



DSC 2CC PROJECTS

Umang Raval

18BCE0170

Btech CSE

Topics

1. Plagiarism check
2. Machine Learning on wildlife strike
3. App for URL Shortening
4. App for Note making

Abstract

Ever since the popularization of the internet, things began to simplify. Prediction of event statistics in less time has made it more comprehensible to help us analyze the big data content which humans are not capable of. Creation of a website made it possible to grow our business, socialize with professionals and friends. New technology is developed everyday to minimise day to day with comfort.

Introduction

My projects involve development of plagiarism check REST api which checks the percentage of plagiarism in two documents or paragraphs. It is very useful in schools and colleges to identify the assignments of students which compares with other classmates assignments to identify copying. Although the use of someone else's work in order to gain academic credit may meet some legal definitions of fraud.

The use of url sharing has increased for things like joining servers on discord or share links from amazon etc. It is difficult to handle long endpoints so it's always better to shorten url. Say you want to provide a link on a business card, in an advertisement, or in another situation where hyperlinking isn't ideal. A shortened URL takes up less space, is more memorable and keeps your text tidy.

The workload in day to day life has increased due to growth in science and technology. It's not possible to remember every task other than routine. Use of pads to make notes is a bit of old fashioned, so using a note making app is easy to use and paperless.

To get to statistical findings from large data is hard for humans to calculate and plot it. It's better to use a Data Visualization technique to simplify data and present it to a layman. It is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from. The main goal of data visualization is to make it easier to identify patterns, trends and outliers in large datasets. The term is often used interchangeably with others, including information graphics, information visualization and statistical graphics.

Methodology

Creating REST apis for plagiarism checkers requires basic authentication of users. A signup was made which checks if the user exists or not and defaults

gives 10 tokens for use. Passwords were hashed with the bcrypt module before storing it to the database. A detection class was made which gets posted data from user checks if username exists if not throws error. Checks on the number of tokens is also made like if it is less than 0 then it will throw an error. Then the natural processing module was loaded (spacy) which gets the ratio of similarity between the two paragraphs posted. After successfully finding, the ratio is sent as a JSON response. Token is updated in the user database. Endpoint for refilling the tokens is also provided. It authenticates users with username and password, if a user exists then it adds a refill amount to current tokens in the database.

The image displays three sequential screenshots of a REST client interface, showing the details of three different API requests and their corresponding JSON responses.

First Screenshot: Register Endpoint

- Request:** POST `http://localhost:5000/register`. Status: 200 OK. Time: 640 ms. Size: 62 B.
- Request Body (JSON):**

```
{  "username": "balso",  "password": "tpas"}
```
- Response Body (JSON):**

```
{  "msg": "You successfully signed up for the API",  "status": 200}
```

Second Screenshot: Detect Endpoint

- Request:** POST `http://localhost:5000/detect`. Status: 200 OK. Time: 1.76 s. Size: 91 B.
- Request Body (JSON):**

```
{  "username": "balso",  "password": "tpas",  "text1": "i am with you",  "text2": "i am besides you"}
```
- Response Body (JSON):**

```
{  "msg": "Similarity score calculated successfully",  "ratio": 0.8984487156505843,  "status": 200}
```

Third Screenshot: Refill Endpoint

- Request:** POST `http://localhost:5000/refill`. Status: 200 OK. Time: 3.59 ms. Size: 45 B.
- Request Body (JSON):**

```
{  "username": "admin",  "admin_pw": "abc123",  "refill": "20"}
```
- Response Body (JSON):**

```
{  "msg": "Refilled successfully",  "status": 200}
```

The problem statement describes a wildlife strike is a collision between an animal (most often a bird, but sometimes another species) and a man made vehicle, especially aircraft. Wildlife strikes constitute a serious hazard to aircraft safety and have caused a number of fatal accidents. The Federal Aviation Authority counted 177,269 wildlife strike reports on civil aircraft between 1990 and 2015, growing 38% in seven years from 2009 to 2015. Birds accounted for 97%. Dataset provided had 66 features related to the damage caused to airplanes and the species of bird that caused the damage. The given dataset was converted to a dataframe. Only columns which were useful for concluding the answer were extracted like

('Windshield Damage', 'Nose Damage', 'Engine1 Damage', 'Engine2 Damage', 'Engine3 Damage', 'Engine4 Damage', 'Propeller Damage', 'Wing or Rotor Damage', 'Fuselage Damage', 'Landing Gear Damage', 'Tail Damage', 'Lights Damage', 'Other Damage'). Then it was grouped by 'Species Name' which gave us the number of planes damaged. From that the answer to 'What bird species has caused the most damage to airplanes?' was Unknown medium bird (38531). For 'Which part of the airplane which is most prone to damage in wildlife strikes?' We use a loop in the data frame to get strikes corresponding to airplane parts which conclude to Wing or Rotor Damage.

```
In [128]: bs = pd.DataFrame(data.groupby('Species Name').count()['Record ID'])
          bs.columns = ['No of planes damaged']
          bs
```

```
Out[128]:
```

Species Name	No of planes damaged
ACADIAN FLYCATCHER	5
ALDER FLYCATCHER	11
ALLEN'S HUMMINGBIRD	1
ALLIGATOR SNAPPING TURTLE	1
AMERICAN ALLIGATOR	19
...	...
YELLOW-THROATED VIREO	2
YELLOW-THROATED WARBLER	19
YUMA MYOTIS	1
ZEBRA DOVE	290
ZENAIIDA DOVE	4

715 rows × 1 columns

```
In [129]: ans1 = bs[bs['No of planes damaged'] == bs['No of planes damaged'].max()]
ans1
```

```
Out[129]:
```

	No of planes damaged
Species Name	
UNKNOWN MEDIUM BIRD	38531

```
In [101]: damage = []
for i, j in strike.iteritems():
    part = [i]
    part.append(j.sum())
    damage.append(part)

ti = pd.DataFrame(damage)
ti.columns = ['Part', 'Strikes']
ti
```

```
Out[101]:
```

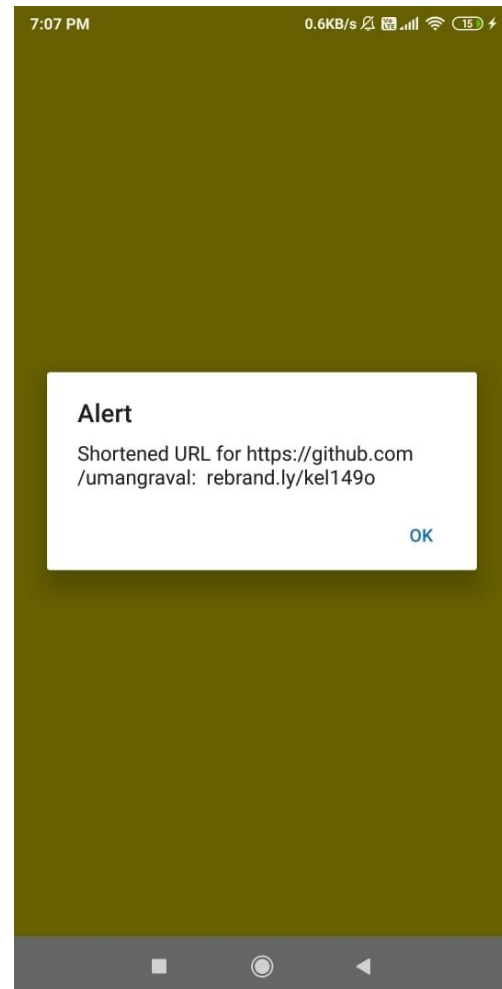
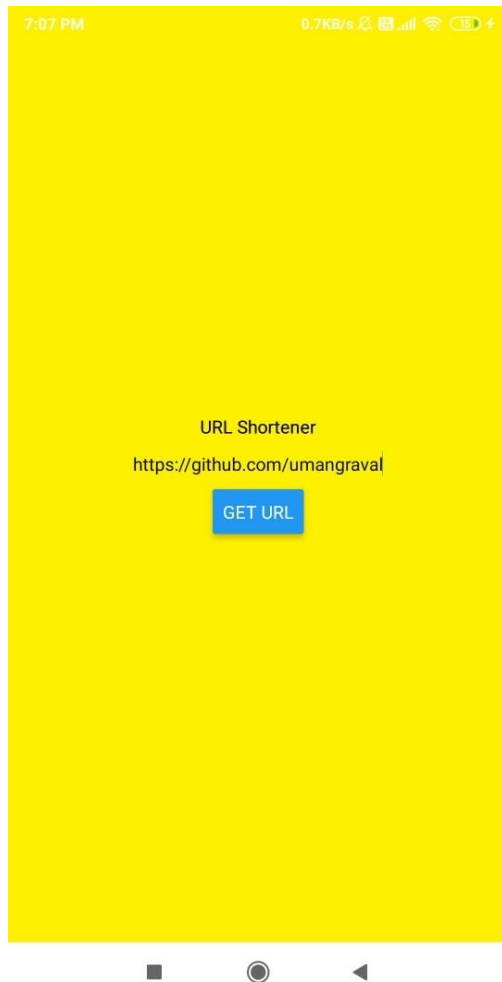
	Part	Strikes
0	Windshield Damage	1037
1	Nose Damage	1145
2	Engine1 Damage	2515
3	Engine2 Damage	2023
4	Engine3 Damage	169
5	Engine4 Damage	74
6	Propeller Damage	579
7	Wing or Rotor Damage	4180
8	Fuselage Damage	824
9	Landing Gear Damage	1012
10	Tail Damage	727
11	Lights Damage	734
12	Other Damage	1565

```
In [112]: ans2 = ti[ti["Strikes"] == ti["Strikes"].max()]
ans2
```

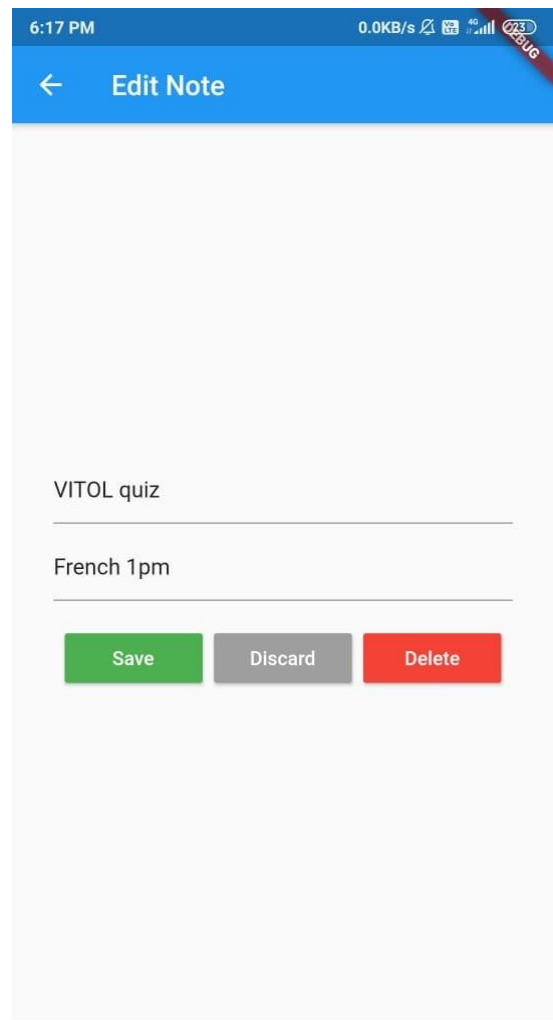
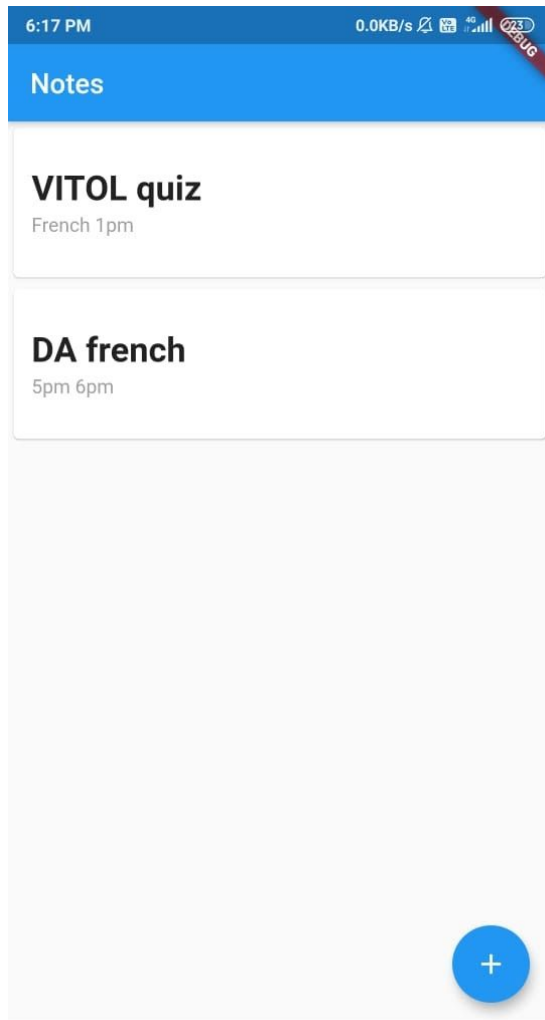
```
Out[112]:
```

	Part	Strikes
7	Wing or Rotor Damage	4180

Basic template was used to set up the app using the expo client. Using react-native npm package inbuilt UI components like textfield, button. A function was created to change the state on change of text in textbox. On submit, a post request was sent to ‘ <https://api.rebrandly.com/v1/links> ’ with registered api key in header. The response was shown as an alert msg with shortened url.



Notes app divided into views like note card and note list. Note list is defined to be a stateful widget. A provider is defined to create a Note class in which static future functions for create, update, delete and get all notes were defined. The home page consisted of all notes created. A floating action button to navigate to create a new note page with title and a body. Also gesture onclick was added to delete and update notes.



Technology Used

Plagiarism checker was built on the Flask web framework of python. Flask has its foundation around Werkzeug and Jinja2 and has become one of the most popular Python web application frameworks. As a developer in developing a web app in python, you may be using a framework to your advantage. A framework is a code storage that should help developers

achieve the required result by making work easier, scalable, efficient and maintainable web applications by providing reusable code or extensions for common operations. For testing APIs postman was used. Postman is a popular API client that makes it easy for developers to create, share, test and document APIs. This is done by allowing users to create and save simple and complex HTTP/s requests, as well as read their responses. For detection similarity, Spacy was used. SpaCy helps you build applications that process and “understand” large volumes of text. It can be used to build information extraction or natural language understanding systems, or to pre-process text for deep learning. Storing data is very important and should be secure. MongoDB is an open-source document database and leading NoSQL database. One single main benefit it has over MySQL is its ability to handle large unstructured data. It is magically faster because it allows users to query in a different manner that is more sensitive to workload.

Visualizing large data python is most often used. One of the main reasons why Python is widely used in the scientific and research communities is because of its ease of use and simple syntax which makes it easy to adapt for people who do not have an engineering background. It is also more suited for quick prototyping. Jupyter Notebook is used for writing code, the notebook lets you write different types of text. Here, you can see formatted explanatory text, a gray block of code, and a visualization. It kind of looks like a textbook, except that this notebook can be accessed by students on their computers, and all of the code is live—students can run through each part of the computation to see the result. Pandas module was used for creation of Data frames.

React Native is used for url shortener. It is an exciting framework that enables web developers to create robust mobile applications using their

existing JavaScript knowledge. It offers faster mobile development, and more efficient code sharing across iOS, Android, and the Web, without sacrificing the end user's experience or application quality. And Rebrandly which is the industry-leading link management platform to brand, track and share short URLs using a custom domain name. The platform used for building was Expo which we found it fast and easy to use.

Note Making app comprises many UI components so we choose to go with Flutter. The first advantage of Flutter is its common UI patterns extracted out to reusable widgets. It eliminates the need to individually set UI properties such as colors, styling, etc. We can adjust UI and business logic globally through editing the code for both IOS and Android Flutter apps. For storing notes locally on device SQLite packages which provide faster inserts, updates, and queries, compared to other local persistence solutions.

Conclusion

Building apps from scratch and understanding layout with each UI component helps better inner workings of native apps. Data extraction and visualizing patterns in large datasets helps simplify the working process and understand the proceeding conclusion.

Github Link: <https://github.com/umangraval/DSC-2CC-Projects>