Prediction of user gender from browser Data

Sergio Cucinotta

9 settembre 2024

1 Introduction

In today's digital landscape, understanding user behaviors is crucial for businesses dealing with the delicate balance of personalization and privacy, especially with the prevalence of social media and e-commerce. My specific focus in this test is the prediction of user gender from browser usage patterns.

The dataset, though synthetic, mirrors real-world user data, comprising ground_truth_ml.csv with identifiers and genders, and variables_ml.csv with web content details. Subsequent sections will elaborate on methodologies, tools, and challenges, aiming for a solution that meets data analysis requirements.

2 Approach and Tools

My approach encompassed several stages:

- Merge the Datasets: ground_truth and variables were merged on the 'id' column to create a combined dataset.

```
import pandas as pd

ground_truth = pd.read_csv("C:\\Users\\Serj\\Documents\\
    Gender_Prediction\\ground_truth_ml.csv")
variables = pd.read_csv("C:\\Users\\Serj\\Documents\\
    Gender_Prediction\\variables_ml.csv")
combined_data = pd.merge(ground_truth, variables, on='id', how
    ='inner')
combined_data
```

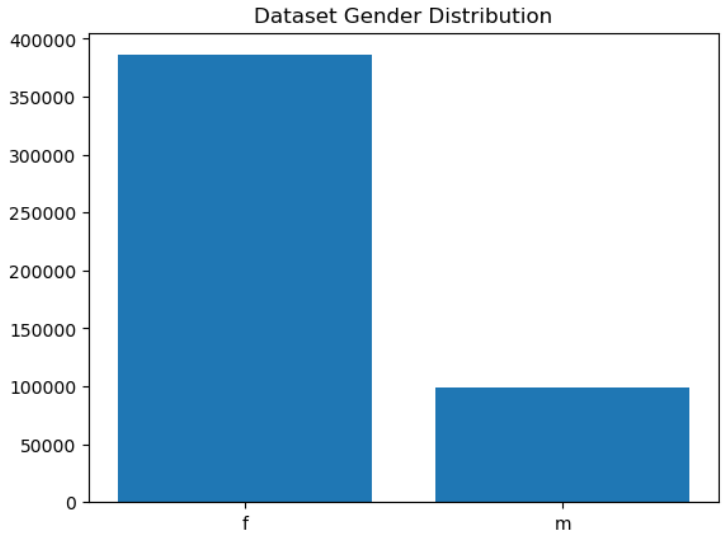
	id	gender	ds	h	content
0	9933667c-5b9c-4e80-a276-78aba991ceb6	f	1682380800	22	508
1	9933667c-5b9c-4e80-a276-78aba991ceb6	f	1681257600	21	38
2	9933667c-5b9c-4e80-a276-78aba991ceb6	f	1682380800	22	621
3	9933667c-5b9c-4e80-a276-78aba991ceb6	f	1681257600	22	508
4	9933667c-5b9c-4e80-a276-78aba991ceb6	f	1680912000	14	712
...	...	...	...	...	...
485293	64da0f76-3804-4b61-8b00-6b923f1c448d	f	1680912000	9	760
485294	e76d941e-5ada-4c94-9424-be59ed27930d	f	1680393600	20	712
485295	0fe56e26-3cea-4d12-b3cc-f9677c3db177	f	1681948800	18	21
485296	26131393-ae39-44f8-b9d7-0eca0a0e6cbe	f	1681257600	18	566
485297	0d7af2f9-2f55-447a-a874-937d49e5f80a	f	1680739200	8	712

485298 rows × 5 columns

- Dataset distribution: Show the distribution of 'm' and 'f' in the 'gender' column. The imbalance in the dataset was one of the hurdles in this project, with a significant prevalence of female gender.

```
import matplotlib.pyplot as plt

plt.bar(combined_data['gender'].unique(), combined_data['
    gender'].value_counts())
plt.title('Dataset Gender Distribution')
plt.show()
```



- Data Cleaning: Ensured data integrity by checking if any missing values are present. The result was 0 missing values.

```
print(combined_data.isnull().sum())
combined_data = combined_data.dropna()
```

- Feature Engineering: Converted the 'ds' column from datetime strings to integers representing Unix timestamps. This conversion was needed for further steps.

```
combined_data['ds'] = pd.to_datetime(combined_data['ds']).
    astype(int) /  10**9
```

- Encoding: Encodex of the categorical features 'content' with LabelEncoder. This transformation allows the model to process and learn from categorical variables effectively

```
from sklearn.preprocessing import LabelEncoder

encoder = LabelEncoder()
combined_data['content'] = encoder.fit_transform(combined_data
    ['content'])
```

- Split the Data: I utilized the train_test_split function from scikit-learn to divide the dataset into features (X) and the target variable (y). After defining X and y, I split them into training and test sets, allocating 80% for training and 20% for testing. This ensures that I can train the machine learning model on one subset and evaluate its performance on another, unseen subset. The random_state=42 parameter ensures reproducibility.

```
from sklearn.model_selection import train_test_split

X = combined_data.drop(['id', 'gender'], axis=1)
y = combined_data['gender']

X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=42)
```

- Model Building: I employed a RandomForestClassifier from scikit-learn, setting the random state for reproducibility. After fitting the model on the training data (X_train, y_train), I made predictions on the test set (X_test) and the accuracy score was calculated.

  I selected the Random Forest algorithm for its adaptability and its ability to manage both numerical and categorical data.

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

clf = RandomForestClassifier(random_state=42)

clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
```

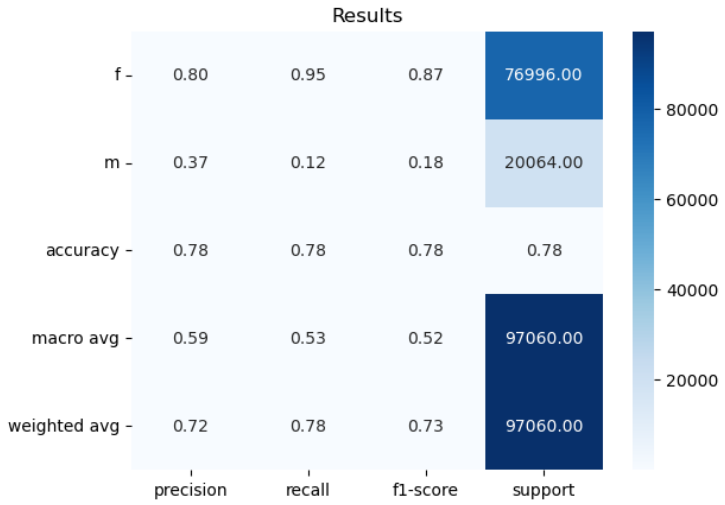
- Evaluation: Assessed the model using metrics like accuracy, precision, recall, and F1 score, visualizing the results through a heatmap for a comprehensive overview.

```python
from sklearn.metrics import classification_report
import seaborn as sns

report = classification_report(y_test, y_pred, output_dict=
    True)

report_df = pd.DataFrame(report).transpose()
sns.heatmap(report_df, annot=True, fmt=".2f", cmap="Blues")

plt.title('Results')
plt.show()
```



3 Conclusions and Utility of the Model

The model exhibits varying performance across gender classes. It demonstrates relatively high precision and recall for the female class (f), indicating accurate identification. However, its performance is notably weaker for the male class (m), with lower precision, recall, and F1-score. The overall accuracy of 78% suggests a reasonable classification performance, but the imbalanced class distribution impacts the model's ability to effectively predict the minority class.

Specifically, the model achieved a precision of 0.80 for females and 0.37 for males, a recall of 0.95 for females and 0.12 for males, and an F1-score of 0.87 for females and 0.18 for males.

These figures still represent a substantial improvement over a simplistic strategy of randomly assigning genders. This insight can be crucial for personalizing user experiences and ensuring compliance with privacy regulations.

4 Future Work

With more time and resources, I'd plan to explore alternative algorithms and conduct a thorough feature selection process. I would also intend to delve into ensemble methods and consider cost-sensitive training to enhance the model's performance even further.

Thank you for considering this report.