Contents

Abstract

The efficient management of backorders is crucial for streamlined planning and resource allocation in production, planning, and transportation. By leveraging the wealth of data generated by Enterprise Resource Planning (ERP) systems, including historical and structured data, a predictive model can be developed to forecast backorders and facilitate initiative-taking planning. This project aims to classify products as either going into backorder (Yes) or not (No) based on past data from inventories, supply chain, and sales. The approach involves various stages, including data exploration, data cleaning, feature engineering, model building, and model testing using classical machine learning techniques. Different machine learning algorithms will be evaluated to determine the most suitable approach for this specific case. The goal is to develop a solution that can accurately predict backorder sales for individual products using the provided dataset. The successful implementation of such a model would enable businesses to anticipate and prepare for backorders, optimizing their operations and minimizing disruptions caused by unexpected stockouts.

Technical specifications

Dataset

```
[9]: df.head(5)

[9]:        sku  national_inv  lead_time  in_transit_qty  forecast_3_month  \
     0  3285085          62.0        NaN             0.0               0.0
     1  3285131           9.0        NaN             0.0               0.0
     2  3285358          17.0        8.0             0.0               0.0
     3  3285517           9.0        2.0             0.0               0.0
     4  3285608           2.0        8.0             0.0               0.0

        forecast_6_month  forecast_9_month  sales_1_month  sales_3_month  \
     0               0.0               0.0            0.0            0.0
     1               0.0               0.0            0.0            0.0
     2               0.0               0.0            0.0            0.0
     3               0.0               0.0            0.0            0.0
     4               0.0               0.0            0.0            0.0

        sales_6_month  _  pieces_past_due  perf_6_month_avg perf_12_month_avg  \
     0            0.0  _              0.0            -99.00            -99.00
     1            0.0  _              0.0            -99.00            -99.00
     2            0.0  _              0.0              0.92              0.95
     3            0.0  _              0.0              0.78              0.75
     4            0.0  _              0.0              0.54              0.71

        local_bo_qty  deck_risk  oe_constraint  ppap_risk stop_auto_buy rev_stop  \
     0           0.0        Yes             No         No           Yes       No
     1           0.0         No             No        Yes            No       No
     2           0.0         No             No         No           Yes       No
     3           0.0         No             No        Yes           Yes       No
     4           0.0         No             No         No           Yes       No

                                    4

       went_on_backorder
     0                No
     1                No
     2                No
     3                No
     4                No

     [5 rows x 23 columns]
```

Source: https://github.com/rodrigosantis1/backorder_prediction/blob/master/dataset.rar

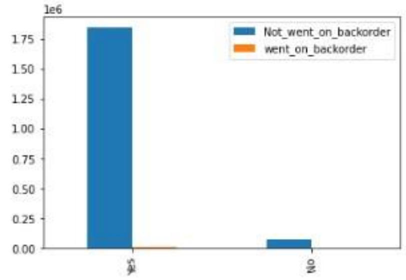
Dataset Overview

```
[10]: df.columns

[10]: Index(['sku', 'national_inv', 'lead_time', 'in_transit_qty',
             'forecast_3_month', 'forecast_6_month', 'forecast_9_month',
             'sales_1_month', 'sales_3_month', 'sales_6_month', 'sales_9_month',
             'min_bank', 'potential_issue', 'pieces_past_due', 'perf_6_month_avg',
             'perf_12_month_avg', 'local_bo_qty', 'deck_risk', 'oe_constraint',
             'ppap_risk', 'stop_auto_buy', 'rev_stop', 'went_on_backorder'],
            dtype='object')
```

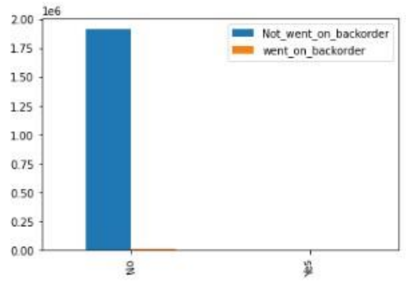
```
[15]: univariate_plot (df, 'stop_auto_buy', 'went_on_backorder')
```

```
     Not_went_on_backorder  went_on_backorder
Yes                1845988              13403
No                   69966                578
stop_auto_buy  state is No Product Not went on backorder 69966 times
3.6253034428620654 %
stop_auto_buy  state is No Product went on backorder 578 times
0.029949195180148554 %
stop_auto_buy  state is Yes Product Not went on backorder 1845988 times
95.65026801420773 %
stop_auto_buy  state is Yes Product went on backorder 13403 times
0.6944793477500537 %
```

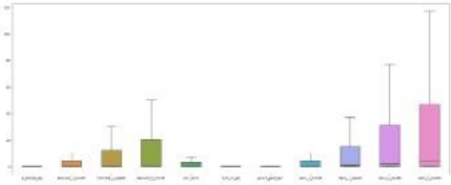

```
[16]: univariate_plot (df, 'oe_constraint', 'went_on_backorder')
```

```
     Not_went_on_backorder  went_on_backorder
No                 1915671              13972
Yes                    283                  9
oe_constraint  state is No Product Not went on backorder 1915671 times
99.26090775077917 %
oe_constraint  state is No Product went on backorder 13972 times
0.7239622059810304 %
oe_constraint  state is Yes Product Not went on backorder 283 times
0.014663706290626369 %
oe_constraint  state is Yes Product went on backorder 9 times
0.0004663369491718633 %
```

```
[22]: num_data = [ 'in_transit_qty', 'forecast_3_month',
                   'forecast_6_month', 'forecast_9_month', 'min_bank',
                   'local_bo_qty', 'pieces_past_due', 'sales_1_month',
                   'sales_3_month', 'sales_6_month', 'sales_9_month',]
      fig,ax = plt.subplots(figsize=(25,10),facecolor='white')
      sns.boxplot(data=df[num_data],ax=ax,width=0.5,fliersize=4,showfliers=False)   #,
       ↪will not Show the outliers beyond the caps.

[22]: <AxesSubplot:>
```



Test_Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	sku	national_i	lead_time	in_transit	forecast_3	forecast_6	forecast_9	sales_1_m	sales_3_m	sales_6_m	sales_9_m	min_bank	potential_	pieces_pa	perf_6_m	perf_12_m	local_bo_	deck_risk	oe_constr	ppap_risk	stop_auto	rev_stop	went_on_backorder	
2	3285085	62		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
3	3285131	9		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	No	No	Yes	No	No	No	
4	3285358	17	8	0	0	0	0	0	0	0	0	0	No	0	0.92	0.95	0	No	No	No	Yes	No	No	
5	3285517	9	2	0	0	0	0	0	0	0	2	0	No	0	0.78	0.75	0	No	No	Yes	Yes	No	No	
6	3285608	2	8	0	0	0	0	0	0	0	0	0	No	0	0.54	0.71	0	No	No	No	Yes	No	No	
7	3285960	15	2	0	0	0	0	0	0	1	2	0	No	0	0.37	0.68	0	No	No	No	Yes	No	No	
8	3286073	0		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
9	3286113	28		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
10	3286206	2		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	Yes	Yes	No	No	
11	3286325	2		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
12	3286917	20		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
13	3287458	0		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
14	3287918	13		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
15	3288094	208	16	0	0	0	0	0	0	0	0	0	No	0	0.66	0.64	0	No	No	No	Yes	No	No	
16	3288245	0		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
17	3288487	0	2	0	0	0	0	0	0	0	0	0	No	0	0.37	0.34	0	No	No	Yes	Yes	No	No	
18	3288567	0		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
19	3288664	26		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
20	3288770	36		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
21	3288805	23		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
22	3289046	2		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
23	3289088	265		0	0	0	0	2	14	29	46	2	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
24	3289182	5	8	0	0	0	0	0	0	0	0	0	No	0	0.95	0.97	0	No	No	No	Yes	No	No	
25	3289409	7		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	

Training_Dataset

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	sku	national_i	lead_time	in_transit	forecast_3	forecast_6	forecast_9	sales_1_n	sales_3_n	sales_6_n	sales_9_n	min_bank	potential_	pieces_pa	perf_6_m	perf_12_n	local_bo_	deck_risk	oe_constr	ppap_risk	stop_auto	rev_stop	went_on_backorder	
2	1026827	0		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
3	1043384	2	9	0	0	0	0	0	0	0	0	0	No	0	0.99	0.99	0	No	No	No	Yes	No	No	
4	1043696	2		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
5	1043852	7	8	0	0	0	0	0	0	0	0	1	No	0	0.1	0.13	0	No	No	No	Yes	No	No	
6	1044048	8		0	0	0	0	0	0	0	4	2	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
7	1044198	13	8	0	0	0	0	0	0	0	0	0	No	0	0.82	0.87	0	No	No	No	Yes	No	No	
8	1044643	1095		0	0	0	0	0	0	0	0	4	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
9	1045098	6	2	0	0	0	0	0	0	0	0	0	No	0	0	0	0	Yes	No	Yes	Yes	No	No	
10	1045815	140		0	15	114	152	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
11	1045867	4	8	0	0	0	0	0	0	0	0	0	No	0	0.82	0.87	0	No	No	No	Yes	No	No	
12	1045918	0	2	0	0	0	0	0	0	0	0	0	No	0	0.91	0.82	0	No	No	No	Yes	No	No	
13	1047146	20		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
14	1047199	18		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
15	1047661	29		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	No	No	No	
16	1049160	10		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
17	1049468	11	8	0	0	0	0	0	0	0	0	0	No	0	0.82	0.78	0	No	No	No	Yes	No	No	
18	1050390	12	2	0	0	0	0	0	0	0	0	0	No	0	1	0.98	0	No	No	No	Yes	No	No	
19	1050440	169	2	0	0	0	0	0	0	0	0	0	No	0	1	1	0	No	No	No	Yes	No	No	
20	1117808	4		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	No	No	No	Yes	No	No	
21	1050856	147	8	0	0	0	0	0	0	0	0	0	No	0	-99	1	0	No	No	Yes	Yes	No	No	
22	1118063	8		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
23	1127154	18		0	0	0	0	0	0	0	0	0	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
24	1127410	9		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	No	Yes	No	No	
25	1128393	33		0	0	0	0	0	0	0	0	1	No	0	-99	-99	0	Yes	No	Yes	Yes	No	No	

Input schema

```
[7]: df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1929935 entries, 0 to 1687859
Data columns (total 23 columns):
 #   Column             Dtype
---  ------             -----
 0   sku                object
 1   national_inv       float64
 2   lead_time          float64
 3   in_transit_qty     float64
 4   forecast_3_month   float64
 5   forecast_6_month   float64
 6   forecast_9_month   float64
 7   sales_1_month      float64
 8   sales_3_month      float64
 9   sales_6_month      float64
 10  sales_9_month      float64
 11  min_bank           float64
 12  potential_issue    object
 13  pieces_past_due    float64
 14  perf_6_month_avg   float64
 15  perf_12_month_avg  float64
 16  local_bo_qty       float64
 17  deck_risk          object
 18  oe_constraint      object
 19  ppap_risk          object
 20  stop_auto_buy      object
 21  rev_stop           object
 22  went_on_backorder  object
dtypes: float64(15), object(8)
memory usage: 353.4+ MB
```

Predicting Backorder (yes/no)
Historical data on product sales, including the quantity sold, date of sale, and any other relevant sales-related information.

Develop a machine learning model to predict product backorders by leveraging historical data from inventories, supply chain, and sales, enabling streamlined planning and preventing strain on production, logistics, and transportation.

Technology stack

Programming language	Python
Data Analysis and Machine Learning Libraries	Pandas, scikit-learn, TensorFlow
Data Visualization	Matplotlib, Seaborn
Integrated Development Environments (IDEs)	Jupyter Notebook

Model training/validation workflow

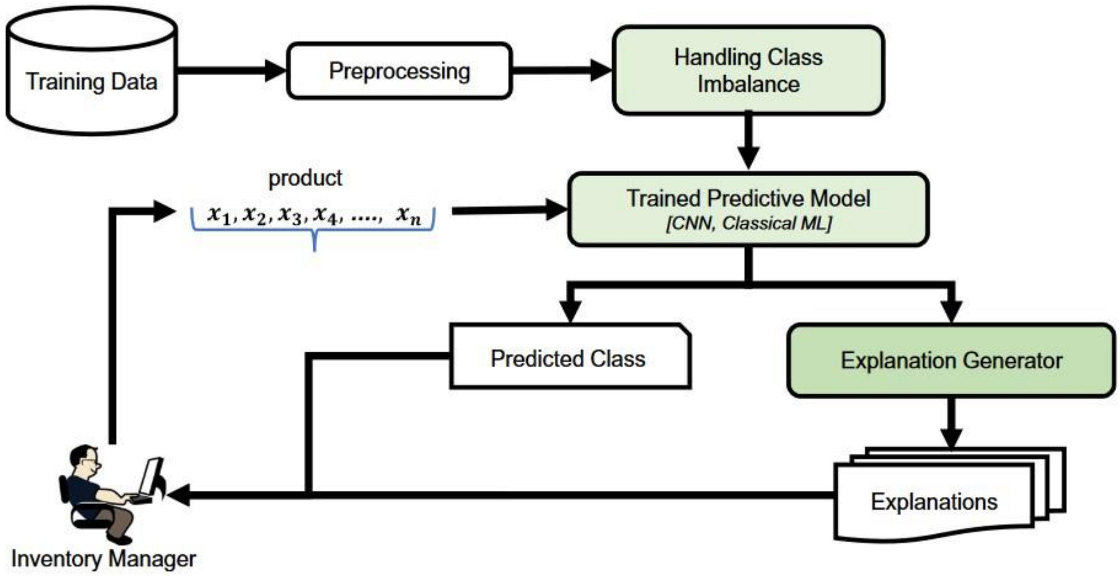
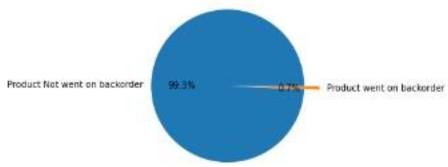


Figure 1Proposed explainable backorder prediction approach [1]

[1] M. Shajalal, A. Boden, and G. Stevens, "Explainable product backorder prediction exploiting CNN: Introducing explainable models in businesses," *Electron. Mark.*, vol. 32, no. 4, pp. 2107–2122, 2022, doi: 10.1007/s12525-022-00599-z.



Product went on backorder 13981 times 0.7244285429302023 %

Product Not went on backorder 1915954 times 99.2755714570698 %