009787924532768163
PizzaSize_Large 0.013878037545435217
PizzaSize_Medium 0.012217731318921635
PizzaSize_Small 0.01393960845801437
PizzaType_Calzone 0.010203609415715902
PizzaType_Funghi 0.013118086937805879
PizzaType_Magherita 0.012584881554355198
PizzaType_Paprika 0.005060877328832198
PizzaType_Salami 0.01363993849297138
PizzaType_Speciale 0.00870511349960095
PizzaType_Veggie 0.0054039309363212796
Variant_1 0.011365062348141892
Variant_2 0.033932571069266754
Variant_3 0.01826343797864046
Variant_4 0.01678004099132692

```
Variant_6 0.005810452323770341
Variant_7 0.0043279054609835135
Variant_8 0.004148164908398067
Variant_9 0.002495438976507484
Profit 0.06698739287594334
IsOrderProfitable 0.011073879830243905
IsCustomerSatisfied 0.16089408306916272
```

[ ]:
```
import pm4py
from pm4py.objects.log.importer.xes import importer as xes_importer
from pm4py.algo.discovery.alpha import algorithm as alpha_miner
```

[ ]:
```
log = xes_importer.apply("../../data/Pizza_Event.xes")
```

```
parsing log, completed traces :: 100%|      | 1826/1826 [00:00<00:00,
3732.59it/s]
```

[ ]:
```
log
```

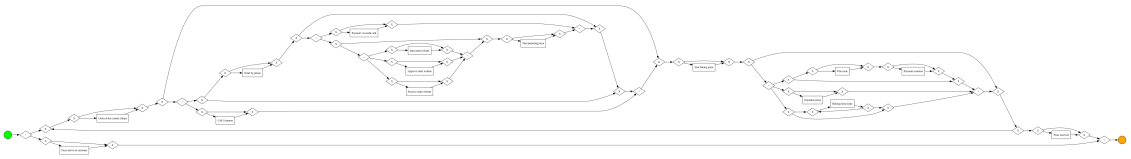
[ ]:
```
[{'attributes': {'concept:name': '1'}, 'events': [{'_CASE_KEY': 1, 'SORTING': 1,
'EVENTTIME': datetime.datetime(2018, 6, 15, 9, 37,
tzinfo=datetime.timezone.utc), 'ACTIVITY_EN': 'Order by phone', 'Automation':
'A', 'concept:name': 'Order by phone', 'time:timestamp': datetime.datetime(2018,
6, 15, 9, 37, tzinfo=datetime.timezone.utc), '@@index': 0, '@@case_index': 0},
'..', {'_CASE_KEY': 1, 'SORTING': 10, 'EVENTTIME': datetime.datetime(2018, 6,
15, 10, 6, tzinfo=datetime.timezone.utc), 'ACTIVITY_EN': 'Payment customer',
'Automation': 'A', 'concept:name': 'Payment customer', 'time:timestamp':
datetime.datetime(2018, 6, 15, 10, 6, tzinfo=datetime.timezone.utc), '@@index':
9, '@@case_index': 0}]}, '…', {'attributes': {'concept:name': '999'},
'events': [{'_CASE_KEY': 999, 'SORTING': 7983, 'EVENTTIME':
datetime.datetime(2018, 2, 16, 23, 42, tzinfo=datetime.timezone.utc),
'ACTIVITY_EN': 'Order at the counter (shop)', 'Automation': 'A', 'concept:name':
'Order at the counter (shop)', 'time:timestamp': datetime.datetime(2018, 2, 16,
23, 42, tzinfo=datetime.timezone.utc), '@@index': 15804, '@@case_index': 1825},
'..', {'_CASE_KEY': 999, 'SORTING': 7987, 'EVENTTIME': datetime.datetime(2018,
2, 16, 23, 48, tzinfo=datetime.timezone.utc), 'ACTIVITY_EN': 'Payment customer',
'Automation': 'A', 'concept:name': 'Payment customer', 'time:timestamp':
datetime.datetime(2018, 2, 16, 23, 48, tzinfo=datetime.timezone.utc), '@@index':
15808, '@@case_index': 1825}]}]
```

[ ]:
```
net, initial_marking, final_marking = alpha_miner.apply(log)
```

[ ]:
```
process_model = pm4py.discover_bpmn_inductive(log)
pm4py.view_bpmn(process_model)
```

```
/Users/passion/IIS/IIS Semester
5/FWP_ProcessInt/RootCauseAnalysis/.venv/lib/python3.11/site-
packages/pm4py/utils.py:508: UserWarning: the EventLog class has been deprecated
and will be removed in a future release.
```

```
  warnings.warn("the EventLog class has been deprecated and will be removed in a
future release.")
```



[ ]:
```
pm4py.stats.get_case_duration(log, case_id="895")
```

[ ]:
```
3780.0
```

[ ]:
```
df
```

[ ]:

	_CASE_KEY	Revenue	Costs	CustomerSatisfaction	Daytime	\
0	895	17	13	3	14	
1	426	19	10	3	13	
2	690	9	16	0	12	
3	185	36	56	1	18	
4	1721	5	35	5	14	
…	…	…	…	…	…	
1821	1958	11	2	5	18	
1822	792	54	23	3	11	
1823	1500	29	2	4	18	
1824	605	12	12	3	21	
1825	178	8	44	1	21	

	CustomerType_Adult	CustomerType_Senior	CustomerType_Student	\
0	False	False	True	
1	False	False	True	
2	False	False	False	
3	True	False	False	
4	False	False	True	
…	…	…	…	
1821	False	False	True	
1822	False	False	True	
1823	False	False	False	
1824	False	False	True	
1825	False	False	True	

	CustomerType_Teenager	CustomerLocation_Munich District Five	…	\
0	False	False	…	
1	False	False	…	
2	True	False	…	
3	False	False	…	

```
4                    False                                    False  …
…                    …                                    …  …
1821                 False                                     True  …
1822                 False                                    False  …
1823                  True                                    False  …
1824                 False                                    False  …
1825                 False                                     True  …

      Variant_1  Variant_2  Variant_3  Variant_4  Variant_6  Variant_7  \
0          True      False      False      False      False      False
1          True      False      False      False      False      False
2          True      False      False      False      False      False
3          True      False      False      False      False      False
4          True      False      False      False      False      False
…          …         …          …          …          …          …
1821      False      False      False      False      False      False
1822      False      False      False      False      False      False
1823      False      False      False      False      False      False
1824      False      False      False      False      False      False
1825      False      False      False      False      False      False

      Variant_8  Variant_9  Profit  IsOrderProfitable
0         False      False       4               True
1         False      False       9               True
2         False      False      -7              False
3         False      False     -20              False
4         False      False     -30              False
…         …          …          …                …
1821      False       True       9               True
1822      False       True      31               True
1823      False       True      27               True
1824      False       True       0              False
1825      False       True     -36              False

[1826 rows x 57 columns]
```

[ ]:
```python
from typing import *
# Enhance dataframe with start and end activities
def getActivityByOrderingTimestamp(caseId, order: str) -> Optional[str]:
    cursor = con.cursor()
    cursor.execute(f"select ACTIVITY_EN from Pizza_Event where _case_key = ?␣
 ↪order by eventtime {order} limit 1", (caseId, ))
    res = cursor.fetchone()
    cursor.close()
    if len(res) == 0:
        return None
    return res[0]
```

```
def getStartActivity(caseId) -> Optional[str]:
    return getActivityByOrderingTimestamp(caseId, "asc")

def getEndActivity(caseId) -> Optional[str]:
    return getActivityByOrderingTimestamp(caseId, "desc")
```

[ ]:
```
# Drop all cases of variant 5 because they are not in the log
#df = df[df["Variant"] != 5]


df["Duration"] = df["_CASE_KEY"].map(lambda x: pm4py.stats.
 ↪get_case_duration(log, case_id=str(x)))
df["StartActivity"] = df["_CASE_KEY"].map(getStartActivity)
df["EndActivity"] = df["_CASE_KEY"].map(getEndActivity)
df.drop("_CASE_KEY", axis=1, inplace=True)
df
```

[ ]:
```
      Revenue  Costs  CustomerSatisfaction  Daytime  CustomerType_Adult  \
0          17     13                     3       14               False
1          19     10                     3       13               False
2           9     16                     0       12               False
3          36     56                     1       18                True
4           5     35                     5       14               False
…         …      …                     …        …                   …
1821       11      2                     5       18               False
1822       54     23                     3       11               False
1823       29      2                     4       18               False
1824       12     12                     3       21               False
1825        8     44                     1       21               False

      CustomerType_Senior  CustomerType_Student  CustomerType_Teenager  \
0                   False                  True                  False
1                   False                  True                  False
2                   False                 False                   True
3                   False                 False                  False
4                   False                  True                  False
…                   …                     …                      …
1821                False                  True                  False
1822                False                  True                  False
1823                False                 False                   True
1824                False                  True                  False
1825                False                  True                  False

      CustomerLocation_Munich District Five  \
0                                     False
1                                     False
```

```
2                                    False
3                                    False
4                                    False
…                                    …
1821                                  True
1822                                 False
1823                                 False
1824                                 False
1825                                  True

      CustomerLocation_Munich District Four  …  Variant_4  Variant_6  \
0                                    False  …      False      False
1                                    False  …      False      False
2                                    False  …      False      False
3                                    False  …      False      False
4                                    False  …      False      False
…                                    …  …        …         …
1821                                 False  …      False      False
1822                                 False  …      False      False
1823                                 False  …      False      False
1824                                 False  …      False      False
1825                                 False  …      False      False

      Variant_7  Variant_8  Variant_9  Profit  IsOrderProfitable  Duration  \
0         False      False      False       4               True    3780.0
1         False      False      False       9               True    2100.0
2         False      False      False      -7              False    2760.0
3         False      False      False     -20              False    3060.0
4         False      False      False     -30              False    2460.0
…         …        …        …     …                …        …
1821      False      False       True       9               True    1020.0
1822      False      False       True      31               True     360.0
1823      False      False       True      27               True     420.0
1824      False      False       True       0              False     720.0
1825      False      False       True     -36              False     660.0

              StartActivity         EndActivity
0            Order by phone    Payment customer
1            Order by phone    Payment customer
2            Order by phone    Payment customer
3            Order by phone    Payment customer
4            Order by phone    Payment customer
…                       …                   …
1821    Start order website  Start baking pizza
1822  Approve order website  Start baking pizza
1823    Start order website       Call Customer
1824    Start order website  Start baking pizza
```

```
1825    Start order website  Start baking pizza

[1826 rows x 59 columns]
```

[ ]:
```python
# convert startactity with one hot encoding
df = pd.get_dummies(df, columns=["StartActivity"])
df = pd.get_dummies(df, columns=["EndActivity"])
```

[ ]:
```python
# Drop Revenue and Costs because that would train the model to give us a non useful result
dfPredictPrfitablity = df.copy()

dfPredictPrfitablity.drop("Profit", axis=1, inplace=True)
dfPredictPrfitablity.drop("Costs", axis=1, inplace=True)
dfPredictPrfitablity.drop("Revenue", axis=1, inplace=True)
algo(dfPredictPrfitablity, "IsOrderProfitable")
```

```
Accuracy: 0.6338797814207651
Recall: 0.4964749536178108
Precision: 0.49097971514889943

CustomerSatisfaction 0.05925539945307147
Daytime 0.07863419982409692
CustomerType_Adult 0.015150226382579988
CustomerType_Senior 0.00828176741724313
CustomerType_Student 0.021706037472616773
CustomerType_Teenager 0.017471904445207204
CustomerLocation_Munich District Five 0.011062117146257542
CustomerLocation_Munich District Four 0.014799541884243204
CustomerLocation_Munich District One 0.020821625679735216
CustomerLocation_Munich District Three 0.01596645212941166
CustomerLocation_Munich District Two 0.014882238794404111
DistributionChannel_BestOrder Inc. 0.019807157900789806
DistributionChannel_Deliver Now Holding 0.020991958395982824
DistributionChannel_Deliveruu Inc. 0.012839798273365706
DistributionChannel_Feedera SE 0.01930050129893492
DistributionChannel_Heropizza Lmtd. 0.006531633555758311
DistributionChannel_Orderly SE 0.010182668076997827
DistributionChannel_TownExpress Inc. 0.008718885185375795
Weekday_Friday 0.02015541994209486
Weekday_Monday 0.006826887590084733
Weekday_Saturday 0.018945202764525264
Weekday_Sunday 0.020889778204205758
Weekday_Thursday 0.007859685075255335
Weekday_Tuesday 0.0069077546825344555
Weekday_Wednesday 0.011369135879729585
CostFactor_Chef 1 0.017992166265161258
CostFactor_Chef 2 0.0176058653447878
```

```
CostFactor_Delivery Guy 1 0.009415022932601788
CostFactor_Delivery Guy 2 0.011961637005576412
CostFactor_Delivery Scooters 0.01438423951725228
CostFactor_Distribution channel fees 0.00637195045417295
CostFactor_Ingredients 0.015806196932643575
CostFactor_Phone Bill 0.012959898430453023
CostFactor_Waiter 0.014061915293220883
PizzaSize_Large 0.01975415611732813
PizzaSize_Medium 0.019085117769082532
PizzaSize_Small 0.020828679632110228
PizzaType_Calzone 0.012424139062153809
PizzaType_Funghi 0.01816931413039929
PizzaType_Magherita 0.015805619306127238
PizzaType_Paprika 0.009798739413295685
PizzaType_Salami 0.019937196113181233
PizzaType_Speciale 0.012037538997880858
PizzaType_Veggie 0.010003023076332088
Variant_1 0.015047622037983217
Variant_2 0.007537559463790657
Variant_3 0.01409031163700238
Variant_4 0.012876644757285819
Variant_6 0.006601978231158519
Variant_7 0.005151912953266339
Variant_8 0.003310359756505477
Variant_9 0.0019314163495059745
Duration 0.11547471650420926
StartActivity_Approve order website 0.0012084753212376967
StartActivity_Call Customer 0.0005655659690234845
StartActivity_Order at the counter (shop) 0.007501501758300429
StartActivity_Order by phone 0.012732869617369804
StartActivity_Receive order website 0.000496037642887064
StartActivity_Start baking pizza 0.0026201064728649834
StartActivity_Start order website 0.008922119355773537
StartActivity_Start preparing pizza 0.009306251268341269
EndActivity_Baking pizza ready 0.006013201026969365
EndActivity_Call Customer 0.000658366660234438
EndActivity_Departure pizza 0.0003694145949649699
EndActivity_Payment customer 0.010112459439099783
EndActivity_Pizza arrives at customer 0.001471798180727972
EndActivity_Pizza received 0.003171277857669317
EndActivity_Plan route 0.0
EndActivity_Start baking pizza 0.002692792248054823
EndActivity_Start preparing pizza 0.0023748476475097657
```

[ ]:
```
dfPredictCustomerSatisfaction = df.copy()
dfPredictCustomerSatisfaction["IsCustomerSatisfied"] =␣
  ↪dfPredictCustomerSatisfaction["CustomerSatisfaction"] >= 3
```

```
algo(dfPredictCustomerSatisfaction, "CustomerSatisfaction")
```

```
Accuracy: 0.48360655737704916
Recall: 0.4130795238319888
Precision: 0.40795427262260864

Revenue 0.05781563053541482
Costs 0.05685174339657642
Daytime 0.049770745575038716
CustomerType_Adult 0.010412179943950437
CustomerType_Senior 0.005810768502011287
CustomerType_Student 0.014533791680878325
CustomerType_Teenager 0.012512188281337705
CustomerLocation_Munich District Five 0.007052110702121478
CustomerLocation_Munich District Four 0.008471429564295767
CustomerLocation_Munich District One 0.017043510378451105
CustomerLocation_Munich District Three 0.011738321480409835
CustomerLocation_Munich District Two 0.011568456040502818
DistributionChannel_BestOrder Inc. 0.013499522471311427
DistributionChannel_Deliver Now Holding 0.01454018486901454
DistributionChannel_Deliveruu Inc. 0.007458243177920027
DistributionChannel_Feedera SE 0.013699679406392962
DistributionChannel_Heropizza Lmtd. 0.0036023153498936384
DistributionChannel_Orderly SE 0.006084630600611256
DistributionChannel_TownExpress Inc. 0.006446528385561587
Weekday_Friday 0.01442739870302257
Weekday_Monday 0.0037733306794624126
Weekday_Saturday 0.012962787236640665
Weekday_Sunday 0.014713513531313126
Weekday_Thursday 0.0049409721252848034
Weekday_Tuesday 0.0039176653182049544
Weekday_Wednesday 0.00803201906945849
CostFactor_Chef 1 0.011443965289342783
CostFactor_Chef 2 0.014043782041015805
CostFactor_Delivery Guy 1 0.005451062946273367
CostFactor_Delivery Guy 2 0.006982570330492539
CostFactor_Delivery Scooters 0.00989178273992555
CostFactor_Distribution channel fees 0.0025671252044867556
CostFactor_Ingredients 0.011143717214048469
CostFactor_Phone Bill 0.008627788952596643
CostFactor_Waiter 0.009753317024216142
PizzaSize_Large 0.013689193905884219
PizzaSize_Medium 0.012353534877423024
PizzaSize_Small 0.014775462810573103
PizzaType_Calzone 0.010536882994249635
PizzaType_Funghi 0.012539172977026582
PizzaType_Magherita 0.012787945479755756
PizzaType_Paprika 0.004843633741648276
```

```
PizzaType_Salami 0.013495656999390078
PizzaType_Speciale 0.009270316028986635
PizzaType_Veggie 0.005505804495904895
Variant_1 0.012660211977207434
Variant_2 0.01693340717374023
Variant_3 0.012295380666351018
Variant_4 0.011622569105916525
Variant_6 0.003295515230328277
Variant_7 0.0031894529921380832
Variant_8 0.0017789676311583701
Variant_9 0.0012416591874354296
Profit 0.06311055889981891
IsOrderProfitable 0.01191695354233953
Duration 0.07760064179497833
StartActivity_Approve order website 0.0006687475785058313
StartActivity_Call Customer 0.00020840750679536454
StartActivity_Order at the counter (shop) 0.014393304387916764
StartActivity_Order by phone 0.012792070586563127
StartActivity_Receive order website 0.00031744015269753495
StartActivity_Start baking pizza 0.0006822177905599718
StartActivity_Start order website 0.006141232234558194
StartActivity_Start preparing pizza 0.0053581400204876575
EndActivity_Baking pizza ready 0.004012825231078674
EndActivity_Call Customer 0.000485830664981778
EndActivity_Departure pizza 1.9443991807580186e-05
EndActivity_Payment customer 0.005748714519861572
EndActivity_Pizza arrives at customer 0.0010334321589042235
EndActivity_Pizza received 0.0018450171372076827
EndActivity_Plan route 0.0
EndActivity_Start baking pizza 0.00153828416302811 4
EndActivity_Start preparing pizza 0.00025212151385971374
IsCustomerSatisfied 0.14747506910145264
```

[ ]: