Универзитет Св. Кирил и Методиј – Скопје

Факултет за информатички науки и компјутерско инженерство

Анализа на податочно множество

E-commerce Customer Behavior Dataset

Линк до репозиториум | Линк до видео

Ментор:

проф. д-р Димитар Трајанов

Студент:

Јована Трајческа 201089

Содржина

Опис

Во продолжение на овој документ се разгледува детална анализа на податочно множество кое опфаќа информации за однесувањето на купувачите и pattern-и при купување и тоа првенствено од ecommerce платформите и платформите за малопродажба. Вклучува карактеристики за однесувањето на корисниците со страницата, односно како тече интеракцијата со сајтот, нивните навики при купување и некои демографски карактеристики. Генерално, ова добро ги опишува карактеристиките кои понатаму ќе ги користиме за анализа на податочното множество и носење на заклучоците. Податоците се генерирани за образовни цели од Gretel AI.

Во продолжение, ќе направиме краток опис на карактеристиките достапни во множеството:

- **Customer ID** – уникатен идентификатор (познат како примарен клуч во областа на базите на податоци). Од суштинско значење е за следење на индивидуалното однесување на корисниците низ различни карактеристики во податочното множество.
- **Age** – возраст на купувачите
- **Gender** – пол на купувачите.
- **Location** – географска местоположба на купувачите
- **Annual Income** – годишен приход на клиентот
- **Purchase History** – листа на продукти кои ги купил купувачот
- **Browsing History** – листа на продукти кои ги гледал купувачот заедно со временската рамка при истото
- **Product Reviews** – критика за производи заедно со рејтинг од 1 до 5 ѕвезди
- **Time on Site –** вкупното време кое клиентот го поминал на сајтот

Преземање и читање на податоците

За изработка на оваа анализа се користи податочно множество од Kaggle. По зачувување на множеството локално, за читање на истото се користи pandas што претставува моќна библиотека за читање, манипулација и анализа на податоци во Python. Нуди структури на податоци како што се Series и DataFrame.

Библиотеката се инсталира преку следната команда:

- **pip install pandas**

А, се импортира со:

- **import pandas as pd**

Забелешка: pd може да биде и нешто друго, како на пример pandas (import pandas as pandas), но тоа значи дека наместо читањето да го правиме со pd, ќе користиме pandas.

Во кодот, ја користиме уште на самиот почеток при читање на податочното множество. Истото е прикажано и на сликата подолу.

```
import pandas as pd

data = pd.read_csv("./E-commerce.csv")
✓ 0.0s
```

Слика 1. Читање на податочното множество

Во овој случај, на data вчитуваме податочно множество кое се наоѓа во истиот директориум како и самиот документ во кој се работи.

Со извршување на **data.head()** добиваме претстава за какво податочно множество се работи, какви вредности имаме, но не ги прикажува сите податоци, туку само неколку.

```
data.head()
✓ 0.0s
```

	Customer ID	Age	Gender	Location	Annual Income	Purchase History	Browsing History	Product Reviews	Time on Site
0	1001	25	Female	City D	45000	[{"Date": "2022-03-05", "Category": "Clothing"...	[{"Timestamp": "2022-03-10T14:30:00Z"}, {"Time...	Great pair of jeans, very comfortable. Rating:...	32.50
1	1001	28	Female	City D	52000	[{"Product Category": "Clothing", "Purchase Da...	[{"Product Category": "Home & Garden", "Timest...	Great customer service!	123.45
2	1001	28	Female	City D	65000	[{"Product Category": "Electronics", "Purchase...	[{"Product Category": "Clothing", "Timestamp":...	Great electronics. The sound quality is excell...	125.60
3	1001	45	Female	City D	70000	{'Purchase Date': '2022-08-15', 'Product Categ...	{'Timestamp': '2022-09-03 14:30:00'}	{"Product 1": {"Rating": 4, "Review": "Great e...	327.60
4	1002	34	Male	City E	45000	{'Purchase Date': '2022-07-25', 'Product Categ...	{'Timestamp': '2022-08-10 17:15:00'}	{"Product 1": {"Rating": 3, "Review": "Good pr...	214.90

Слика 2. Команда – head

Проверка за Missing Values

Првиот начин да се провери дали во податочното множество ни фалат податоци или не, е преку извршување на командата **data.isnull().** Ова ќе ни ја врати табелата, но во овој случај истата ќе содржи content True/False и тоа во зависност дали во соодветната ќелија во табелата имаме missing value или пак немаме. True значи дека имаме, False дека немаме.

```
# checking for missing values

data.isnull()
```

✓ 0.0s

	Customer ID	Age	Gender	Location	Annual Income	Purchase History	Browsing History	Product Reviews	Time on Site
0	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False
5	False	False	False	False	False	False	False	False	False
6	False	False	False	False	False	False	False	False	False
7	False	False	False	False	False	False	False	False	False
8	False	False	False	False	False	False	False	False	False
9	False	False	False	False	False	False	False	False	False
10	False	False	False	False	False	False	False	False	False
11	False	False	False	False	False	False	False	False	False
12	False	False	False	False	False	False	False	False	False
13	False	False	False	False	False	False	False	False	False
14	False	False	False	False	False	False	False	False	False
15	False	False	False	False	False	False	False	False	False
16	False	False	False	False	False	False	False	False	False
17	False	False	False	False	False	False	False	False	False
18	False	False	False	False	False	False	False	False	False
19	False	False	False	False	False	False	False	False	False
20	False	False	False	False	False	False	False	False	False
21	False	False	False	False	False	False	False	False	False
22	False	False	False	False	False	False	False	False	False
23	False	False	False	False	False	False	False	False	False
24	False	False	False	False	False	False	False	False	False
25	False	False	False	False	False	False	False	False	False
26	False	False	False	False	False	False	False	False	False
27	False	False	False	False	False	False	False	False	False
28	False	False	False	False	False	False	False	False	False
29	False	False	False	False	False	False	False	False	False
30	False	False	False	False	False	False	False	False	False

Слика 3. Приказ на резултат по извршена команда

Со командата **data.isnull().sum()** ќе добиеме нешто слично, но наместо табела, во овој случај би добиле „мини табела“ која од лева страна го содржи името на колоната, а од десно бројот на вредности кои недостигаат.

Во овој случај немаме вредности кои ни фалат, па не се прави чистење на податоците.

```
data.isnull().sum()
```

✓ 0.0s

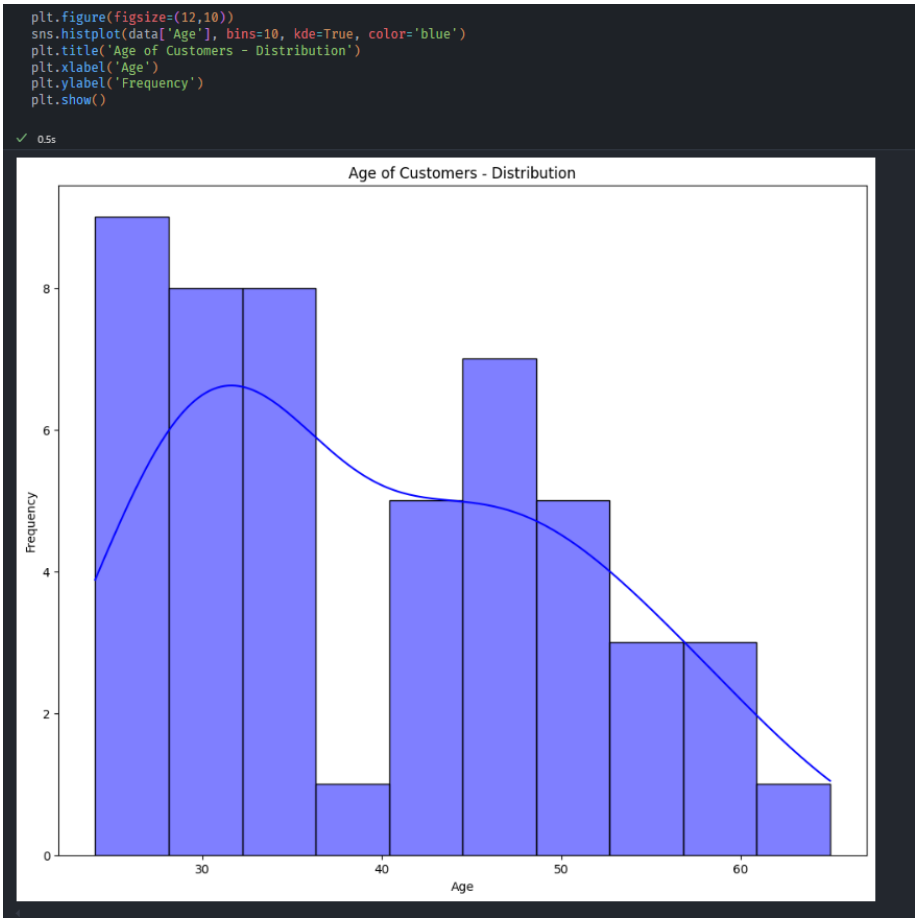
```
Customer ID          0
Age                  0
Gender               0
Location             0
Annual Income        0
Purchase History     0
Browsing History     0
Product Reviews      0
Time on Site         0
Cleaned_Reviews      0
Sentiment_Textblob   0
bert_sentiment       0
dtype: int64
```

Слика 4. Втора команда за проверка на missing values

Data Visualization

Можеме да направиме различни видови на анализи и визуелизации според карактеристиките во податочното множество.

Во продолжение ќе разгледаме дел од визуелизациите кои се корисни за анализа и дел од нив ќе бидат накратко објаснети.



Слика 5. Дистрибуција според возраста на купувачите

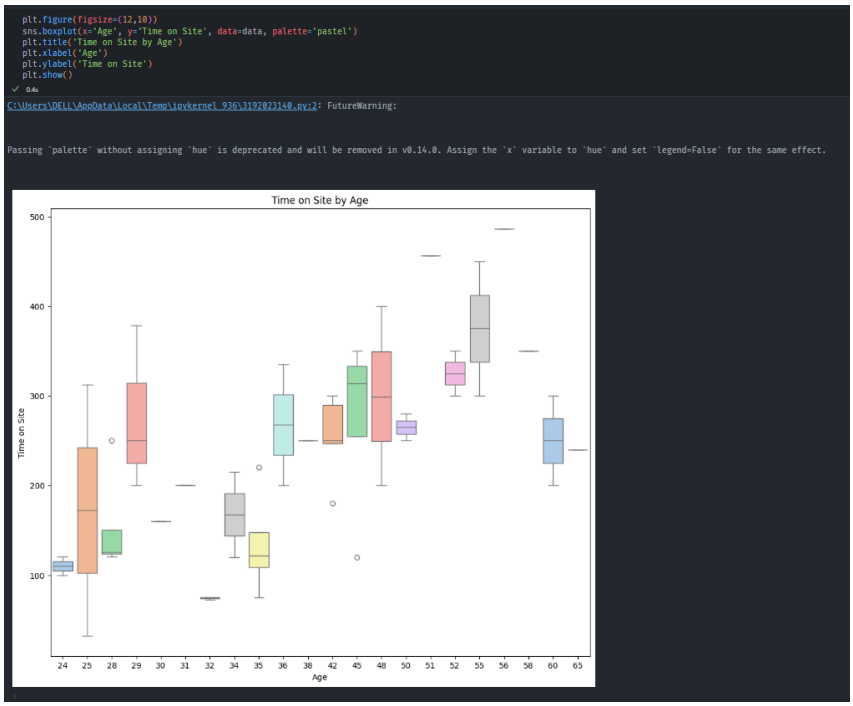
Освен по возраст, направени се визуелизации и за пол и годишен приход и може да се видат детално во кодот.

- ✓ **plt.figure(figsize=(12,10))** – креира график со специфични димензии од 12 инчи ширина и 10 инчи висина. plt е кратенка за библиотеката Matplotlib и ја користиме за визуелизација на податоци.
- ✓ **sns.histplot(data['Age'], bins=10, kde=True, color='blue')** – Seaborn библиотека за создавање на хистограм. Податоците се поделени на 10 интервали.
- ✓ **plt.show()** – користиме за всушност да ја прикажеме визуелизацијата

Слика 6. Годишен приход според пол

Boxplot во случајов ефективно ја сумира распределбата на годишниот приход според пол истакнувајќи ја медијаната, квартилите, како и јасен приказ при постоење на outliers.

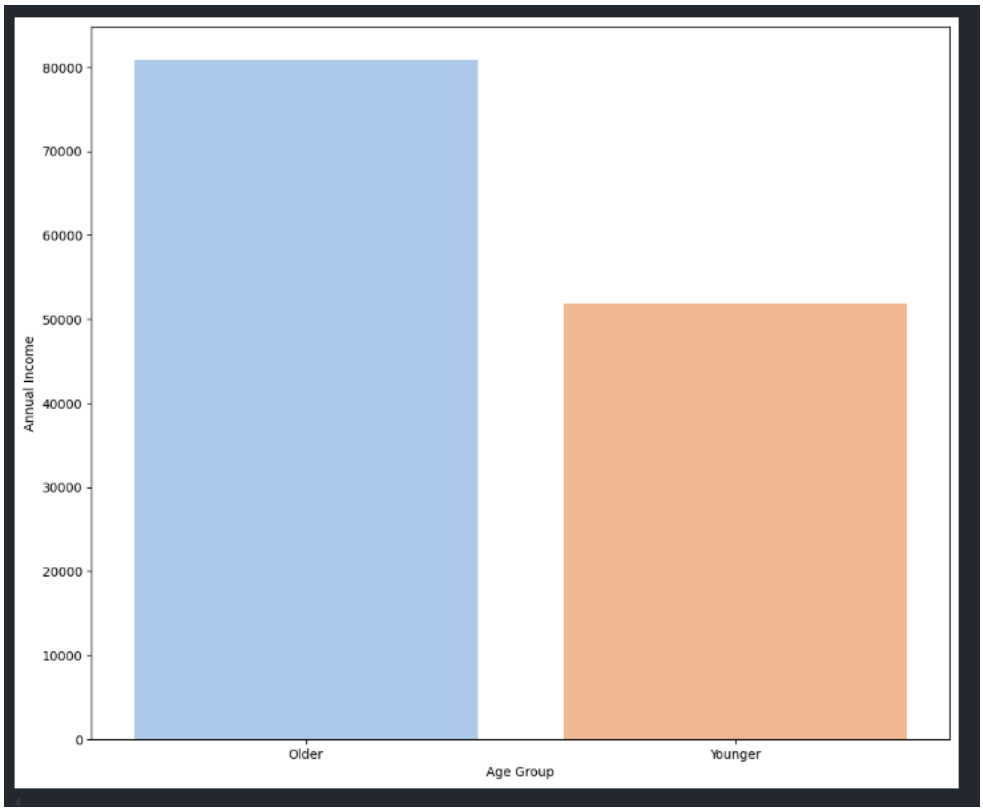
Слично на ова погоре, визуелизација за времето поминато на сајтот според возраст.



Слика 7. Време поминато на сајтот според возраст

Дополнително направени се и анализи за влијанието на критиките врз купувањето, дистрибуција на критики, корелација меѓу карактеристиките на податочното множество. Понатаму, има визуелизација и за тоа каква интеракција со сајтот има според локацијата на корисниците, визуелизација за времето кое го поминуваат според локацијата, поврзаноста помеѓу времето кое го поминуваат на сајтот, историјата на купување, годишниот приход и слично. Постојат и визуелизации за тоа како времето поминато на сајтот и историјата на пребарување се поврзани и слично. Во интерес на ограничувањето на бројот на страни, истово може да се прегледа во самиот код, каде се оставени резултатите при извршување на кодот.

На следнава слика се прикажува распределба на годишниот приход во две групи според возраст. Направена е поделба на купувачите на постари од просекот и помлади од просекот. Исто така направени се и пресметки за медијаната и просекот на нивниот годишен приход.



Слика 8. Годишен приход според возрасна група

На следнава слика е прикажана бројката на купување во одреден временски интервал.

```
plt.figure(figsize=(10,6))
data.groupby('Purchase Date').size().plot(kind='line', color='purple')
plt.title('Purchases')
plt.xlabel('Date')
plt.ylabel('Purchases Counter')
plt.xticks(rotation=45)
plt.show()
```

✓ 0.3s

Слика 9. Бројач на купување според датум

Исто така, можеме да направиме и визуелизација според купување на продукти од одредена категорија и задржување на сајтот, или пак според категорија и годишен приход. На следнава слика може да се разгледа кодот за овие две визуелизации.

```
clothing_data_time = data[data['Purchase History'].str.contains('Clothing', na=False)]

sns.histplot(x='Purchase History', y='Time on Site', data=clothing_data_time)

clothing_data_income = data[data['Purchase History'].str.contains('Clothing', na=False)]

sns.histplot(x='Purchase History', y='Annual Income', data=clothing_data_income)
```

Слика 10. Визуелизација според категорија на продукт и приход или време поминато на сајтот

Sentiment Analysis

Најпрво се прави чистење на податоците со цел да се избегнат информации кои би влијаеле лошо на моделот. Избришани се интерпункциски знаци, броеви и некои зборови кои би влијаеле на резултатот. Истото е направено во различни функции и на крај се споени во една.

```python
def remove_punctuation(text):
    return text.translate(str.maketrans('', '', string.punctuation))


def remove_numbers(text):
    return re.sub(r'\d+', '', text)


def remove_unwanted_phrases(text):
    text = re.sub(r'product\s+rating\s+review', '', text, flags=re.IGNORECASE)
    text = re.sub(r'review\s+text', '', text, flags=re.IGNORECASE)
    return text


def clean_text(text):
    text = remove_punctuation(text)
    text = remove_numbers(text)
    text = remove_unwanted_phrases(text)
    text = text.lower()
    return text.strip()
```

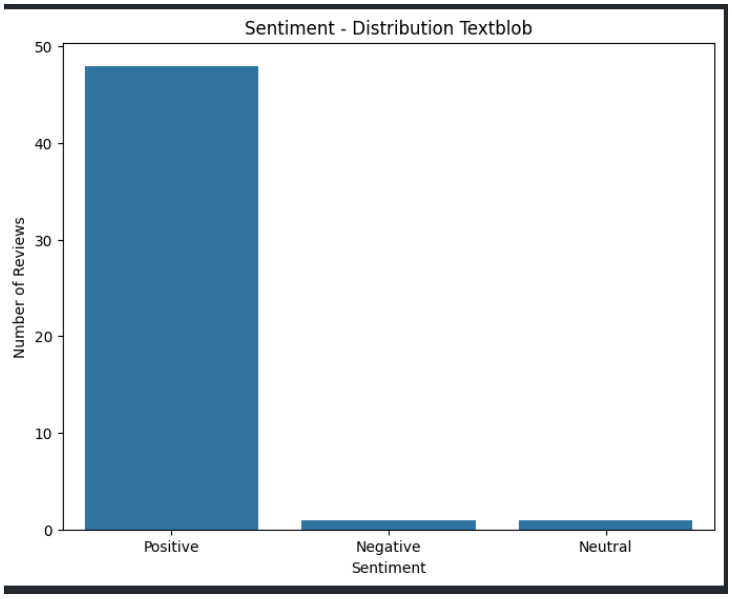
Слика 11. Cleaning Text

TextBlob е библиотека за процесирање на текстуални податоци и екстрахирање на карактеристики како sentiment. Дава polarity score од -1 до 1 и тоа:

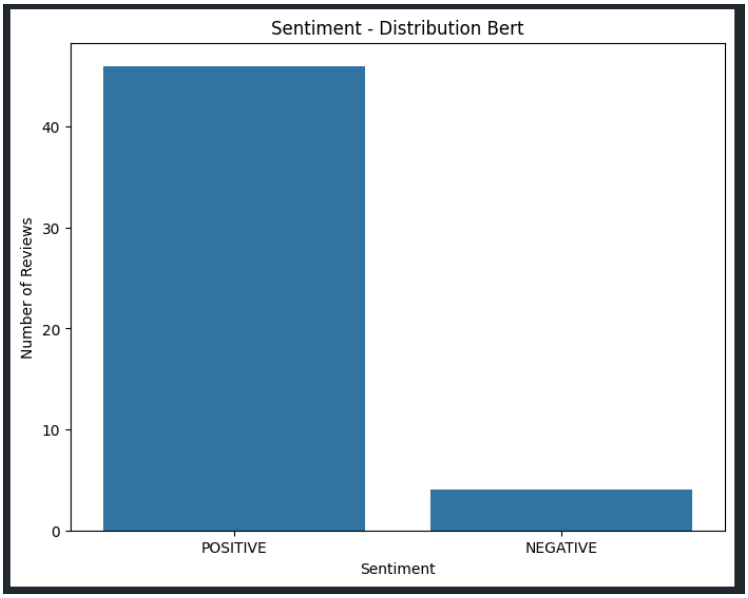
- (-1) – многу негативно
- (0) – неутрално
- (+1) – многу позитивно

Се користи уште една sentiment-analysis со користење на bert_sentiment.

Со TextBlob испитуваме поедноставни лексички sentiment, додека пак со вториот, всушност користиме трансформер кој го разбира целиот контекст. Резултатите кои се добиени и со двете се слични, па слободно може да кажеме дека критиките се едноставни, не се двосмилени, без сарказам се и слично.



Слика 12. Sentiment Analysis TextBlob



Слика 13. Sentiment Analysis Transformer (Bert)

Model Train

Најпрво со користење на LabelEncoder се енкодираат колоните Location и Gender бидејќи потоа се користат како features.

Податочното множество е поделено во размер 80% и 20%, тестирачко и тренирачко соодветно.

Како модели за предвидување избрав Random Forest, Decision Tree и XGBoost и на крај за истите направив споредба на добиените вредности по тренирањето. Најдобри вредности даде Random Forest со R-squared 0.52.

```
features = data[['Age', 'Gender', 'Time on Site', 'Location']]

target = data['Annual Income']
```
✓ 0.0s

```
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2, random_state=42)
```
✓ 0.0s

```
# RandomForest

rf_model = RandomForestRegressor(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
y_pred_rf = rf_model.predict(X_test)
```
✓ 0.2s

```
# DecisionTree

dt_model = DecisionTreeRegressor(random_state=42)
dt_model.fit(X_train, y_train)
y_pred_dt = dt_model.predict(X_test)
```
✓ 0.0s

```
#XGBoost

xgb_model = XGBRegressor(random_state=42)
xgb_model.fit(X_train, y_train)
y_pred_xgb = xgb_model.predict(X_test)
```
✓ 1.1s

Слика 14. Модели за предвидување