Online retail II

Vo Minh Thien An (s3916570)
Do Ha Minh Long (s3634734)
Nguyen Phuong Nam (s3877256)

RMIT UNIVERSITY

Retrieving and Preparing the Data

The origin dataset

```
In [4]: #display data
        data
```

Out[4]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom
...	...	...	...	...	...	...	...	...
525456	538171	22271	FELTCRAFT DOLL ROSIE	2	2010-12-09 20:01:00	2.95	17530.0	United Kingdom
525457	538171	22750	FELTCRAFT PRINCESS LOLA DOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525458	538171	22751	FELTCRAFT PRINCESS OLIVIA DOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525459	538171	20970	PINK FLORAL FELTCRAFT SHOULDER BAG	2	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525460	538171	21931	JUMBO STORAGE BAG SUKI	2	2010-12-09 20:01:00	1.95	17530.0	United Kingdom

525461 rows × 8 columns

RMIT UNIVERSITY

```
In [331]: pd.set_option("display.max_rows", None)
          data['Description'].value_counts().sort_index()

Out[331]: DOORMAT UNION JACK GUNS AND ROSES      53
          3 STRIPEY MICE FELTCRAFT              117
          4 PURPLE FLOCK DINNER CANDLES          17
          ANIMAL STICKERS                        12
          BLACK PIRATE TREASURE CHEST            14
          BROWN  PIRATE TREASURE CHEST            7
          Bank Charges                            3
```

```
In [6]: data.isna().sum()

Out[6]: Invoice          0
        StockCode        0
        Description   2928
        Quantity         0
        InvoiceDate      0
        Price            0
        Customer ID  107927
        Country          0
        dtype: int64
```

```
In [7]: data = data.dropna()
        data
```

Out[7]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom
...	...	...	...	...	...	...	...	...
525456	538171	22271	FELTCRAFT DOLL ROSIE	2	2010-12-09 20:01:00	2.95	17530.0	United Kingdom
525457	538171	22750	FELTCRAFT PRINCESS LOLA DOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525458	538171	22751	FELTCRAFT PRINCESS OLIVIA DOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525459	538171	20970	PINK FLORAL FELTCRAFT SHOULDER BAG	2	2010-12-09 20:01:00	3.75	17530.0	United Kingdom
525460	538171	21931	JUMBO STORAGE BAG SUKI	2	2010-12-09 20:01:00	1.95	17530.0	United Kingdom

417534 rows × 8 columns

```
In [8]: data.info()

<class 'pandas.cor
Int64Index: 417534
Data columns (tota
 #   Column
---  ------
 0   Invoice
 1   StockCode
 2   Description
 3   Quantity
 4   InvoiceDate
 5   Price
 6   Customer ID
 7   Country      417534 non-null  object
dtypes: datetime64[ns](1), float64(2), int64(1), object(4)
memory usage: 28.7+ MB
```

RMIT UNIVERSITY

Feature engineering

cription	Quantity	InvoiceDate	Price	Customer ID	Country	TotalPrice	month	year	month_year	revenue	Last Active
LIGHTS	12	2009-12-01 07:45:00	6.95	13085.0	United Kingdom	83.40	12	2009	2009-12-01	83.40	373 days 12:16:00
LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom	81.00	12	2009	2009-12-01	81.00	373 days 12:16:00
LIGHTS	12	2009-12-01 07:45:00	6.75	13085.0	United Kingdom	81.00	12	2009	2009-12-01	81.00	373 days 12:16:00
LESIZE	48	2009-12-01 07:45:00	2.10	13085.0	United Kingdom	100.80	12	2009	2009-12-01	100.80	373 days 12:16:00
ETBOX	24	2009-12-01 07:45:00	1.25	13085.0	United Kingdom	30.00	12	2009	2009-12-01	30.00	373 days 12:16:00
...	...	...	...	...	...	...	...	...	...	...	...
ROSIE	2	2010-12-09 20:01:00	2.95	17530.0	United Kingdom	5.90	12	2010	2010-12-01	5.90	0 days 00:00:00
ADOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom	3.75	12	2010	2010-12-01	3.75	0 days 00:00:00
ADOLL	1	2010-12-09 20:01:00	3.75	17530.0	United Kingdom	3.75	12	2010	2010-12-01	3.75	0 days 00:00:00
ERBAG	2	2010-12-09 20:01:00	3.75	17530.0	United Kingdom	7.50	12	2010	2010-12-01	7.50	0 days 00:00:00
AGSUKI	2	2010-12-09 20:01:00	1.95	17530.0	United Kingdom	3.90	12	2010	2010-12-01	3.90	0 days 00:00:00

RMIT UNIVERSITY

Ultra Mega Hyper Aim

We use the clustering for:

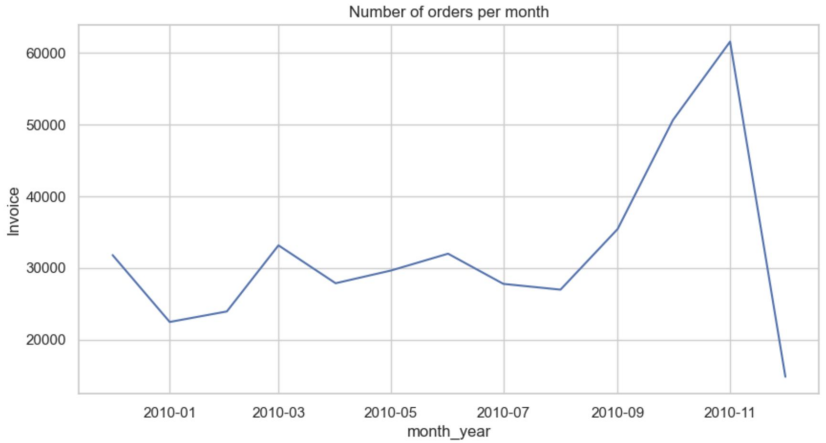
- Find the loyal customer
- Look for the way to make more for this group
- Find the reason
- Find the trend, the shopping habit

And we use regression for:

- Predict the total revenue of next year
- Predict revenue of the same month next year

Out[33]:

	Customer ID	TotalPrice	Invoice	Last Active
0	12346.0	-64.68	46	66
1	12347.0	1323.32	71	2
2	12348.0	222.16	20	73



RMIT UNIVERSITY

Clustering

1. We calculate customer's last active day
2. Merge column and delete outlier

So, What we do here?

```
# find the most recent date the user active
last= data['InvoiceDate'].max()
last
```

```
X = customers['TotalPrice'].quantile(0.05)
Y = customers['TotalPrice'].quantile(0.95)
iqr = Y - X
```

✓ 0.3s Python

```
#Use the quartile variable to remove outliers
customers = customers[(customers['TotalPrice'] >= X - 1.5*iqr) & (customers['TotalPrice'] < Y + 1.5*iqr)]
```

Python

```
recent = recent.reset_index()
recent['Last Active']= recent['Last Active'].dt.days
recent
```

Python

RMIT UNIVERSITY

Regression

1. Define X and y
2. Put it to train and test

```
In [99]: X = data["Price"].values
         y = data["revenue"].values

In [120]: X = data.drop(["revenue", "Description", "InvoiceDate", "Country", "month_year", "StockCode", "Invoice"], axis = 1).v
          y = data["revenue"].values

In [121]: # split to test and train set
          X_train,X_test,y_train,y_test=train_test_split(X,y, test_size=0.3, random_state=0)
          ml = LinearRegression()
          ml.fit(X_train, y_train)

Out[121]: LinearRegression()

In [122]: y_pred = ml.predict(X_test)
          plt.figure(figsize = (10,5))
          plt.scatter(y_test, y_pred)
          plt.xlabel("actual")
          plt.ylabel("predict")
          print("Accuracy: ", r2_score(y_pred, y_test) * 100)

          Accuracy:  100.0
```

RMIT UNIVERSITY

Conclusion

RMIT UNIVERSITY

Q&A Time

RMIT UNIVERSITY

Thank for listening

RMIT UNIVERSITY