INVENTORY MANAGEMENT

-

Mohamed Bayas
Kushi Kiranraj
Jyotsna Doonga
Deepak Vishwakarma
Nileshkumar Mandal
Arnav Sharma

MODEL SELECTION : INVENTORTY MANAGEMENT

"Inventory is the supply of raw materials, partially finished goods called work-inprogress and finished goods, an organization maintains to meet its operational needs. It represents a sizeable investment and a potential source of waste that needs to be carefully controlled. Inventory is defined as a stock of goods that is maintained by a business in anticipation of some future demand".

Inventory management is a major requirement even for the small and medium shop owners.
A system which tracks inventory levels, orders and sales in order to perform predictive analysis, obtaining forecasted demand will help to reduce overstock and out of-stock situations. A good inventory management system must ensure keeping enough stock in the warehouse to ensure the business keep functioning but not enough stock to drain its limited cash reserves.
This is a task where all business needs are to be anticipated, most of the calls made are crisis calls, yet where the people in-charge of the inventory need to find a solution. All done while simultaneously not using all the capital into non-moving stock."
Forecasting product demand is one of the core challenges in any retail business ".Another question that needs to be answered is how we plan on using artificial intelligence into inventory management. Does it have any use in managing inventory?
Artificial intelligence needs to be teamed with human oversight and should be treated as part of the system but not as a replacement for the system or the system itself.
That being said, there are several companies that are using artificial intelligence in the inventory process and it is yielding very impressive results which leads us to believe that artificial intelligence might after all have a significant impact on demand forecasting. We are transitioning from the traditional ways of managing inventory, which is the direct result of the availability of the huge amounts of real-time data that are now routinely generated on the internet and through the interconnected world of enterprise software systems and smart products. Amazon, one of the giants in the business have used inventory management teamed with artificial intelligence in almost every part of the prediction process. "Optimum inventory should be maintained by all organization so that under inventory can be eliminated which disrupt the financial figures. Careful evaluation of internal and external factors through better planning can improve the status of inventory ". Introduction:
Inventory plays a key role in every retail and e-commerce company.

PROBLEM STATEMENT

A major requirement for small/medium-sized businesses is Inventory Management since a lot of money and skilled labour has to be invested to do so. E-commerce giants use Machine Learning models to maintain their inventory based on demand for a particular item. Inventory Management can be extended as a service to small/medium sized businesses to improve their sales and predict the demand of various products. Demand forecasting is a crucial part of all businesses and brings up the following question: How much stock of an item should a company/business keep to meet the demands, i.e., what should the predicted demand of a product be? We discuss the challenges of building an Inventory system and discuss the design decisions.

PROTOTYPE DEVELOPMENT

- For the given dataset we can have product management which products need to be kept and which products need to be removed from the inventorydescribed in the dataset.
- There are three columns in the dataset which are among all crucial columns:

  SoldFlag, File_Type, and LowNetPrice.

- SoldFlag refers to the flag given to the status that if the product is sold ornot. The '1' refers to the status that the product is sold '0' refers to the status that the product is not sold.
- In File_type, the current active inventory that is in need of evaluation (i.e.,File Type = "Active"). The "Historical" data shows sales for the past 6 months.
- The LowNetPrice shows the net price of each product.

There are 198917 products available in the dataset as shown below:

- There are 75996 products whose status is “Historical” as shown below:

```
# 3. Count of the historical and active state
print("******************")
print(sales_data[sales_data['File_Type'] == 'Historical']['SKU_number'].count())
print(sales_data[sales_data['File_Type'] == 'Active']['SKU_number'].count())
```

```
******************
75996
122921
```

- Out of 75996, there are 12996 products whose File_type status is“Historical”

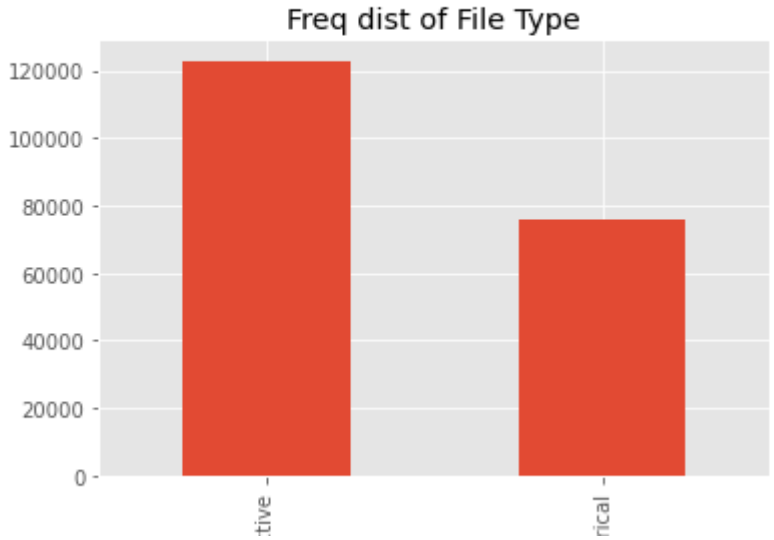
```
# 3.1 Split the dataset into two parts based on the file_type
sales_data_hist = sales_data[sales_data['File_Type'] == 'Historical']
sales_data_act = sales_data[sales_data['File_Type'] == 'Active']
```

```
******************
75996
122921
```

- We will be using the historical dataset for the analysis and training themodel Univariate distribution plots This section shows a frequency histogram for the selected variable along with the density and normalcurves for the data.

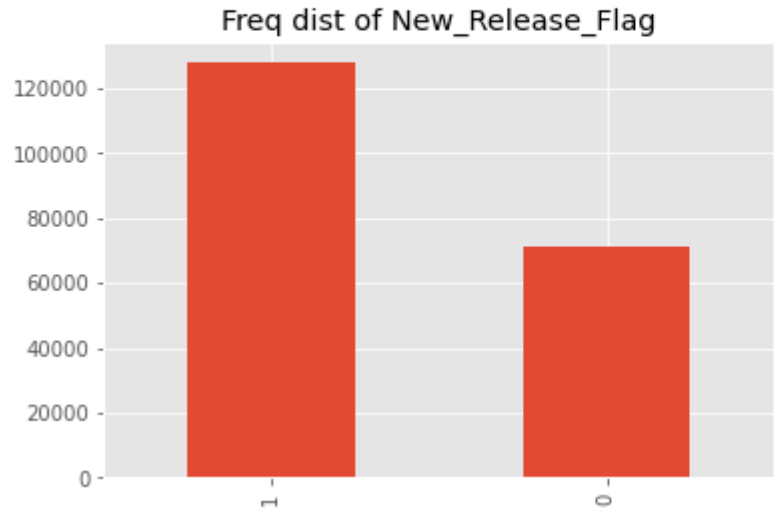
```
sales_data['File_Type'].value_counts().plot.bar(title="Freq dist of File Type")
```

```
<AxesSubplot:title={'center':'Freq dist of File Type'}>
```



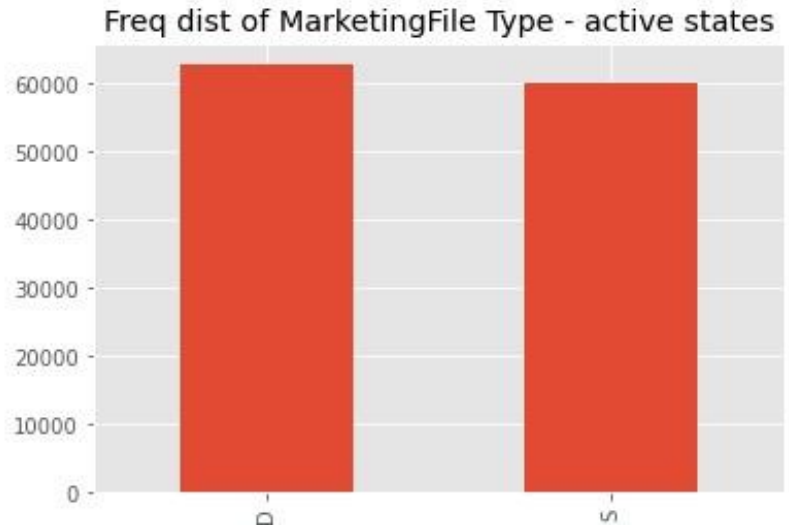

```
[ ] sales_data['New_Release_Flag'].value_counts().plot.bar(title="Freq dist of New_Release_Flag")
```

```
<AxesSubplot:title={'center':'Freq dist of New_Release_Flag'}>
```



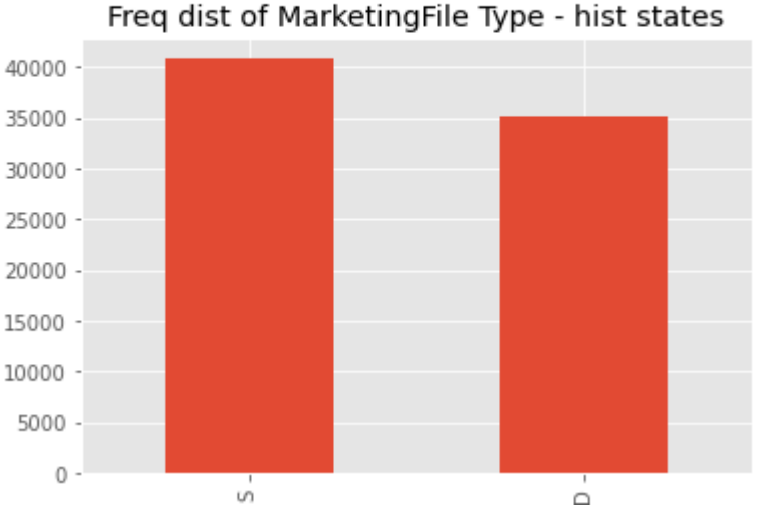
```
sales_data_act['MarketingType'].value_counts().plot.bar(title="Freq dist of MarketingFile Type - active states")
```

```
<AxesSubplot:title={'center':'Freq dist of MarketingFile Type - active states'}>
```



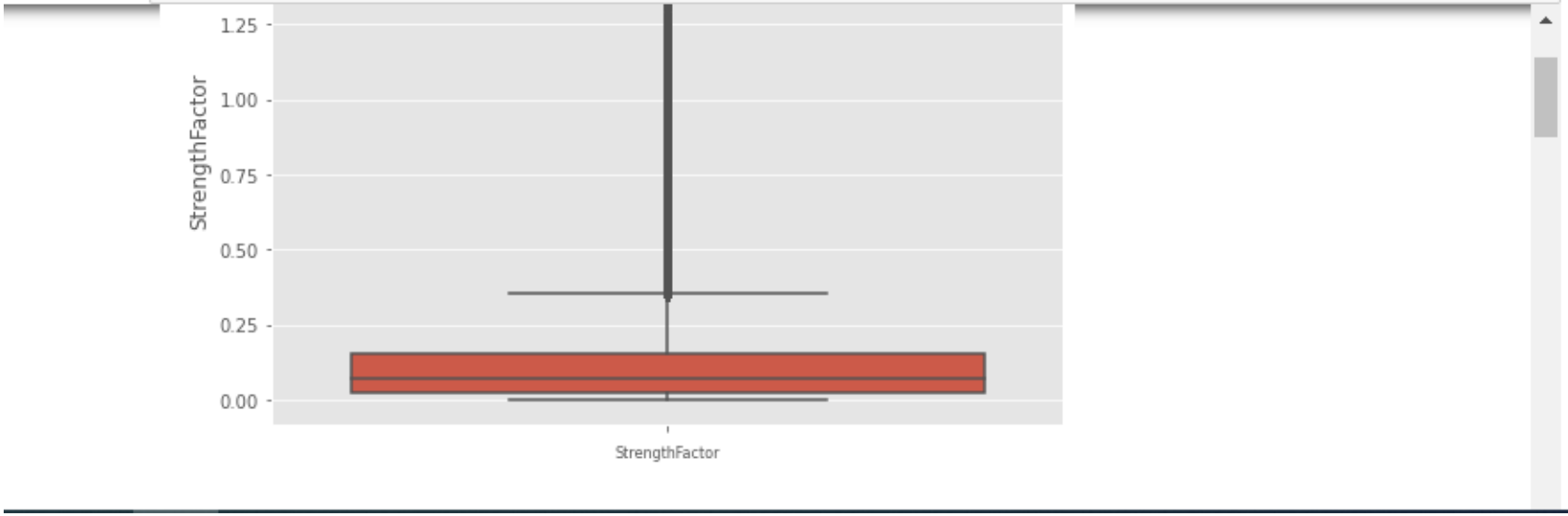

```
sales_data_hist['MarketingType'].value_counts().plot.bar(title="Freq dist of MarketingFile Type - hist states")
```

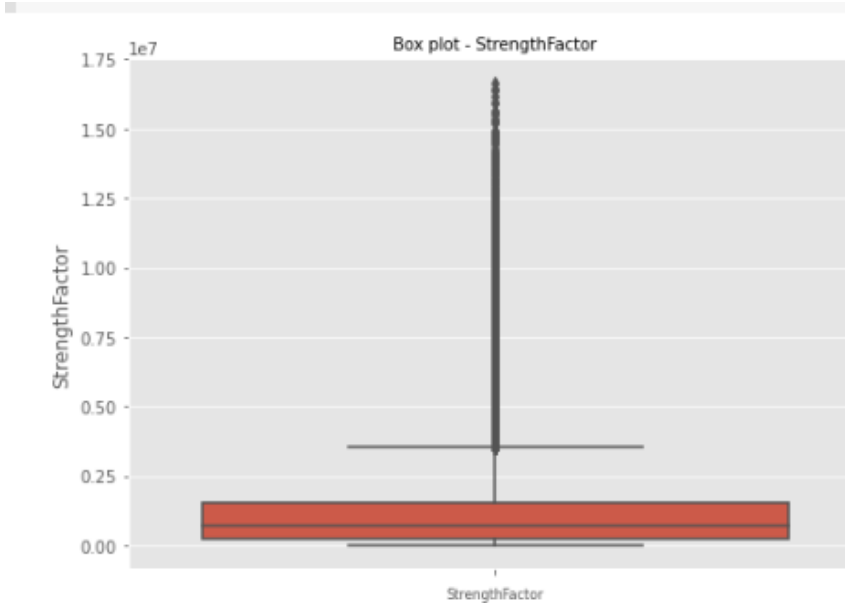
```
<AxesSubplot:title={'center':'Freq dist of MarketingFile Type - hist states'}>
```

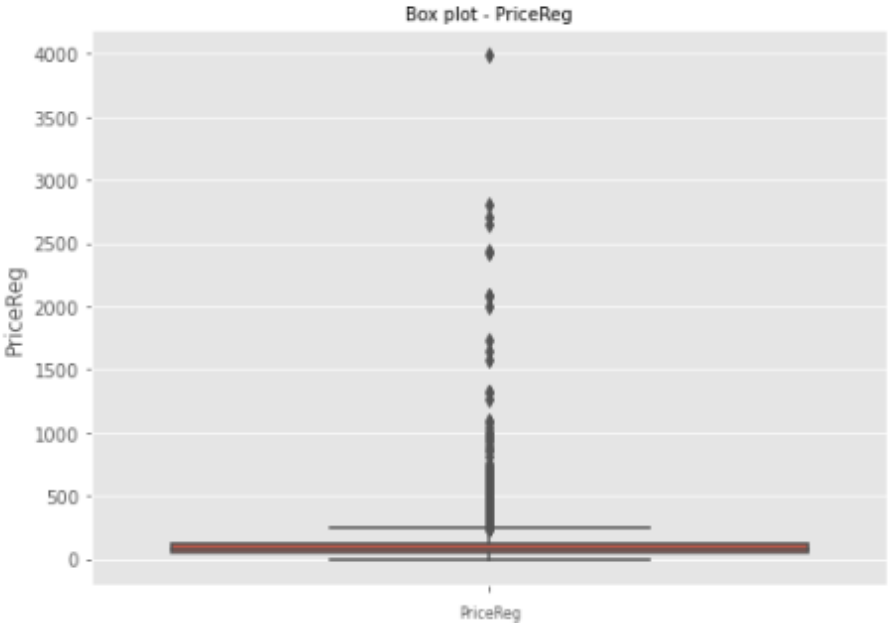


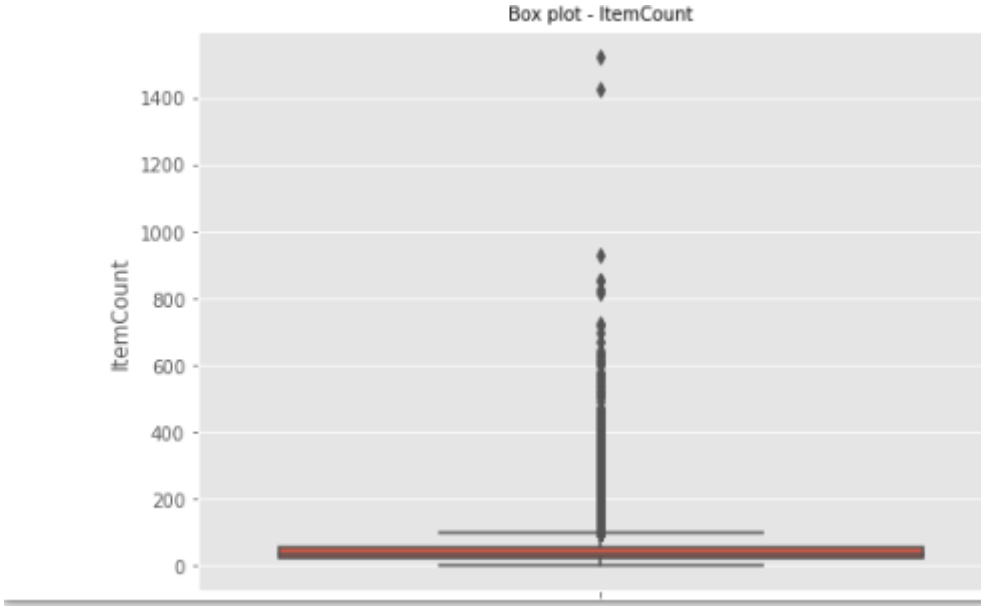
- Univariate outlier detection To analysis the outliers in the numericfeatures of the dataset

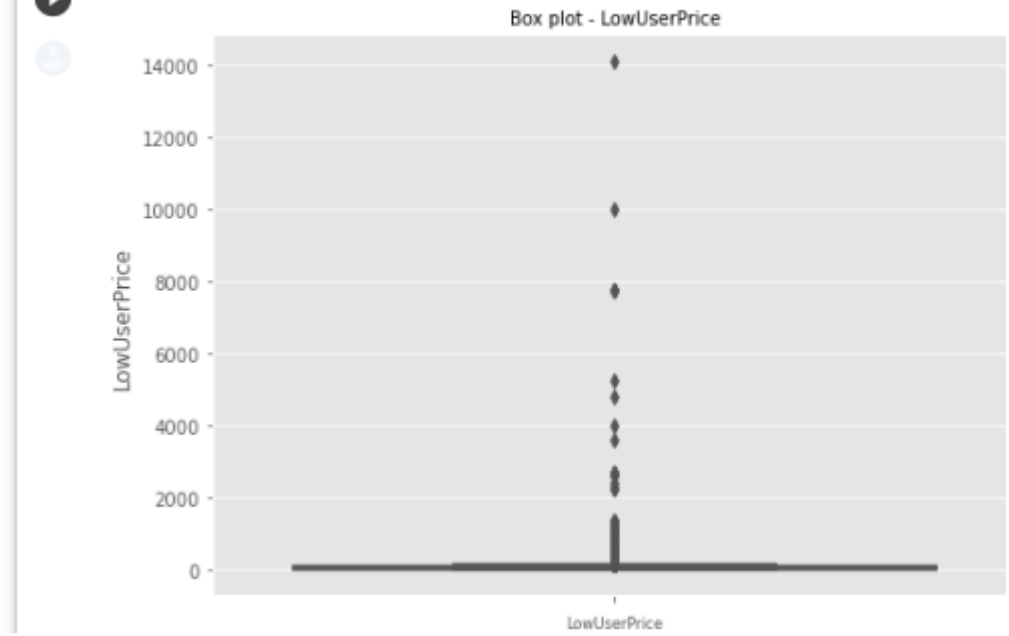
In [13]:
```
col_names = ['StrengthFactor','PriceReg', 'ReleaseYear', 'ItemCount', 'LowUserPrice', 'LowNetPrice']

fig, ax = plt.subplots(len(col_names), figsize=(8,40))

for i, col_val in enumerate(col_names):

    sns.boxplot(y=sales_data_hist[col_val], ax=ax[i])
    ax[i].set_title('Box plot - '+col_val, fontsize=10)
    ax[i].set_xlabel(col_val, fontsize=8)

plt.show()
```

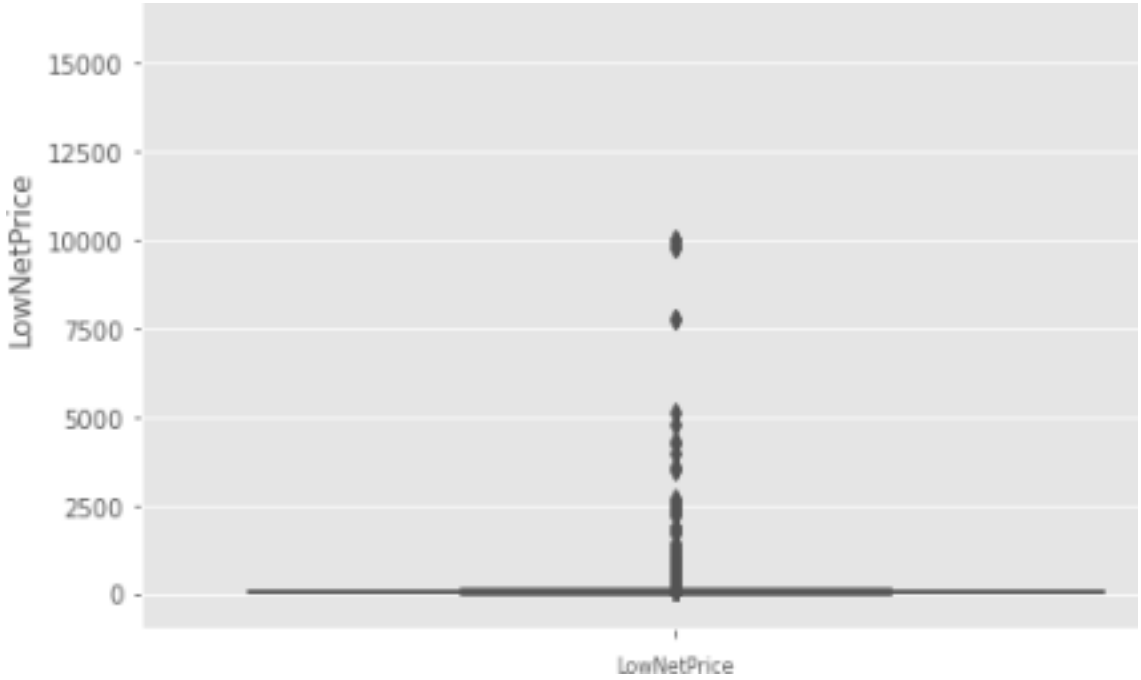



Box plot - StrengthFactor
1e7
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
StrengthFactor
StrengthFactor


Box plot - PriceReg
4000
3500
3000
2500
2000
1500
1000
500
0
PriceReg
PriceReg


Box plot - ItemCount
ItemCount
1400
1200
1000
800
600
400
200
0

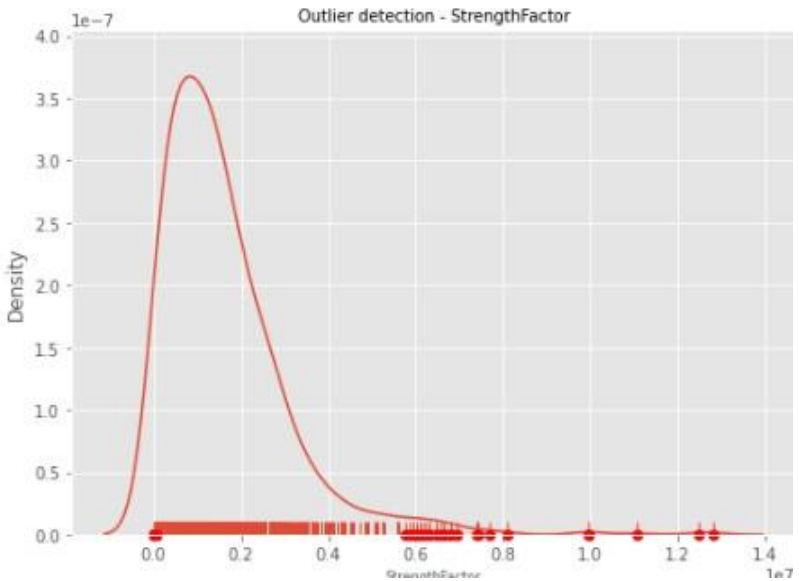

Box plot - LowUserPrice
LowUserPrice
14000
12000
10000
8000
6000
4000
2000
0
LowUserPrice

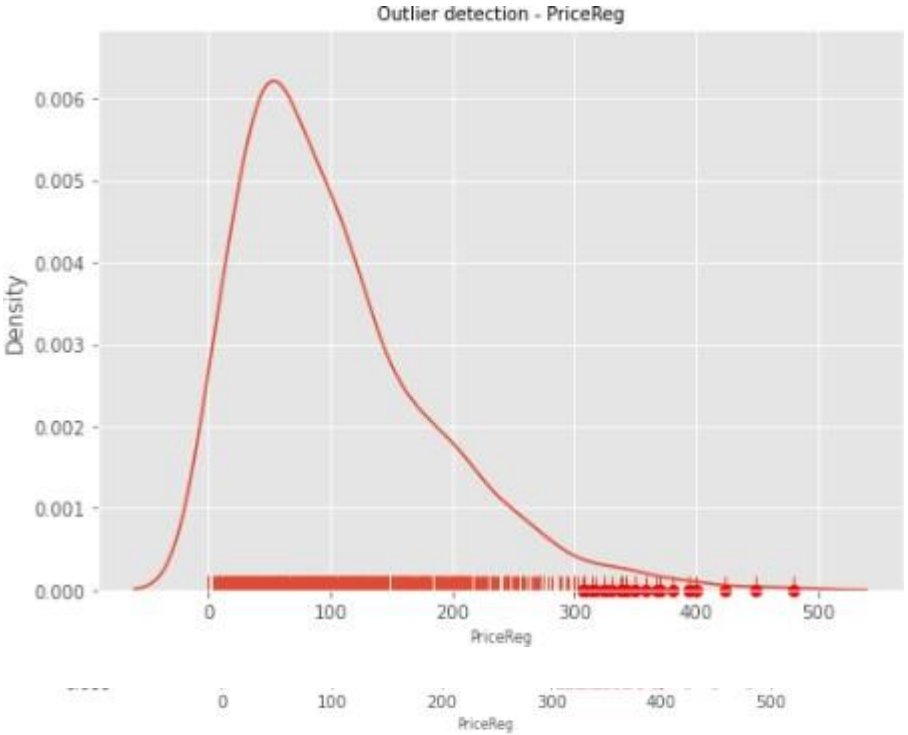


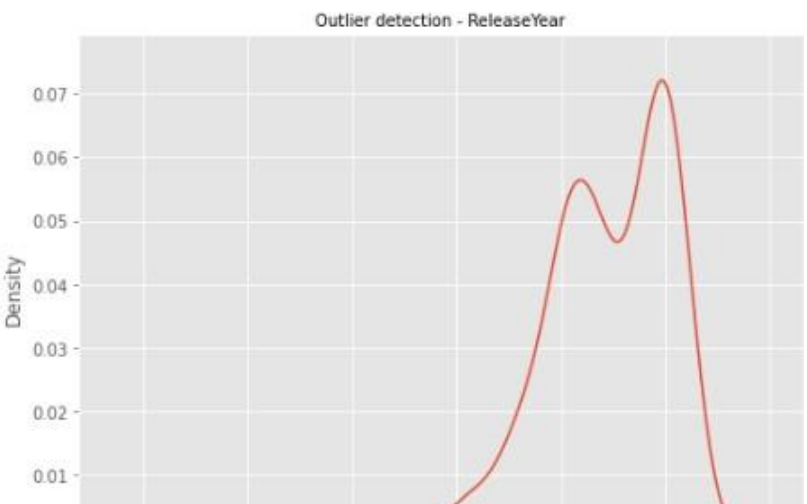
- By using Univariate Outlier treatment Many algorithms are sensitiveto the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the results and make results less reliable, that's why we have to recognize all the outliers and treat them.

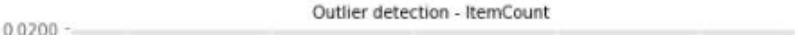
In [14]:
```
# Percentile based outlier removal
def percentile_based_outlier(data, threshold=95):
    diff = (100 - threshold) / 2.0
    minval, maxval = np.percentile(data, [diff, 100 - diff])
    return (data < minval) | (data > maxval)

col_names = ['StrengthFactor','PriceReg', 'ReleaseYear', 'ItemCount', 'LowUserPrice', 'LowNetPrice']

fig, ax = plt.subplots(len(col_names), figsize=(8,40))

for i, col_val in enumerate(col_names):
    x = sales_data_hist[col_val][:1000]
    sns.distplot(x, ax=ax[i], rug=True, hist=False)
    outliers = x[percentile_based_outlier(x)]
    ax[i].plot(outliers, np.zeros_like(outliers), 'ro', clip_on=False)

    ax[i].set_title('Outlier detection - '+col_val, fontsize=10)
    ax[i].set_xlabel(col_val, fontsize=8)

plt.show()
```

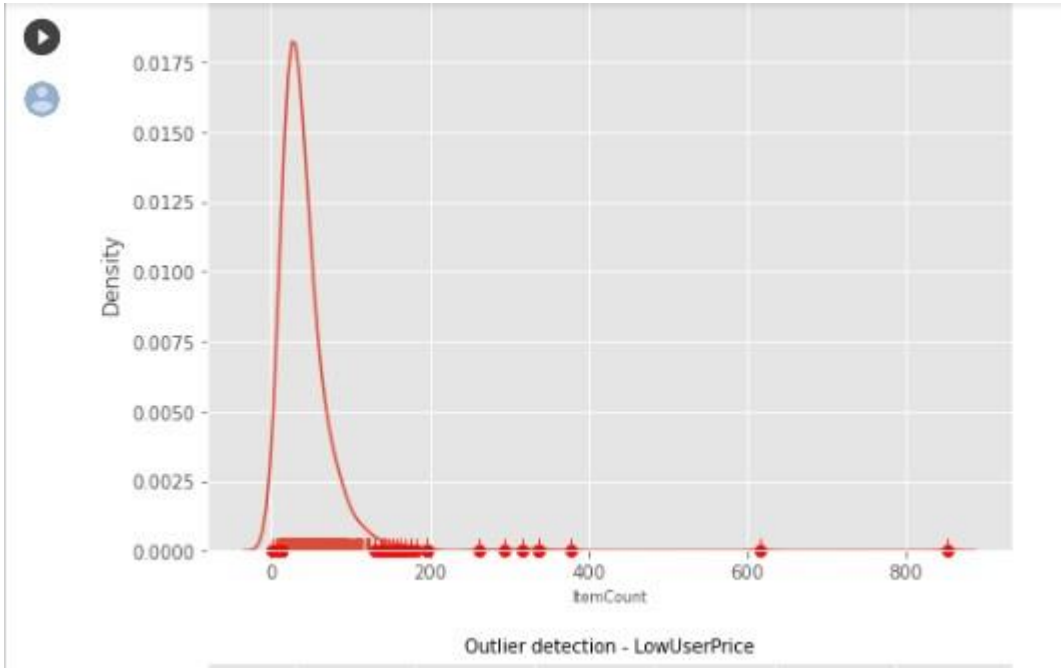
x}




Outlier detection - PriceReg
Density
0.006
0.005
0.004
0.003
0.002
0.001
0.000
0
100
200
300
400
500
PriceReg


Outlier detection - ReleaseYear
Density
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
1960
1970
1980
1990
2000
2010
2020
ReleaseYear


Outlier detection - ItemCount
0.0200



In [15]:
```
# Converting maarketing type to categorical variable
sales_data['MarketingType'] = sales_data['MarketingType'].astype('category')
sales_data['MarketingType'] = sales_data['MarketingType'].cat.codes

# Splitting the historical and active state
sales_data_hist = sales_data[sales_data['File_Type'] == 'Historical']
sales_data_act = sales_data[sales_data['File_Type'] == 'Active']
```

In [16]:
```
# Columns to remove
remove_col_val = ['Order', 'File_Type', 'SKU_number', 'SoldCount', 'ReleaseNumber', 'SoldFlag']

y = sales_data_hist['SoldFlag']

sales_data_hist = sales_data_hist.drop(remove_col_val, axis=1)
sales_data_act = sales_data_act.drop(remove_col_val, axis=1)
```

- We need to create training and testing variables

In [17]:
```
# create training and testing vars
training_features, testing_features, training_target, testing_target = train_test_split(sales_data_hist
print(training_features.shape, training_target.shape)
print(testing_features.shape, testing_target.shape)
```

```
(60796, 8) (60796,)
(15200, 8) (15200,)
```

In [18]:
```
print("Class 0 numbers: " , len(training_target[training_target==0.0]))
print("Class 1 numbers: " , len(training_target[training_target==1.0]))
```

```
Class 0 numbers:  50405
Class 1 numbers:  10391
```

In [23]:
```
clf_rf = RandomForestClassifier(n_estimators=25, random_state=12)
clf_rf.fit(x_train_res, y_train_res)
```

Out[23]:
```
RandomForestClassifier(n_estimators=25, random_state=12)
```

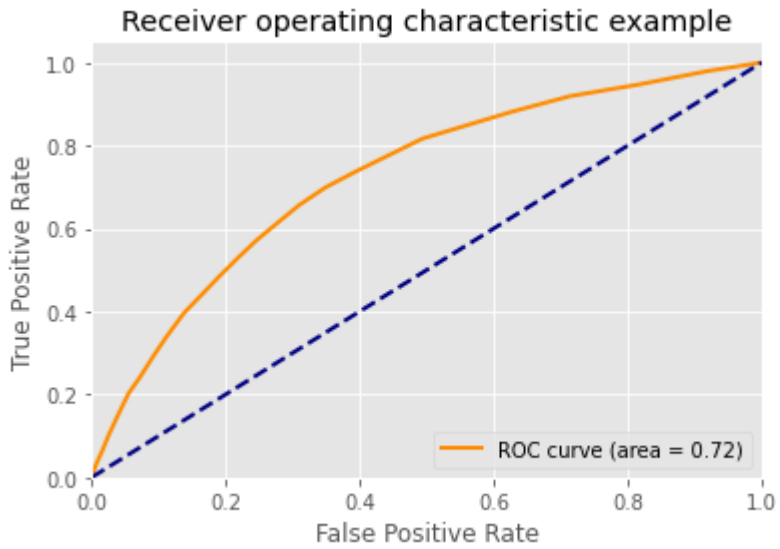
In [24]:
```
print('Validation Results')
print(clf_rf.score(x_val, y_val))
print(recall_score(y_val, clf_rf.predict(x_val)))

pred = clf_rf.predict(testing_features)

print('\nTest Results')
print(clf_rf.score(testing_features, testing_target))
print(recall_score(testing_target, pred))

print('\nROC AUC score')
print(roc_auc_score(testing_target, pred))
```

```
Validation Results
0.7481907894736842
0.5268714011516314

Test Results
0.7384868421052632
0.5289827255278311

ROC AUC score
0.6554004536730066
```

ROC AUC score 0.6554004536730066

- Finding the Compute ROC area for each class.

In [26]:
```
# Compute ROC curve and ROC area for each class
fpr = dict()
tpr = dict()
roc_auc = dict()
n_classes = 2
y_score = clf_rf.predict_proba(testing_features)

# Compute micro-average ROC curve and ROC area
fpr["micro"], tpr["micro"], _ = roc_curve(testing_target.ravel(), y_score[:,1].ravel())
roc_auc["micro"] = auc(fpr["micro"], tpr["micro"])

plt.figure()
lw = 2

plt.plot(fpr['micro'], tpr['micro'], color='darkorange',
         lw=lw, label='ROC curve (area = %0.2f)' % roc_auc['micro'])

plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic example')
plt.legend(loc="lower right")
plt.show()
```



MARKET/CUSTOMER/BUSINESS-NEED ASSESSMENT

One of the most valuable assets of a company is its inventory. In various industries, such as retail, food services and manufacturing, a lack of inventory can have detrimental effects. Aside from being a liability, inventory can also be considered a risk. It can be prone to theft, damage, and spoilage. Having a large inventory can also lead to reduction in sales. Regardless of the size of your company, having a proper inventory management system is very important for any business. It can help you keep track of all your supplies and determine the exact prices. It can also help you manage sudden changes in demand without sacrificing customer experience or product quality. This is especially important for brands looking to become a more customer-centric organization. Balancing the risks of overstocks and shortages is an especially challenging process for companies with complex supply chains. A company's inventory is typically a current asset that it plans to sell within a year. It must be measured and counted regularly to be considered a current asset.

SOURCE CODE

GitHub Link : https://github.com/fayaz123-coder/Inventory-Management-ML.git

FINANCIAL MODELING

In finance modeling, on Inventory Management should be collaborative. Collaborative reduces error, speeds up development time and lowers cost. The Financial Modeling Handbook is a collaborative, crowd-sourced guide to building better financial models using the FAST standard.

FORMULA : The cost per widget x widget sold x CPI escalation. CPI escalation is currently marked as a placeholder (a temporary modeling solution)

CONCLUSION:

The Demand Forecasting will help Small/Medium businesses to maintain inventory and minimize manual labour. Allowing it to reduce the capital spent on maintaining inventory. In this process it aims at simultaneously improve in profitability. With the forecasting technique, the overstock and stock-out of items are reduced as the stocks are ordered based on the Demand. In the future, the accuracy of the model can be improved by incorporating categorical embedding in neural networks as it is still a budding topic in the field of neural networks and requires more research