HACK FEST

PROBLEM STATEMENT:

CODE AUTHORSHIP IDENTIFICATION

TEAM:

CLUSTER-1

IC 5

TABLE OF CONTENTS

1.Problem Statement

CODE AUTHORSHIP IDENTIFICATION

Build a machine Learning model that can identify the author of a code change based on their coding style and identify any anomalies in their code compared to their usual coding patterns.

2.Abstract

Trends in data mining are increasing over the time. Current world is of internet and everything is available over internet, which leads to criminal and malicious activity. So the identity of available content is now a need. Available content is always in the form of text data. Authorship analysis is the statistical study of linguistic and computational characteristics of the written documents of individuals. This project describes review of method for authorship analysis and identification for a set of provided text. Surely research in authorship analysis and identification will continue and even increase over decades. In this article, we put our vision of future authorship analysis and identification with high performance and solution for behavioural feature extraction from set of text documents.

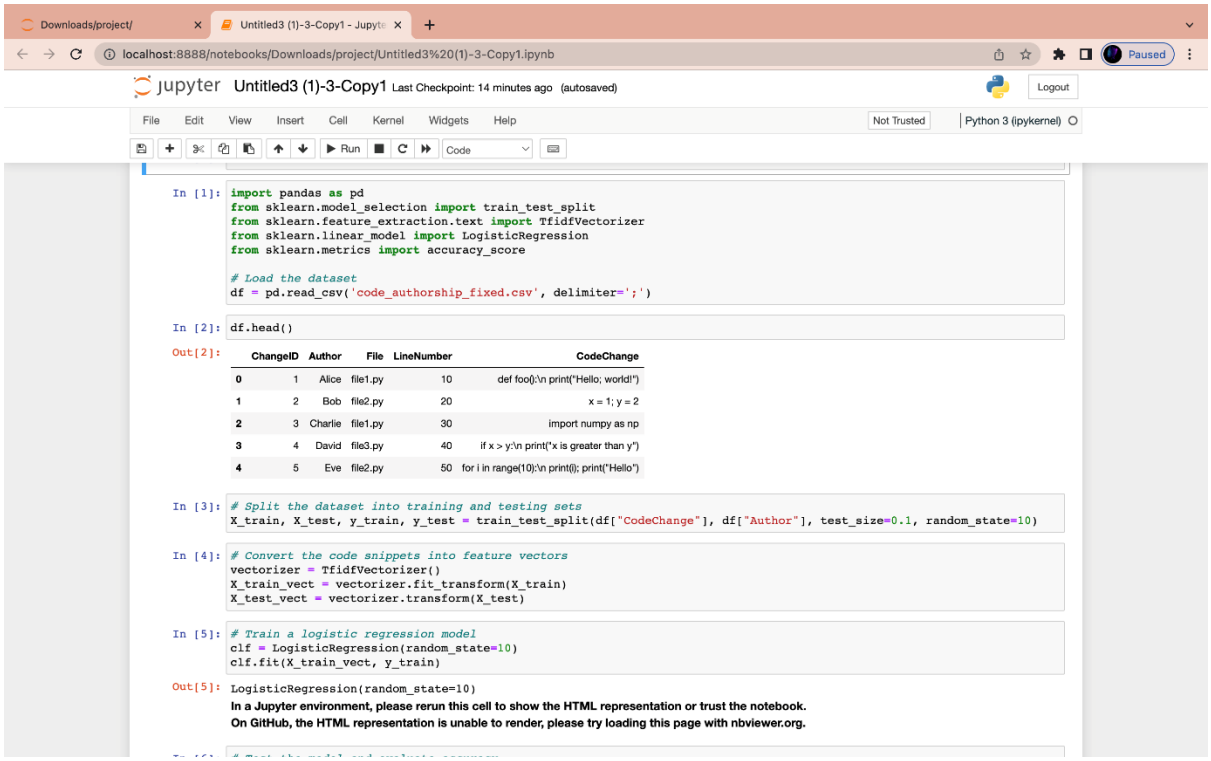
3.Introduction

Way to determine the authorship of handwritten document, and text document is a very old one. Now large volume of text is available in the form of digital content. Attribution for any text to a known ancient authority was essential to determining the text's veracity. This problem of author attribution is most important because of application in forensic analysis, humanities scholarship, electronic commerce, and the development of computational methods for addressing the problems. In recent years, the use of data mining techniques are increased for several purposes. Data available might be in any format like text, images, binary, and multimedia. And several techniques of mining increased, modified, improved over the time. Here we focus on author identification techniques. Even before the world of computer, this technique was in its way shows in work of Mendenhall (1887). Today the availability of text document in electronic form increases the importance of using automatic methods to analyze the content of text documents. Initially identifying document was very time consuming, expensive and has its limit. That emerges text categorization in predefined categories called as classification. Categorization is based on certain properties called as features. There are various method for extraction of features. Write print is one analogue to finger print method. Another n□gram features, will give information about increasing word sequence according to its length. Next is focus on stylometric features. It includes a set of the style markers which are adapted for the automatic analysis of the text. A Source Code (programming code) Author Profiles (SCAP) represents a new, highly accurate approach to source code authorship identification. Another section, text categorization based on keywords that may appear uniquely, may dual sequences like computer science, genetic algorithm etc. In discriminative syntactic tree approach, there is direct mining from a given set of syntactic trees.

4.Solution

- A Machine Learning Model which deals code and Author of the code as input and gives the output as yes, if the coding style of the author matches with the dataset and gives output as no, if the coding style of the author does not match with the database.
- Here, we used logistic regression to train and test the Model.
- The dataset contains author name, coding style of every individual author.

Source Code



Downloads/project/ | Untitled3 (1)-3-Copy1 - Jupyter

localhost:8888/notebooks/Downloads/project/Untitled3%20(1)-3-Copy1.ipynb

jupyter Untitled3 (1)-3-Copy1 Last Checkpoint: 14 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

```
In [1]: import pandas as pd
        from sklearn.model_selection import train_test_split
        from sklearn.feature_extraction.text import TfidfVectorizer
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import accuracy_score

        # Load the dataset
        df = pd.read_csv('code_authorship_fixed.csv', delimiter=';')
```

```
In [2]: df.head()
```

Out[2]:

	ChangeID	Author	File	LineNumber	CodeChange
0	1	Alice	file1.py	10	def foo():\n print("Hello; world!")
1	2	Bob	file2.py	20	x = 1; y = 2
2	3	Charlie	file1.py	30	import numpy as np
3	4	David	file3.py	40	if x > y:\n print("x is greater than y")
4	5	Eve	file2.py	50	for i in range(10):\n print(i); print("Hello")

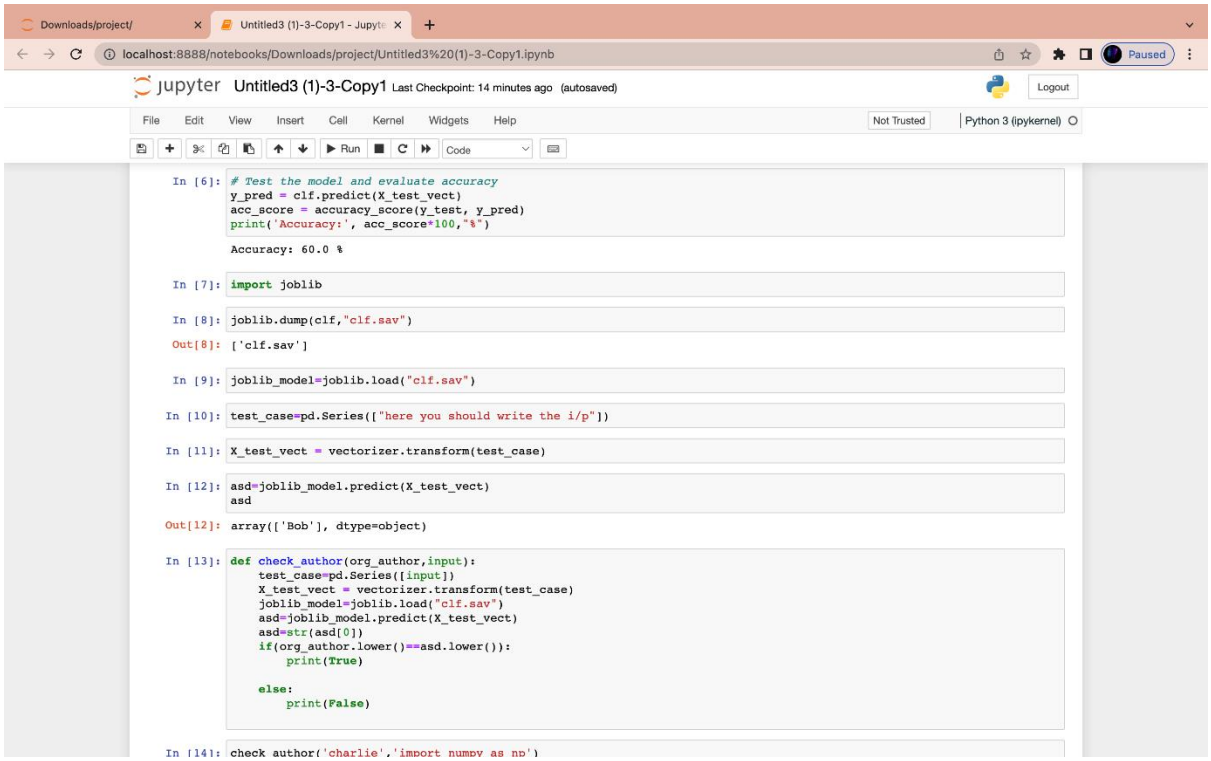
```
In [3]: # Split the dataset into training and testing sets
        X_train, X_test, y_train, y_test = train_test_split(df["CodeChange"], df["Author"], test_size=0.1, random_state=10)
```

```
In [4]: # Convert the code snippets into feature vectors
        vectorizer = TfidfVectorizer()
        X_train_vect = vectorizer.fit_transform(X_train)
        X_test_vect = vectorizer.transform(X_test)
```

```
In [5]: # Train a logistic regression model
        clf = LogisticRegression(random_state=10)
        clf.fit(X_train_vect, y_train)
```

```
Out[5]: LogisticRegression(random_state=10)
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.



Downloads/project/ | Untitled3 (1)-3-Copy1 - Jupyter

localhost:8888/notebooks/Downloads/project/Untitled3%20(1)-3-Copy1.ipynb

jupyter Untitled3 (1)-3-Copy1 Last Checkpoint: 14 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help | Not Trusted | Python 3 (ipykernel)

```
In [6]: # Test the model and evaluate accuracy
        y_pred = clf.predict(X_test_vect)
        acc_score = accuracy_score(y_test, y_pred)
        print('Accuracy:', acc_score*100,"%")
```

```
Accuracy: 60.0 %
```

```
In [7]: import joblib
```

```
In [8]: joblib.dump(clf,"clf.sav")
```

```
Out[8]: ['clf.sav']
```

```
In [9]: joblib_model=joblib.load("clf.sav")
```

```
In [10]: test_case=pd.Series(["here you should write the i/p"])
```

```
In [11]: X_test_vect = vectorizer.transform(test_case)
```

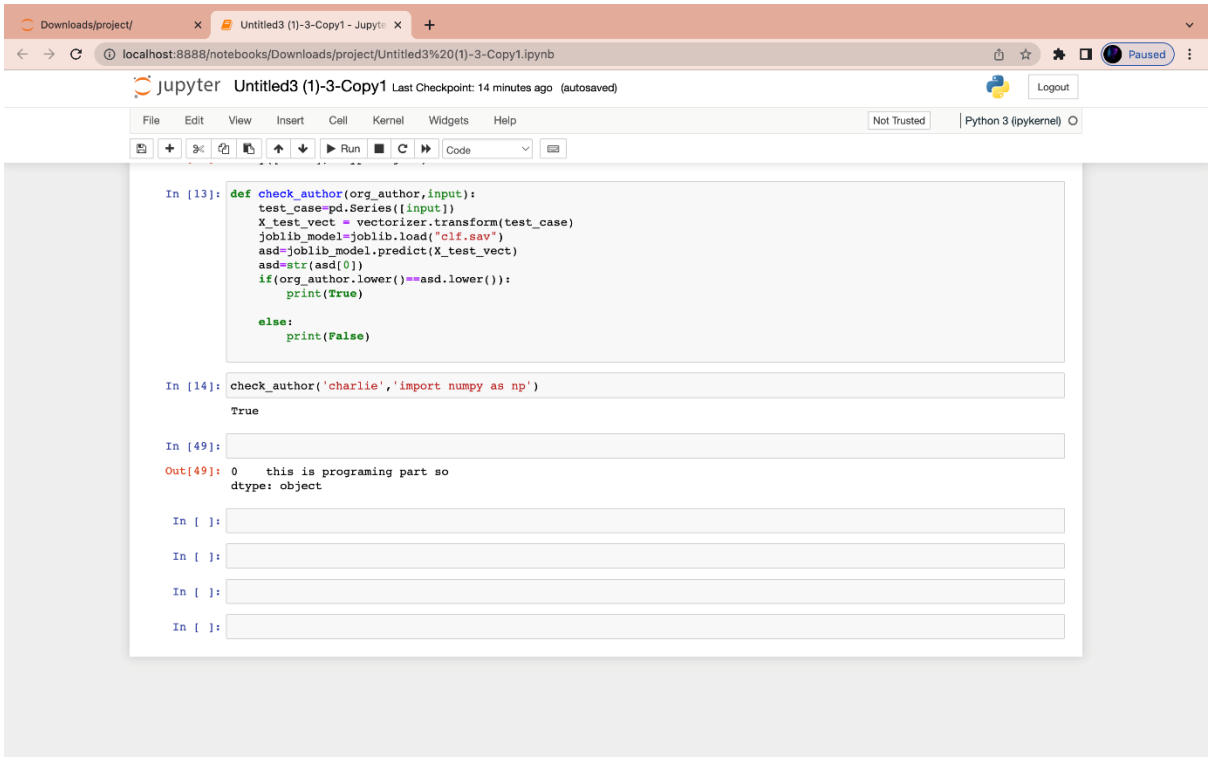
```
In [12]: asd=joblib_model.predict(X_test_vect)
         asd
```

```
Out[12]: array(['Bob'], dtype=object)
```

```
In [13]: def check_author(org_author,input):
             test_case=pd.Series([input])
             X_test_vect = vectorizer.transform(test_case)
             joblib_model=joblib.load("clf.sav")
             asd=joblib_model.predict(X_test_vect)
             asd=str(asd[0])
             if(org_author.lower()==asd.lower()):
                 print(True)

             else:
                 print(False)
```

```
In [14]: check_author('charlie','import numpy as np')
```



This project introduces the method for classification of author based on high level programming features. Paper describes author identification using high-level features that contribute to source code authorship identification using Logistic Regression Model.

Conclusion

Feature extraction and classification to identify the authorship of document is active research area. Above discussion provides the direction to which we could move. In review of various research papers we found unigram feature which focused on word sequences but other features like two and more sequences.

References

[1] Johannes Furnkranz, “A Study using n-gram Feature for 30, 1998

[2] Maria Fernanda Caropreso, “Statistical Prases in Automated Text Categorization,” IEI-B4-07-2000. Pisa, IT, (2000).