‘Predicto’

An app used for the prediction of the sales of the small and medium business

Swapnil Biswas

06/03/2022

Abstract

In this report I have proposed the idea of making an app named Predicto that would help to predict the future sales of the small and medium business of our Indian market.

Most of the people who have opened their business aren’t able to sustain in the tough market due to insufficiency of the market prediction. In current scenario businesses can't get away from are artificial intelligence (AI) and data analytics. These two technologies are a match made in heaven that allows businesses to not only collect massive troves of data, but use machine learning to make sense of that data. The insights businesses gain in this way can be used to better target marketing campaigns or find new efficiencies in internal processes.

GOAL OF THIS APP:

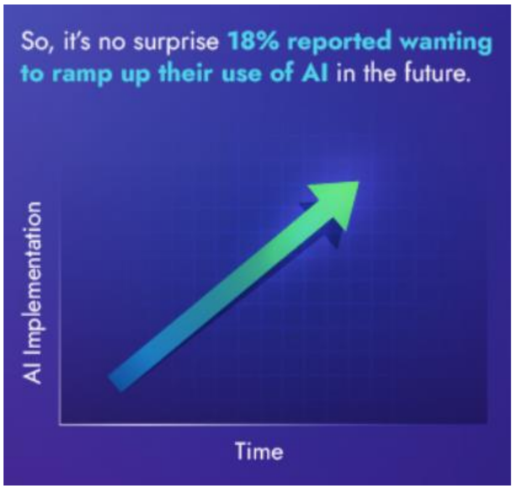
The aim of the app is to forecast store sales of the small and medium businesses of the Indian market.

This app will accurately predict the unit sales for thousands of items sold in a given shop and help the owner to take accurate decisions and flourish their business.

BUSINESS NEED ASSESMENT:

Big industrial companies have been the main AI adopters, using it primarily to power robotics for manufacturing. They're also using it to build bleeding-edge machine-learning (ML) algorithms for advanced data-processing techniques, such as predictive analytics.

Forecasts are especially to brick -and -mortar grocery stores which must dance delicately by providing the accurate information to the owner how much goods to buy and how much sale that is going to take in specific period of time.



3. Target Specification:

The main target of this product would be owners of the small and medium business who can use this app to predict their sales by giving the history of their sales records. With the high efficiency of this app, they can make accurate decisions about the efficient growth of their company.

4. External Search:

Dataset for the sales forecasting can found in https://www.kaggle.com/c/store-sales-time-series-forecasting/data

4.1 Dataset Analysis

This datasets consists of the following datasets oil , holidays , transactions , train , test.

In [71]:
```python
print(df_oil)
```

```
            dcoilwtico
date
2013-01-01         NaN
2013-01-02       93.14
2013-01-03       92.97
2013-01-04       93.12
2013-01-07       93.20
...                ...
2017-08-25       47.65
2017-08-28       46.40
2017-08-29       46.46
2017-08-30       45.96
2017-08-31       47.26

[1218 rows x 1 columns]
```

In [72]:
```python
print(df_holidays)
```

```
                  type    locale locale_name                    description  \
date
2012-03-02     Holiday     Local       Manta             Fundacion de Manta
2012-04-01     Holiday  Regional    Cotopaxi  Provincializacion de Cotopaxi
2012-04-12     Holiday     Local      Cuenca            Fundacion de Cuenca
2012-04-14     Holiday     Local    Libertad      Cantonizacion de Libertad
2012-04-21     Holiday     Local    Riobamba      Cantonizacion de Riobamba
...                ...       ...         ...                            ...
2017-12-22  Additional  National     Ecuador                     Navidad-3
2017-12-23  Additional  National     Ecuador                     Navidad-2
2017-12-24  Additional  National     Ecuador                     Navidad-1
2017-12-25     Holiday  National     Ecuador                       Navidad
2017-12-26  Additional  National     Ecuador                     Navidad+1

            transferred
date
2012-03-02        False
2012-04-01        False
2012-04-12        False
2012-04-14        False
2012-04-21        False
...                 ...
2017-12-22        False
2017-12-23        False
2017-12-24        False
2017-12-25        False
2017-12-26        False

[350 rows x 5 columns]
```

In [73]: `print(df_stores)`

```
                   city                           state type  cluster
store_nbr
1                 Quito                       Pichincha    D       13
2                 Quito                       Pichincha    D       13
3                 Quito                       Pichincha    D        8
4                 Quito                       Pichincha    D        9
5         Santo Domingo  Santo Domingo de los Tsachilas    D        4
6                 Quito                       Pichincha    D       13
7                 Quito                       Pichincha    D        8
8                 Quito                       Pichincha    D        8
9                 Quito                       Pichincha    B        6
10                Quito                       Pichincha    C       15
11              Cayambe                       Pichincha    B        6
12            Latacunga                        Cotopaxi    C       15
13            Latacunga                        Cotopaxi    C       15
14             Riobamba                      Chimborazo    C        7
15               Ibarra                        Imbabura    C       15
16        Santo Domingo  Santo Domingo de los Tsachilas    C        3
17                Quito                       Pichincha    C       12
18                Quito                       Pichincha    B       16
19             Guaranda                         Bolivar    C       15
20                Quito                       Pichincha    B        6
21        Santo Domingo  Santo Domingo de los Tsachilas    B        6
22                 Puyo                         Pastaza    C        7
23               Ambato                      Tungurahua    D        9
24            Guayaquil                          Guayas    D        1
25              Salinas                     Santa Elena    D        1
26            Guayaquil                          Guayas    D       10
27                Daule                          Guayas    D        1
28            Guayaquil                          Guayas    E       10
29            Guayaquil                          Guayas    E       10
30            Guayaquil                          Guayas    C        3
31             Babahoyo                        Los Rios    B       10
32            Guayaquil                          Guayas    C        3
33              Quevedo                        Los Rios    C        3
34            Guayaquil                          Guayas    B        6
35               Playas                          Guayas    C        3
36             Libertad                          Guayas    E       10
37               Cuenca                           Azuay    D        2
38                 Loja                            Loja    D        4
39               Cuenca                           Azuay    B        6
40              Machala                          El Oro    C        3
41              Machala                          El Oro    D        4
42               Cuenca                           Azuay    D        2
43           Esmeraldas                      Esmeraldas    E       10
44                Quito                       Pichincha    A        5
45                Quito                       Pichincha    A       11
46                Quito                       Pichincha    A       14
47                Quito                       Pichincha    A       14
48                Quito                       Pichincha    A       14
49                Quito                       Pichincha    A       11
50               Ambato                      Tungurahua    A       14
51            Guayaquil                          Guayas    A       17
52                Manta                          Manabi    A       11
53                Manta                          Manabi    D       13
54            El Carmen                          Manabi    C        3
```

In [75]:
```
print(df_train)
```

```
               date  store_nbr                      family     sales  \
id
0        2013-01-01          1                  AUTOMOTIVE     0.000
1        2013-01-01          1                   BABY CARE     0.000
2        2013-01-01          1                      BEAUTY     0.000
3        2013-01-01          1                   BEVERAGES     0.000
4        2013-01-01          1                       BOOKS     0.000
...             ...        ...                         ...       ...
3000883  2017-08-15          9                     POULTRY   438.133
3000884  2017-08-15          9              PREPARED FOODS   154.553
3000885  2017-08-15          9                     PRODUCE  2419.729
3000886  2017-08-15          9  SCHOOL AND OFFICE SUPPLIES   121.000
3000887  2017-08-15          9                     SEAFOOD    16.000

         onpromotion
id
0                  0
1                  0
2                  0
3                  0
4                  0
...              ...
3000883            0
3000884            1
3000885          148
3000886            8
3000887            0

[3000888 rows x 5 columns]
```

In [76]:
```
print(df_test)
```

```
               date  store_nbr                      family  onpromotion
id
3000888  2017-08-16          1                  AUTOMOTIVE            0
3000889  2017-08-16          1                   BABY CARE            0
3000890  2017-08-16          1                      BEAUTY            2
3000891  2017-08-16          1                   BEVERAGES           20
3000892  2017-08-16          1                       BOOKS            0
...             ...        ...                         ...          ...
3029395  2017-08-31          9                     POULTRY            1
3029396  2017-08-31          9              PREPARED FOODS            0
3029397  2017-08-31          9                     PRODUCE            1
3029398  2017-08-31          9  SCHOOL AND OFFICE SUPPLIES            9
3029399  2017-08-31          9                     SEAFOOD            0

[28512 rows x 4 columns]
```

In [78]:
```
## finding the missing values in the oil datasets

print(df_oil.dcoilwtico.isna().sum())
df_oil.head()
```

43

Out[78]:

date	dcoilwtico
2013-01-01	NaN
2013-01-02	93.14
2013-01-03	92.97
2013-01-04	93.12
2013-01-07	93.20

In [79]:
```
## filling the NAN values

na_index = df_oil[df_oil.dcoilwtico.isna()].index
df_oil.loc[na_index, "dcoilwtico"] = (df_oil.dcoilwtico.fillna(method="ffill") + df_oil.dcoilwtico.fillna(method="bfill"))/2
df_oil.dcoilwtico[0] = df_oil.dcoilwtico[1]
```

Here we can see that in the oil dataset there are 43 missing values which I have filled with the mean of the values of the ‘dcoilwtico’ columns.

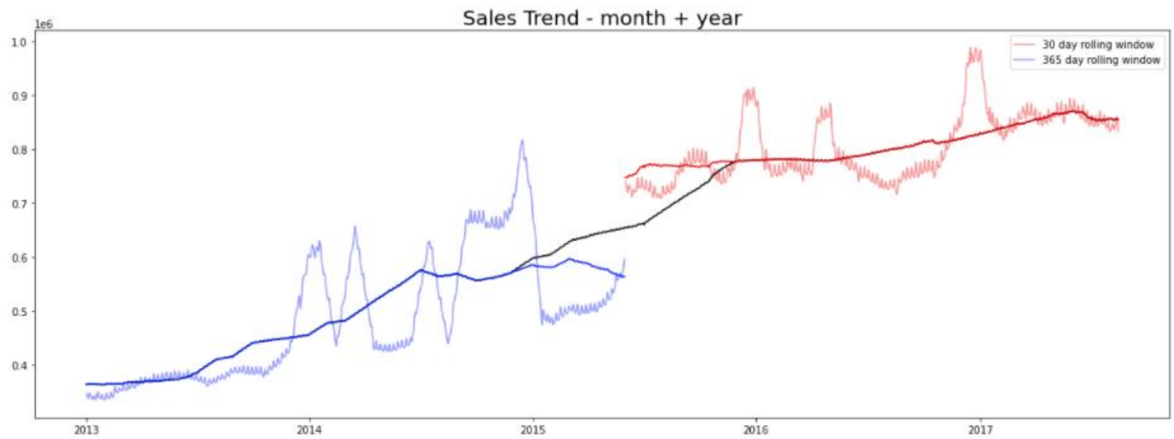
4.2 Data Visualization

In [80]:
```
# plotting the oil prices and a moving average over a week

fig, ax = plt.subplots(figsize=(20,7))
ax.plot(df_oil.rolling(window=7,
                       center=True,
                       min_periods=3).mean(),
        linewidth=2,
        color="red")
sns.scatterplot(data=df_oil, x="date", y="dcoilwtico", color="0.5", alpha=0.5, ax=ax)
sns.lineplot(data=df_oil, x="date", y="dcoilwtico", alpha=0.5, ax=ax, linewidth=0.5)
ax.set_title("Oil Prices", fontsize=18)
```

I worked upon the train data where I implemented the visualizations of the Sales Trends.

In [83]:
```
## Plotting for the Sales Trend


fig, ax = plt.subplots(figsize=(20,7))
data_lim = df_train.loc[:, ["date", "sales"]][df_train.date>="2015-06"].groupby("date").sum()
data_pre = df_train.loc[:, ["date", "sales"]][df_train.date<"2015-06"].groupby("date").sum()
data = df_train.loc[:, ["date", "sales"]].groupby("date").sum()
#data.head()
ax.plot(data_lim.rolling(window=30,
                    center=True,
                    min_periods=15).mean(),
        aa=True,
        color="red",
        alpha=0.4
       )
ax.plot(data_pre.rolling(window=30,
                    center=True,
                    min_periods=15).mean(),
        aa=True,
        color="blue",
        alpha=0.4
       )
ax.plot(data.rolling(window=365,
                    center=True,
                    min_periods=182).mean(),
       aa=True,
       color="black"
       )
ax.plot(data_lim.rolling(window=365,
                        center=True,
                        min_periods=182).mean(),
       aa=True,
       color="red")
ax.plot(data_pre.rolling(window=365,
                        center=True,
                        min_periods=182).mean(),
       aa=True,
       color="blue")
ax.legend(['30 day rolling window', '365 day rolling window'])
ax.set_title("Sales Trend - month + year", fontsize=20)
```

Out[83]: `Text(0.5, 1.0, 'Sales Trend - month + year')`



1. red = post 2015-06
2. blue = pre 2015-06

We can see a couple of notable things.

- over the years we see an upwards trend of the sales.
- over the month-wide average, there is a clear spike at the end of the years, which gets more prominent in later years
- 2013 is almost flat apart from christmas
- 2014 is all over the place with large spikes an valleys with another spike mid 2015 that seems to establish a new baseline
- after about half the year of 2015, the chart seems to flatten out, showing only 3 major spikes
    - two end-of-year spikes as usual
    - one spike around the end of the first quarter 2016, probably related to the April 16, 2016 earthquake and subsequent donations happening
- the monthly line is ragged, showing some periodic behaviour.

In [30]:
```
average_sales = df_train.groupby('date').mean()['sales']
df = average_sales.to_frame()


time = np.arange(len(df.index))

df['time'] = time


X = df.loc[:, ['time']]  # features
y = df.loc[:, 'sales']  # target

# Train the model
model = LinearRegression()
model.fit(X, y)

# Store the fitted values as a time series with the same time index as
# the training data
y_pred = pd.Series(model.predict(X), index=X.index)
print(y_pred)
```

```
date
2013-01-01    194.232790
2013-01-02    194.427137
2013-01-03    194.621484
2013-01-04    194.815831
2013-01-05    195.010178
                 ...
2017-08-11    520.541320
2017-08-12    520.735667
2017-08-13    520.930014
2017-08-14    521.124361
2017-08-15    521.318708
Length: 1684, dtype: float64
```

I applied Linear Regression to predict the sales of the given datasets.

4.3 Applicable Constraints

i. The company datasets need to be stored with high confidentiality.

ii. A large period of historical data needs to be taken as an input inorder to prevent overfitting.

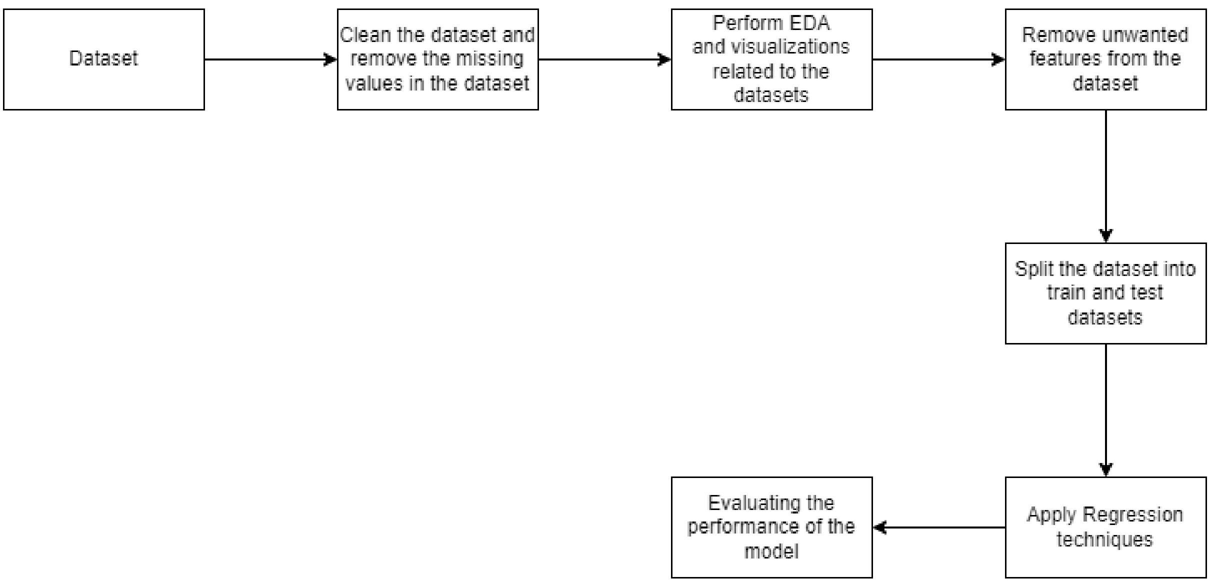
4.4 Applicable Patents

The process to extract to insights and predict accurate results from the datasets given as an input by the user using efficient machine learning algorithms can be considered as the patent for the given statement problem.

5. Business opportunity:

This product can be used by owners of all small and medium business to predict their sales accurately so that they can make well structured plan to flourish their business.

6. Product Prototype:



7. Product Details

This product is working on the dataset taken from Kaggle https://www.kaggle.com/c/store-sales-time-series-forecasting/data.

But this app would have a database management system where the large quantities of the data given by the user would be evaluated by the regression techniques that provide the business with insights that result in tangible business value and help the owners to make efficient solutions to flourish their sales.

8. Conclusion:

Artificial intelligence refers to the simulation of human intelligence in machines.

Artificial intelligence in business sector streamlines the data science process so that the users get high quality predictions in fraction of seconds thus resulting in simplifying the tasks and reducing the human error.