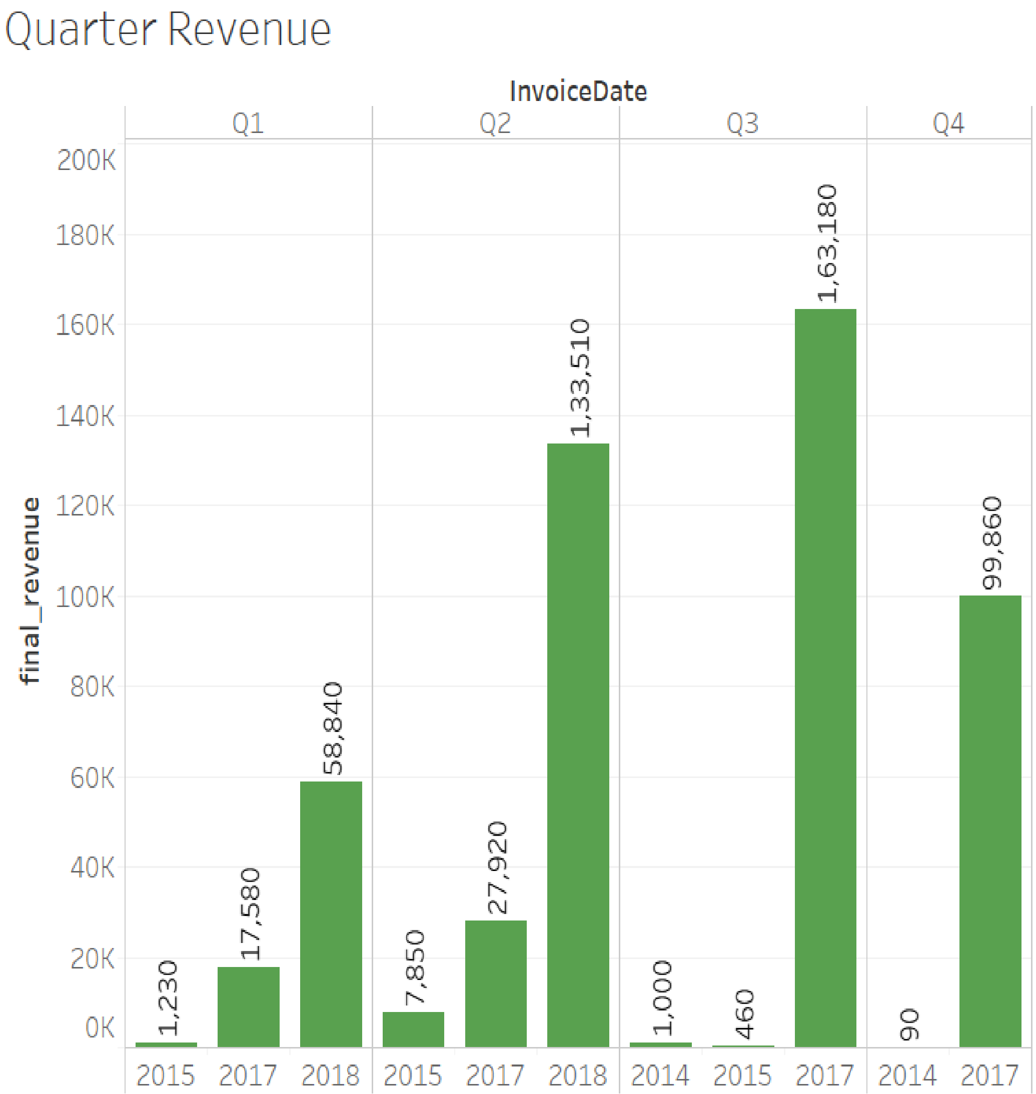
ABC – Non Technical Presentation

Date: 24-08-2020

Presenter: Monisha Anila

Decreasing product purchases across all stores up to 40% resulted increase in ware house maintenance costs up to 25% and discounts are unfavourable for marginal costs prediction of 2018 reporting period.



Analytics Insights:

1. Decreased revenue from retail stores UK(↓1,00,000), Canada(↓3000), Greece(↓100), Malta(↓10000), UAE(↓400) when compared with 2017.
2. Most sales are seasonal (1,30,000) purchases at weekends(80,000).
3. Collaboration with star brands in 2017 resulted profit (↑150000) but product and delivery issues hit bottom sales up to (40%).
4. Data driven decisions to be made across all stores to bring revenue losses up to 0% by the end of 2018.

Source: ABC retail data (2014-2018) and in Euros currency

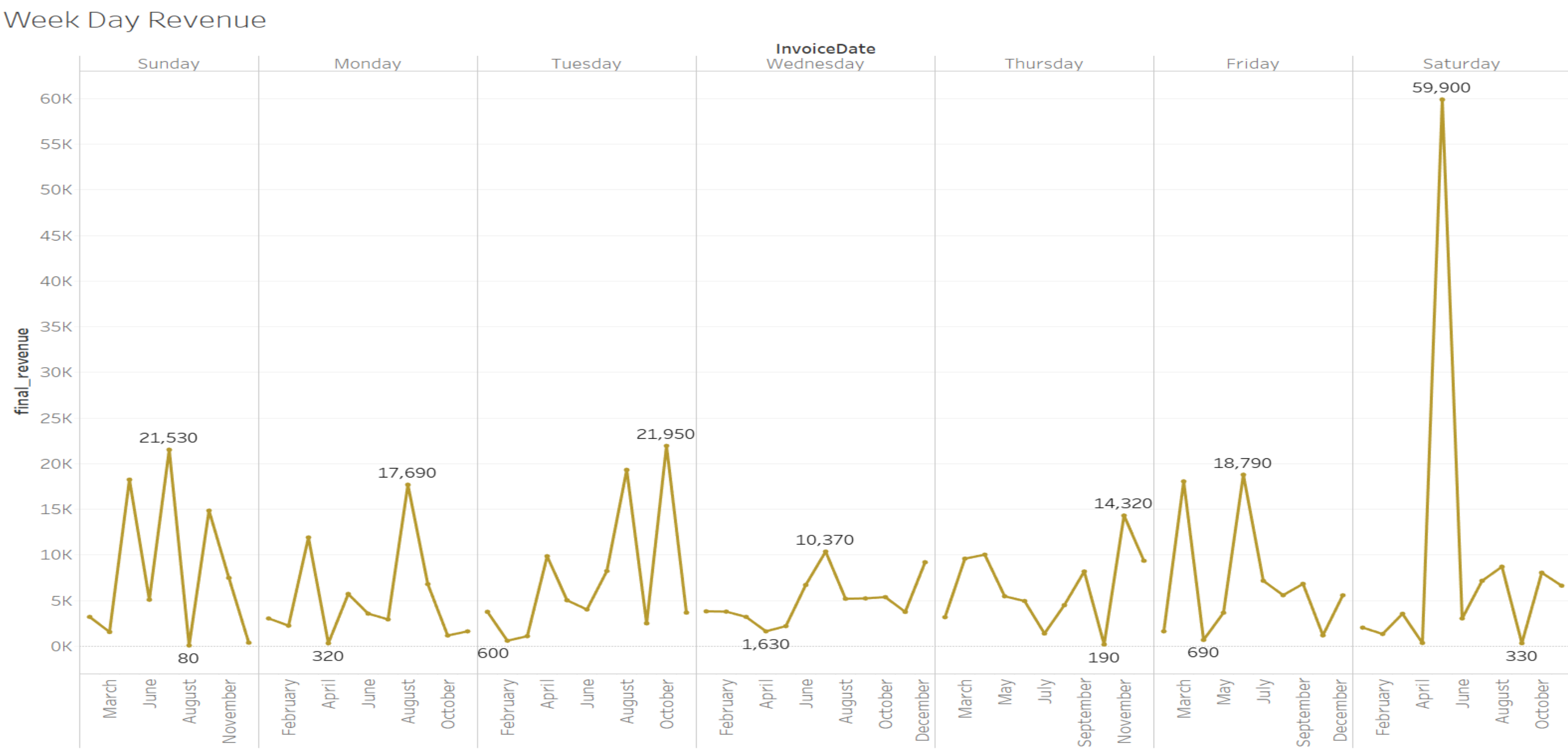
The formation SMART problem statement and it’s problem solving methods.

What opportunities exists for ABC(SME) start up after seeing revenue fallen by 25% during 1sthalf of 2018 compared to 2017 data to bring revenue losses by 0% at the end of 2018 through new product and delivery strategy or closing stores at low revenue?

Customer and Market Strategy

Product Divestment

Weekend sales boost up to 60% of total sales during July-August .

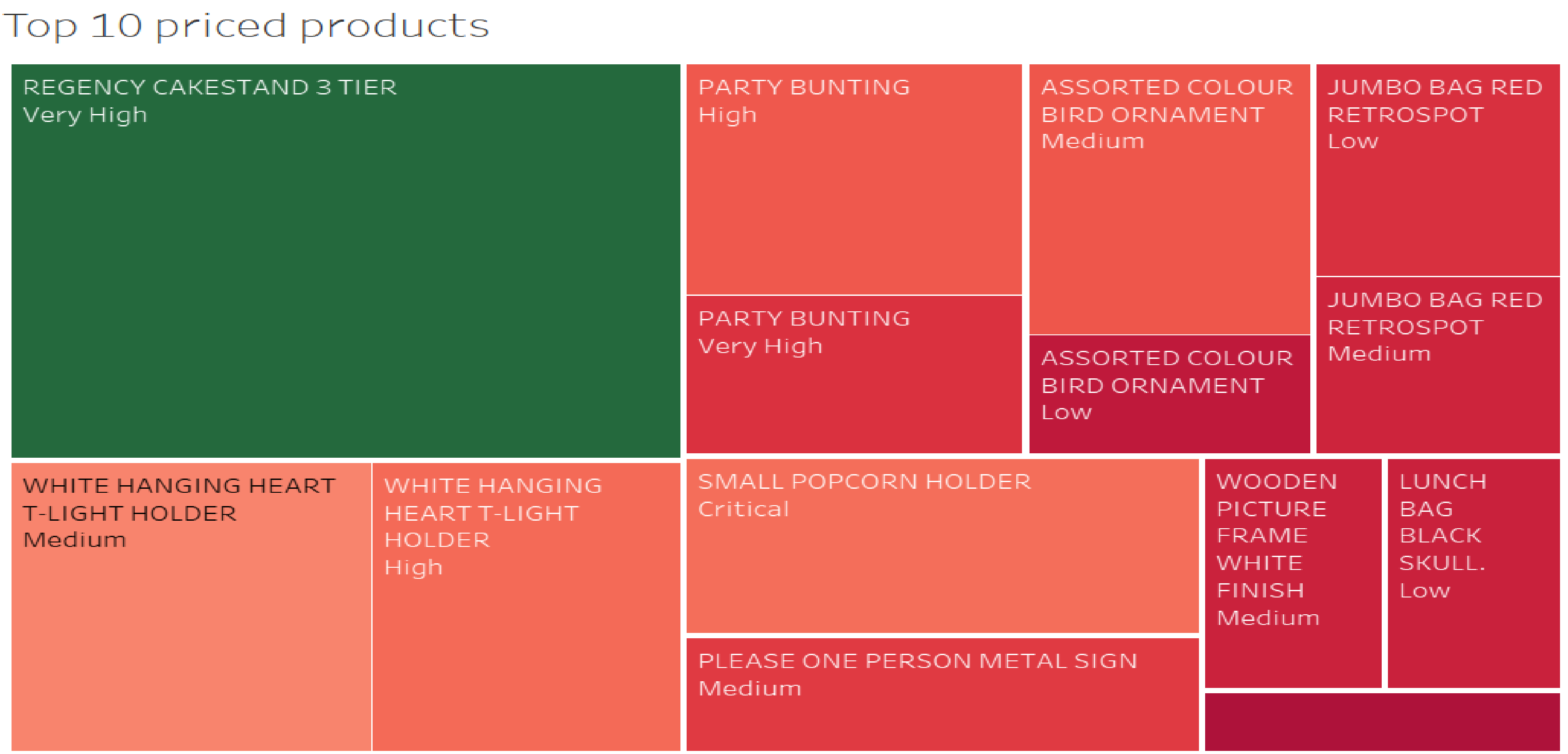


Analytical Insights:

1. Most sales are seasonal (1,30,000) purchases at weekends(80,000).
2. Purchase history shows orders on seasonal goods like Cake stand, Lights holder, Party bunders and bird ornaments.

Source: ABC retail data (2014-2018) in Euros currency

Customer segmentation using odd-even strategy shows revenue driven business up to 67% across all the stores.



Analytical Insights:

1. Price buckets – `[(0.099, 0.85] < (0.85, 1.65] < (1.65, 2.55] < (2.55, 4.25] < (4.25, 195.0]]`
   labels=['Critical','Low','Medium','High','Very High'])
2. Odd-even strategy holds good for light holders, party bunting , ornaments and wooden frames.

Source: ABC retail data (2014-2018) in Euros currency

While doing customer segmentation based on revenue and price. We found out almost revenue and price are directly proportional according to their bucket range.

Key Insights

- Revenue buckets `[(0.119, 2.58] < (2.58, 5.95] < (5.95, 12.6] < (12.6, 19.8] < (19.8, 3828.0]]`
- Price buckets `[(0.099, 0.85] < (0.85, 1.65] < (1.65, 2.55] < (2.55, 4.25] < (4.25, 195.0]]`
- Interestingly the low priced products brought 50% of revenue with high priced products leading to the best customer segmentation and we need product divestment implementation in best way.

Implementation of predictive model to avoid out of stock for the Top 10 products as they make 70% of revenue and analysis is done in UK.

```
X_train = train.drop('Quantity', axis=1).values # Drop the dependent variable
X_test = test.drop('Quantity', axis=1).values   # Drop the dependent variable
y_train = train['Quantity'].values              # Find the dependent variable
y_test = test['Quantity'].values                # Find the dependent variable
```

```
import lightgbm as lgb # Using LightGBM as predictive model
from sklearn.metrics import mean_squared_error # MSE for accuracy

lgb_train = lgb.Dataset(X_train, y_train) # Sending train date
lgb_eval = lgb.Dataset(X_test, y_test, reference=lgb_train) # Sending test data

params = {'task':'train', 'boosting_type':'gbdt', 'objective':'regression',
          'metric': {'rmse'}, 'num_leaves': 10, 'learning_rate': 0.05,
          'feature_fraction': 0.8, 'max_depth': 5, 'verbose': 0,
          'num_boost_round':20000, 'early_stopping_rounds':5000, 'nthread':-1} # setting model parameters

gbm = lgb.train(params,
                lgb_train,
                num_boost_round=20,
                valid_sets=lgb_eval,
                early_stopping_rounds=5) # sending values to models

# predict
y_pred = gbm.predict(X_test, num_iteration=gbm.best_iteration) # predict using test
# eval
print('The rmse of prediction is:', mean_squared_error(y_test, y_pred) ** 0.5) # RMSE for accuracy
```

```
The rmse of prediction is: 309.637776991945
```

Key Insights

```
# Array:Description,Country,UnitPrice,revenue_buckets,price_buckets,final_revenue,dayofmonth,dayofyear,dayofweek,month,
# year,weekofyear
X_prediction=np.array([[1,1,3,4,5,16,7,8,9,11,2018,30]]) # new values
predictions = gbm.predict(X_prediction) # predictive function
predictions # display 'Quantity'value
```

```
array([121.37791008])
```

Implementation of predictive model to avoid over pricing for the Top 10 products as they make 70% of revenue and analysis is done in UK.

```
X_train = train.drop('UnitPrice', axis=1).values  # Drop the dependent variable
X_test = test.drop('UnitPrice', axis=1).values    # Drop the dependent variable
y_train = train['UnitPrice'].values               # Find the dependent variable
y_test = test['UnitPrice'].values                 # Find the dependent variable
```

```
from sklearn.neighbors import KNeighborsClassifier # Importing predictive model function
from sklearn.metrics import mean_squared_error     # MSE for model accuracy

KNN_model= KNeighborsClassifier(n_neighbors=5)     # Setting model parameters
KNN_model.fit(X_train,y_train)



# predict
y_pred = KNN_model.predict(X_test)                  # Predict the test data
# eval
print('The rmse of prediction is:', mean_squared_error(y_test, y_pred) ** 0.5)  # RMSE for accuracy
```

```
The rmse of prediction is: 4.460941604639093
```

Key Insights

```
# Array format:Description,Country,Quantity,revenue_buckets,price_buckets,final_revenue,dayofmonth,dayofyear,dayofweek,
# month,year,weekofyear
X_prediction=np.array([[1,1,121,4,5,16,7,8,9,11,2018,30]]) # new data
predictions = KNN_model.predict(X_prediction) # predictive function
predictions # display optimised price
```

```
array([2], dtype=int64)
```

Introducing items brought together category to upsell the products with discounts from optimized price calculations in UK.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
2	(HEART OF WICKER LARGE)	(HEART OF WICKER SMALL)	0.054526	0.058888	0.030534	0.560000	9.509630	0.027323	2.138892
3	(HEART OF WICKER SMALL)	(HEART OF WICKER LARGE)	0.058888	0.054526	0.030534	0.518519	9.509630	0.027323	1.963678
6	(LUNCH BAG RED RETROSPOT)	(LUNCH BAG BLACK SKULL.)	0.052345	0.067612	0.031625	0.604167	8.935820	0.028086	2.355507
8	(LUNCH BAG SPACEBOY DESIGN)	(LUNCH BAG BLACK SKULL.)	0.042530	0.067612	0.031625	0.743590	10.997932	0.028749	3.636314
16	(WOODEN PICTURE FRAME WHITE FINISH)	(WOODEN FRAME ANTIQUE WHITE)	0.062159	0.043621	0.034896	0.561404	12.870175	0.032185	2.180545
17	(WOODEN FRAME ANTIQUE WHITE)	(WOODEN PICTURE FRAME WHITE FINISH)	0.043621	0.062159	0.034896	0.800000	12.870175	0.032185	4.689204

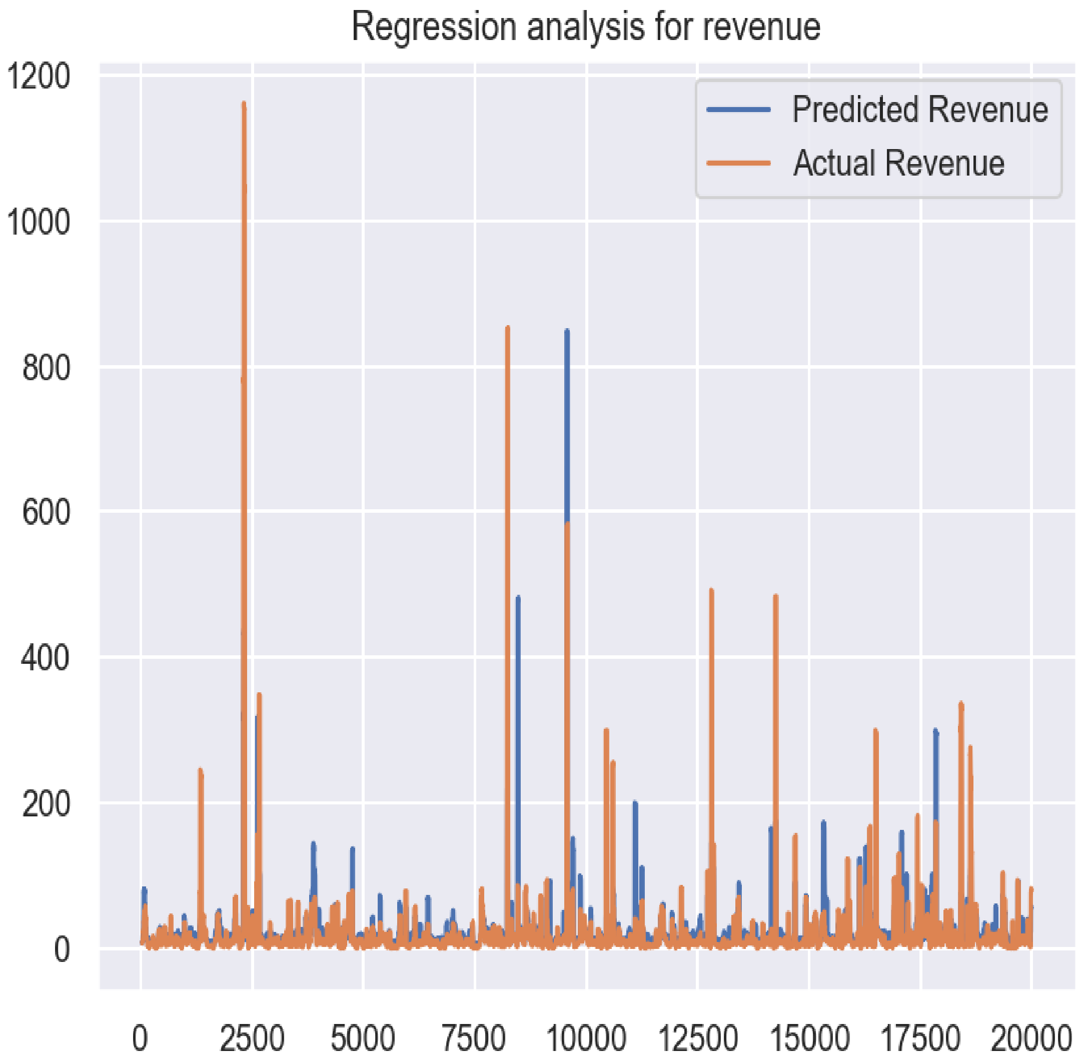
Key Insights

- Lift
- Confidence

```
rules[ (rules['lift'] >=5) &
       (rules['confidence'] >= 0.5) ] # Filtering desired outputs
```

- UK's recommended items brought together products are Wicker, Retro spot and wooden frames.

Building a predictive analytics model for revenue using linear regression.



Model Parameters

Dep. Variable:	final_revenue
Model:	OLS
Method:	Least Squares
R-squared:	0.447
Adj. R-squared:	0.447
F-statistic:	3234.
Prob (F-statistic):	0.00

Source: Southern Water Corp Statistical Records