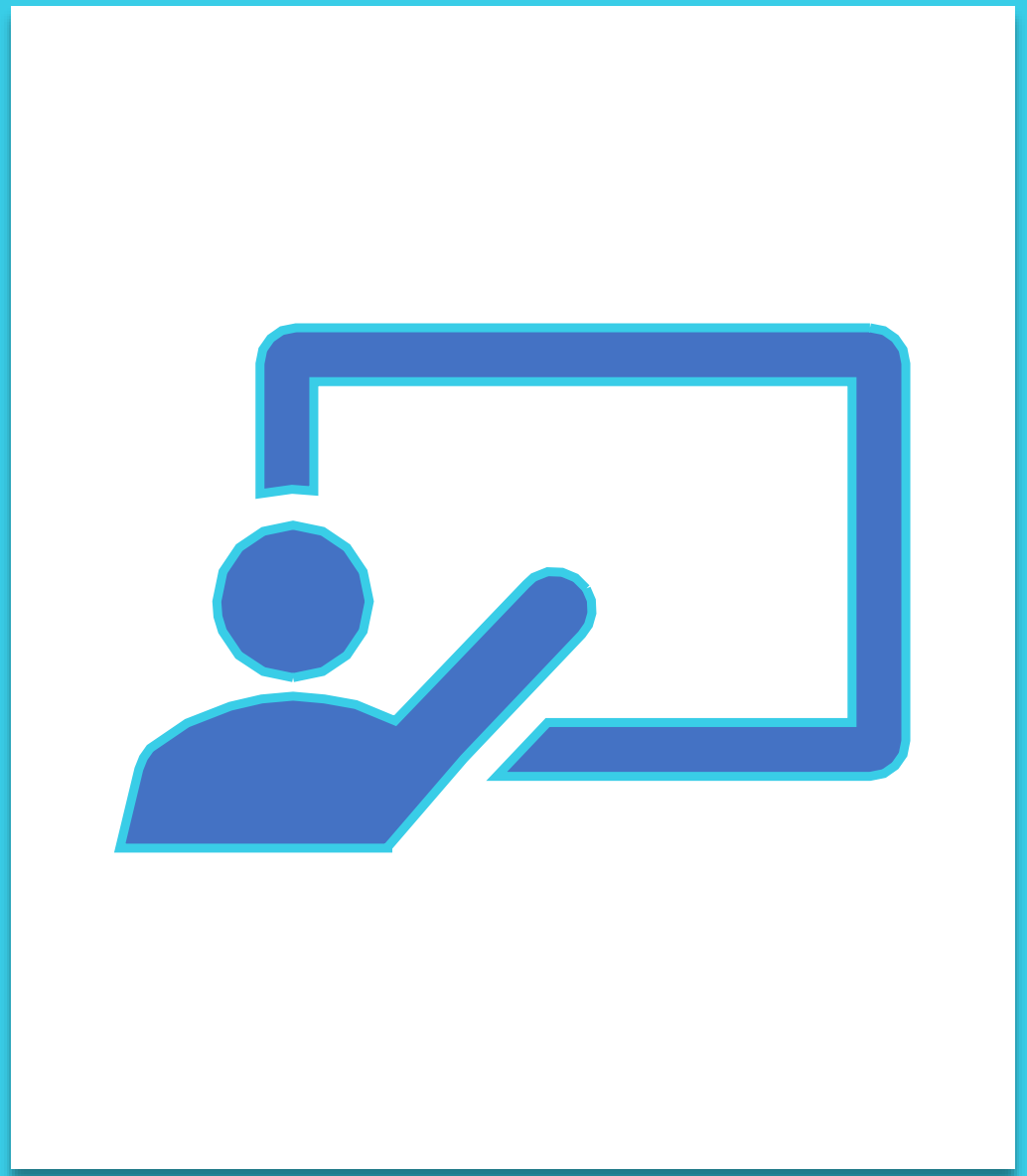
Predicting Next Clothing selection

Capstone Project 4 Presentation

Sarifah bte Sapuan



Topics

Introduction and Objective

Methodology

Process Workflow

Results

Conclusion

Introduction
Objective

I am student with the Data Science faculty

Task to revisit a prior research subject and apply machine learning

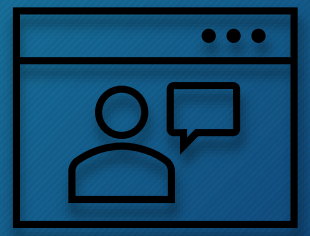
To predict the type of clothing category a customer will consider given a set of a parameters

Recommend product or category

Methodology

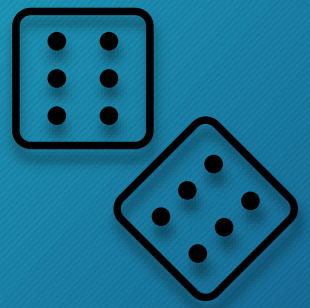
- Dataset : **Clickstream Data for Online Shopping**
- https://archive.ics.uci.edu/ml/datasets/clickstream+data+for+online+shopping#

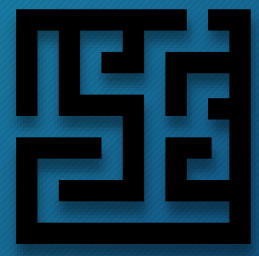
- Models
  - Decision Trees
  - Naïve Bayes
  - Logistic Regression



Methodology

- Metrics
  - Precision
  - Recall
  - F1-score
  - Support





Methodology

- Tools
  - Scikit-Learn
  - pandas
  - matplotlib
  - seaborn

Process Workflow

EDA & Data Preprocessing

Excel

- Added a new column – Next_choice
- Input is the clothing category from the next row
- Used as the target

Python

- Removed unwanted columns
- Converted categorical data to numerical data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 165474 entries, 0 to 165473
Data columns (total 17 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   year            165474 non-null  int64
 1   month           165474 non-null  int64
 2   day             165474 non-null  int64
 3   Date            165474 non-null  object
 4   Category_desc   165474 non-null  object
 5   ModelID         165474 non-null  object
 6   Colour_Desc     165474 non-null  object
 7   Next_choice     165474 non-null  object
 8   price           165474 non-null  int64
 9   Price2_desc     165474 non-null  object
 10  Region          165474 non-null  object
 11  Ctry_Name       165474 non-null  object
 12  SessionID       165474 non-null  int64
 13  Click_sequence  165474 non-null  int64
 14  Pageno          165474 non-null  int64
 15  location_desc   165474 non-null  object
 16  Model_desc      165474 non-null  object
dtypes: int64(7), object(10)
memory usage: 21.5+ MB
```

Process Workflow

Data preparation and Analysis

- columns
- content
- remove irrelevant/outlier data

```
def preprocess(df):

    # Drop features which are not important the model
    #
    df = df.drop(["SessionID", "Click_sequence", "Category_desc", "Colour_Desc", "Ctry_Name", \
                  "Pageno", "Date", "location_desc", "Model_desc","year"], axis=1)
    col_name="Next_choice"
    first_col = df.pop(col_name)
    df.insert(0, col_name, first_col)
    df = df[df['Next_choice'] != '0']
    class_names = df['Next_choice'].unique()


    # Drop all rows which have NaN values
    df = df.dropna()

    #Converting categorical to numeric values
    categorical_features=["ModelID","Region"]
    df = pd.get_dummies(df, columns = categorical_features)
    df['Price2_desc'] = df['Price2_desc'].apply(lambda x: 1 if x == 'Yes' else 0)
    df['Target'] = df['Next_choice']

    from sklearn.preprocessing import LabelEncoder
    le = LabelEncoder()
    df['Target'] = le.fit_transform(df['Target'])

    return df
```

```
df = preprocess(df)
df
```

Process Workflow

Data preparation and Analysis

- columns
- content

- remove irrelevant/outlier data

Input

```
month              5
day               31
price             20
Price2_desc        2
ModelID_A1         2
                  ..
ModelID_P9         2
Region_EU          2
Region_Non-EU      2
Region_Others      2
Region_Poland      2
Length: 225, dtype: int64
```

Output

```
trousers    49741
sale        38747
blouses     38577
skirts      38408
Name: Next_choice, dtype: int64
```

```
3    49741
1    38747
0    38577
2    38408
Name: Target, dtype: int64
```

Process Workflow

Decision Trees (Baseline)

- Split the data and train the model
  - DecisionTreeClassifier
  - GridSearchCV

- Predict and Evaluate the model
  - classification_report
  - confusion_matrix

```
X = df.drop(['Next_choice','Target'], axis = 1)# input
y = df['Target']  # output (dependent variable)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X ,y,
                                                    test_size = 0.2,
                                                    shuffle=True,
                                                    stratify=y,
                                                    random_state=SEED)
```

```
Classification report:

              precision    recall  f1-score   support

    trousers       0.71      0.71      0.71      7715
      skirts       0.75      0.75      0.75      7750
     blouses       0.71      0.71      0.71      7682
        sale       0.75      0.74      0.75      9948

    accuracy                           0.73     33095
   macro avg       0.73      0.73      0.73     33095
weighted avg       0.73      0.73      0.73     33095


Confusion Matrix:

array([[5471,  544,  812,  888],
       [ 744, 5793,  549,  664],
       [ 645,  630, 5438,  969],
       [ 896,  761,  884, 7407]], dtype=int64)
```

Process Workflow

Naive Bayes

- Split the data and train the model
  - MultinomialNB
  - cross_val_score
  - cross_validate
  - GridSearchCV
  - RandomizedSearchCV

- Predict and Evaluate the model
  - classification_report
  - confusion_matrix

```
X = df.drop(['Next_choice','Target'], axis = 1)# input
y = df['Target']  # output (dependent variable)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X ,y,
                                                    test_size = 0.2,
                                                    shuffle=True,
                                                    stratify=y,
                                                    random_state=SEED)
```

```
Classification report:

              precision    recall  f1-score   support

    trousers       0.71      0.71      0.71      7715
       skirts       0.73      0.74      0.74      7750
     blouses       0.69      0.69      0.69      7682
        sale       0.75      0.75      0.75      9948

    accuracy                           0.72     33095
   macro avg       0.72      0.72      0.72     33095
weighted avg       0.72      0.72      0.72     33095


Confusion Matrix:

array([[5480,  528,  865,  842],
       [ 738, 5720,  650,  642],
       [ 633,  797, 5301,  951],
       [ 861,  759,  885, 7443]], dtype=int64)
```

Process Workflow

Logistic Regression

- Split the data and train the model
  - LogisticRegression

- Predict and Evaluate the model
  - classification_report
  - confusion_matrix

```
X = df.drop(['Next_choice','Target'], axis = 1)# input
y = df['Target']  # output (dependent variable)
```

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X ,y,
                                                    test_size = 0.2,
                                                    shuffle=True,
                                                    stratify=y,
                                                    random_state=SEED)
```

```
Classification report:

              precision    recall  f1-score   support

    trousers       0.71      0.72      0.72      7715
      skirts       0.76      0.75      0.76      7750
     blouses       0.71      0.72      0.72      7682
        sale       0.75      0.75      0.75      9948

    accuracy                           0.74     33095
   macro avg       0.74      0.73      0.74     33095
weighted avg       0.74      0.74      0.74     33095


Confusion Matrix:

array([[5534,  496,  814,  871],
       [ 754, 5841,  525,  630],
       [ 599,  616, 5499,  968],
       [ 868,  736,  853, 7491]], dtype=int64)
```

Results

The recall means "how many of this class you find over the whole number of element of this class"

The precision will be "how many are correctly classified among that class"

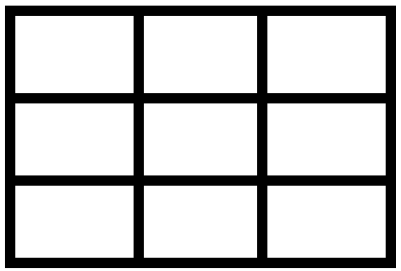
The f1-score is the harmonic mean between precision & recall

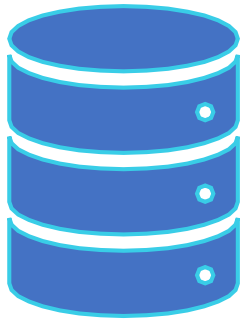
The support is the number of occurrences of the given class in your dataset

Results

Logistic Regression
- best model

Accuracy
F1-score = 0.74





Conclusion

To predict the type of clothing category a customer will consider given a set of a parameters

- 74% accuracy
- clothing category
- Possible to include as an active sales strategy

Future considerations

- Collection of a more current dataset with more meaningful features
- Other ML models

Appendix

SOURCE Citation:

Łapczyński M., Białowąs S. (2013) Discovering Patterns of Users' Behaviour in an E-shop - Comparison of Consumer Buying Behaviours in Poland and Other European Countries

Clickstream Data for Online Shopping

https://archive.ics.uci.edu/ml/datasets/clickstream+data+for+online+shopping#

Q & A