Pepsico Data Challenge 2019

Presenter:
Pallavi Gupta
Ph.D Informatics Student
University of Missouri(Columbia)

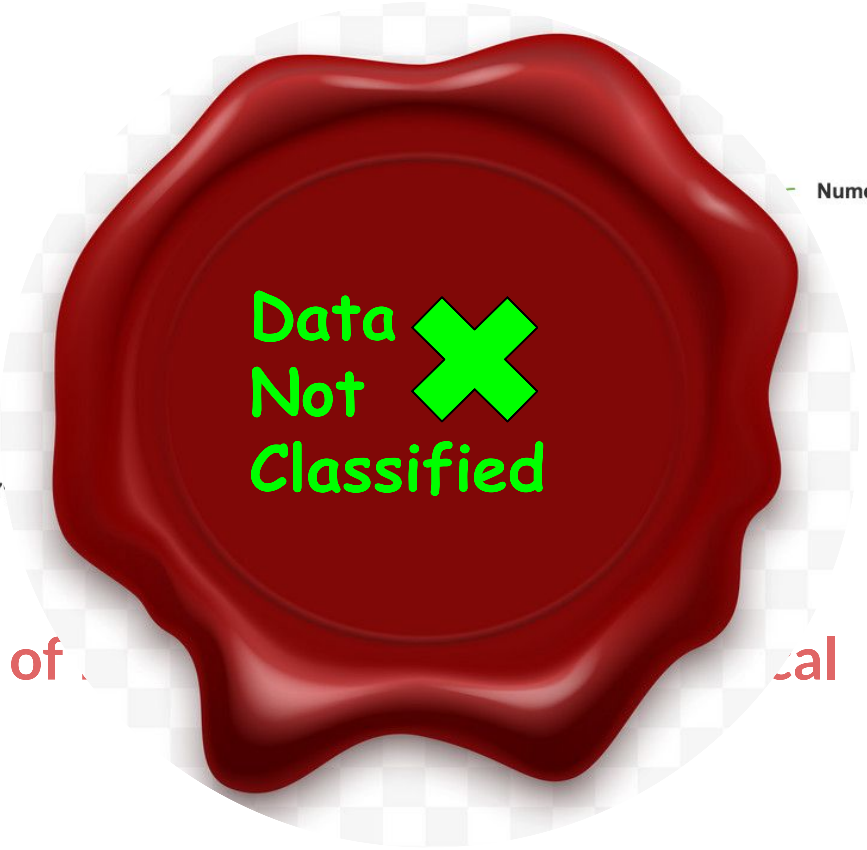
Define the success metric ???

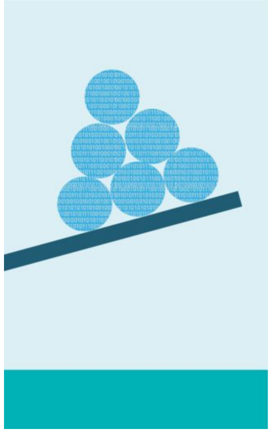
To build a model which is able to predict food products which have "difference from fresh taste" due to low shelf life.

Data explorations and challenges with data?

Numerical: 4 - 33.3%

Categorical: 8 - 66.7





8 out of

cal

Only 20.6

observed

te was

Data carpentry

- Binary categorization/classification of data points:
    - The target columns “difference from fresh” had continuous data.
    - Hence, for the difference in fresh value >=20, categorized as 1 (Different from fresh);
    - For difference in fresh value <20, categorized as 0 (No difference)
- Solution to categorical variables -
    - Binarised the “Yes” and “No” values to 0 and 1.
    - “One hot encoding” for columns having >2 values..
- Dealing with Missing values:
    - Which variables/columns/ features are worth keeping?
    - Dropping the rows was not a good idea either.

Dealing with the missing values :

Table: List of number of missing values in each column/variable in data.

Name of the columns (Variables)	Number of missing values
ProductType	0
BaseIngredient	109
ProcessType	0
SampleAge(Weeks)	0
StorageConditions	294
PackagingStabilizerAdded	282
TransparentWindowinPackage	647
ProcessingAgentStabilityIndex	0
PreservativeAdded	480
Moisture(%)	236
ResidualOxygen(%)	329
Hexanal(ppm)	460

Color	Code
Green	No missing values
Blue	Had <30% missing values
Red	Had **>30%** missing values

Dealing with the missing values:

- Kept the features/variables with 0 missing values.
  - Important features.
- Dropped the features/variables having >30% missing values.
  - Not important features.
- "Storage conditions" column had >30% missing values still this was one exception that I considered to retain.
- Features having <30% of missing values
  - Replaced the missing values with mode of all the values.
  - Averaging was not a good idea, especially in case of categorical variables.

Train and test split

- Data set was shuffled to avoid similar examples in one dataset
- Split with 33% test data and 67% train data.
- The test set was required to make sure the model was not overfitting/underfitting.

Model and its performance

- Random Forest classifier with maximum trees=50, max_depth of tree=16.
- Trained using the training data.
- Tested for the performance using test data.
- Stats below were obtained by running the trained model on the complete dataset.
- Accuracy in predictions:
    - Overall-> 88%
    - **For different from fresh - 68%**
    - For no difference - 93%

ACTUAL

PREDICTIONS

	Different from fresh	**No difference**
Different from fresh	True Positive (105)	False Positive (37)
No difference	False Negative (50)	True Negative (557)
Total Actual	105+50 = 155	37+557 = 594

=749

Conclusion

- Success metric - predict products that are different from fresh, correctly.
- 105/155(68%) predictions were correct which shows that the model was considerably successful.
- The success metric was not over weighted by the data imbalance.
- The analysis was done purely on statistical knowledge. However, collaborating and getting more knowledge on products would have allowed to make better predictions.
- As they say, there is always a room for improvement.