Customer Sentiment Prediction

Yuan Zhang

AGENDA

01 BACKGROUND

- The Raw Data Summary
- Goal and KPI
- The Baseline Model
- Challenges

02 Result Summary

- The Preferred Model and Feature Selection Insights
- Interactive Streamlit Dashboard

03 Technical Summary

- CV strategy and Metric
- Pipelines and Models

Background: The Raw Data Summary

- Y := target attribute (Y) with values indicating 0 (unhappy) and 1 (happy) customers
- X1 := my order was delivered on time
- X2 := contents of my order was as I expected
- X3 := I ordered everything I wanted to order
- X4 := I paid a good price for my order
- X5 := I am satisfied with my courier
- X6 := the app makes ordering easy for me

Attributes X1 to X6 indicate the responses for each question and have values from 1 to 5 where the smaller number indicates less and the higher number indicates more towards the answer.

Data summary: There are total 126 data points, 69 of which have Y=1 (happy customer). 69/126 is roughly 0.5476, indicating that this data set, as a whole, is quite balanced.

Background: Goal and KPI

- Goal: Predict if a customer is happy or not based on the answers they give to questions asked.
- KPI: The official KPI is the Accuracy Score. The indicated goal is "Reach 73% accuracy score or above".

Background: The Baseline Model

We will use the "assume everyone is happy" (trivial) model as the baseline model. Which, by construction, has an accuracy score of 0.5476.

Background: Challenges

- The low amount of data prevents of from using more complicated models that are better at understanding while take longer to learn. Thus, we will focus on "simpler" models.
- The low amount of data also makes it so that any "one split" train and test split can be "unlucky" or "lucky", making it unreliable to evaluate our model. This causes our decision on CV strategy.

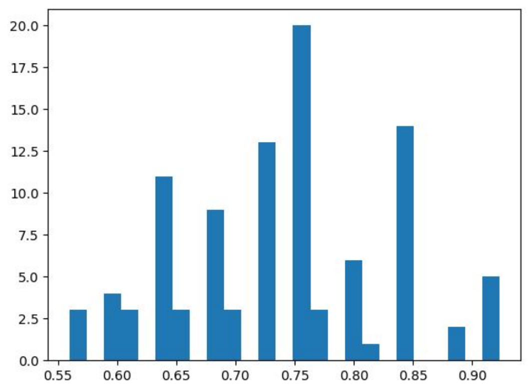
Result Summary: The Preferred Model and Feature Selection Insights

KNN with 5 neighbors:

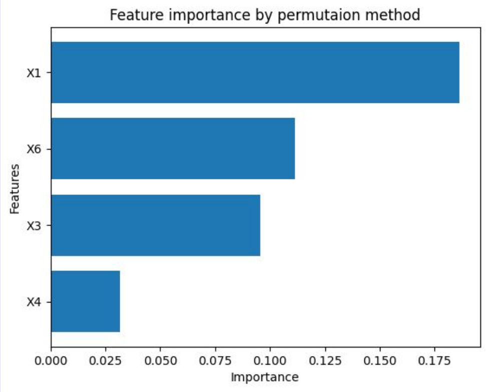
- Using raw features X1, X3, X4, and X6.
- Force Data Selector to only use X1, X6, and F score weighted mean.

Performance of the preferred model:

- This model produced average accuracy of 73.76%.
- Improving upon the trivial baseline model by 34.70%.
- By Applying this model repeatedly, we have 80.48% confidence that the average accuracy is at least 73%.



Histogram of distribution of the accuracy scores



Feature importance analysis

- X1 := my order was delivered on time
- X3 := I ordered everything I wanted to order
- X4 := I paid a good price for my order
- X6 := the app makes ordering easy for me

We will discuss technical details later.

Result Summary: Interactive Streamlit Dashboard

Please See Demonstration.

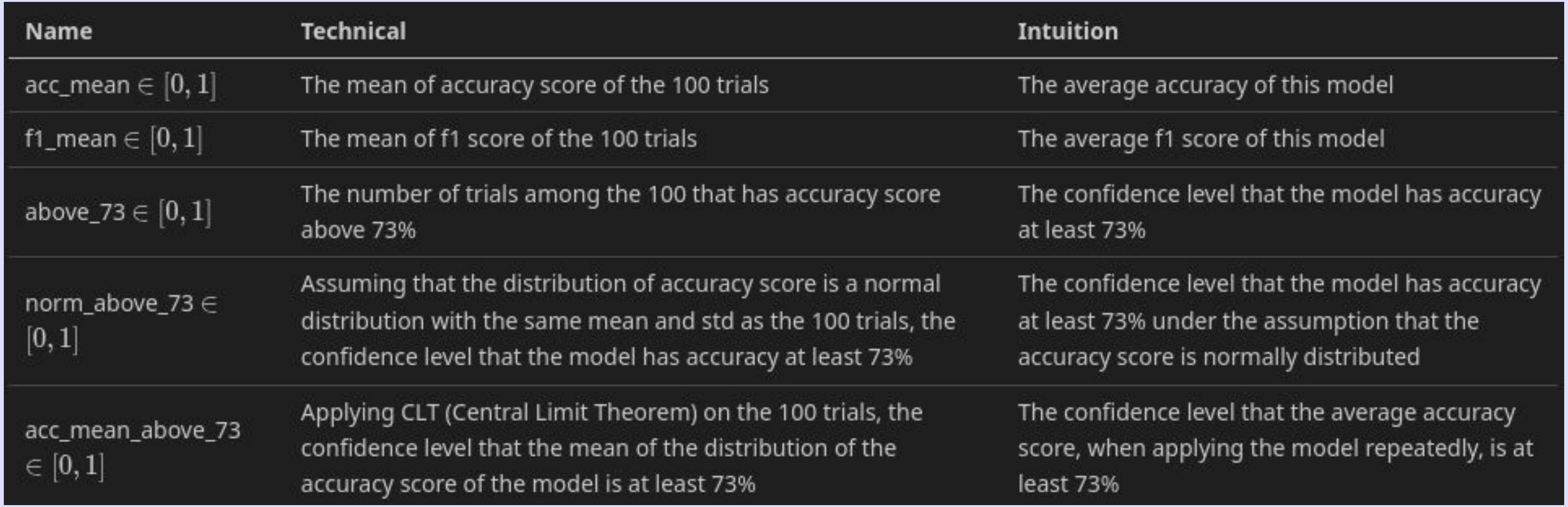
This concludes the non-technical part of the demonstration, thank you for your time.

If you are also interested in the more technical details, please continue watching.

Technical Summary: CV Strategy and Metric

We will take advantage of the pre-built repeated stratified kfold method of sklearn package. We will universally use 5 splits and 20 repeats, that is 100 trials in total for each model (with certain hyper-parameters)(so that the CV concludes before I concluded being alive).

This brings us to the metric(s) we will use to judge our model, we will describe them in both technical and intuitive context (The intuitive description might not be fully rigorous):



Name	Technical	Intuition
acc_mean $\in [0, 1]$	The mean of accuracy score of the 100 trials	The average accuracy of this model
f1_mean $\in [0, 1]$	The mean of f1 score of the 100 trials	The average f1 score of this model
above_73 $\in [0, 1]$	The number of trials among the 100 that has accuracy score above 73%	The confidence level that the model has accuracy at least 73%
norm_above_73 $\in [0, 1]$	Assuming that the distribution of accuracy score is a normal distribution with the same mean and std as the 100 trials, the confidence level that the model has accuracy at least 73%	The confidence level that the model has accuracy at least 73% under the assumption that the accuracy score is normally distributed
acc_mean_above_73 $\in [0, 1]$	Applying CLT (Central Limit Theorem) on the 100 trials, the confidence level that the mean of the distribution of the accuracy score of the model is at least 73%	The confidence level that the average accuracy score, when applying the model repeatedly, is at least 73%

Technical Summary: Pipelines and Models (Part 1: The Summary)

- **Models:**
  I have applied *Log Regression* (*LogReg*), *(Gaussian, Multinomial, Complement, and Categorical)-Naive Bayesian* (*NB*), *Decision Tree* (*DT*), *Support Vector Classifier* (*SVC*), *and K Nearest Neighborhood* (*KNN*) *models* (as the main steps of a pipelines).
- **A Standard Model Pipeline:**
  I will be using the Built-in Pipeline function of sklearn. There are two custom layers that perform data transforming:

  Data Creator: Manufactures new features with the raw features provided.

  Data Selector: Selects features either automatically by applying pre-set conditions, or simply select with a pre-set list: When done automatically, the layer ranks the features by F test (by sklearn f_classif), and select according to pre-set conditions on F scores and p values.

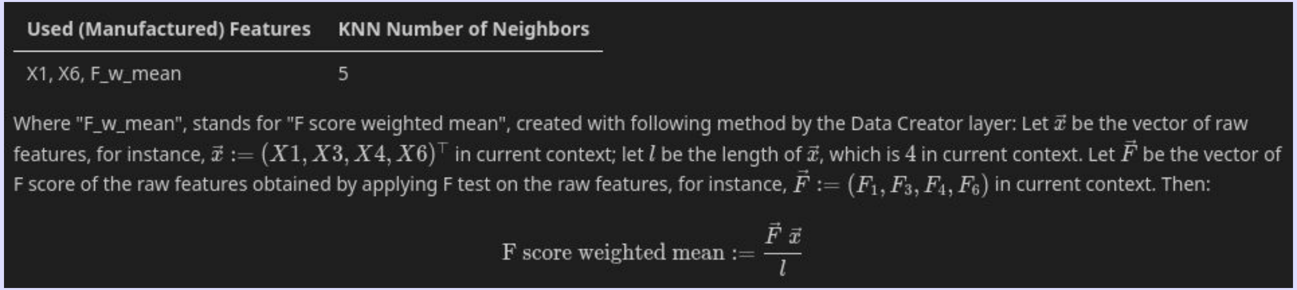
  After above data transformation, we load it to the main model layers.

Technical Summary: Pipelines and Models (Part 2: An example with the preferred model)

- Pipeline:

  Data Creator —> Data Selector —> StandardScaler —> KNN

- Raw features used: X1, X3, X4, X6
- Hyper-parameter table:

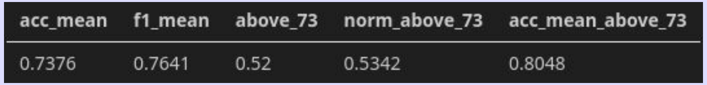


Used (Manufactured) Features	KNN Number of Neighbors
X1, X6, F_w_mean	5

Where "F_w_mean", stands for "F score weighted mean", created with following method by the Data Creator layer: Let $\vec{x}$ be the vector of raw features, for instance, $\vec{x} := (X1, X3, X4, X6)^\top$ in current context; let l be the length of $\vec{x}$, which is 4 in current context. Let $\vec{F}$ be the vector of F score of the raw features obtained by applying F test on the raw features, for instance, $\vec{F} := (F_1, F_3, F_4, F_6)$ in current context. Then:

$$\text{F score weighted mean} := \frac{\vec{F}\,\vec{x}}{l}$$

- Performance Table:



acc_mean	f1_mean	above_73	norm_above_73	acc_mean_above_73
0.7376	0.7641	0.52	0.5342	0.8048

Thank you for your time.