**FAKE JOB DETECTION**

Considering that little has been done on the website provided, I’ll be providing a list of methods as a suggestion for Data collection and the classification of job alerts into 2 parts, **Fake** and **Real** using Machine Learning.

Data Collection

Now there are a number of ways we can collect data for this project. A number of sources could be contacted job alerts, as a matter of fact, agencies like Jobberman and Indeed could be direct sources of such job alerts. I recommend that these data be collected and stored in SQL tables because that’s the most popular means of data storage and also it is easily accessible and our machine learning model can be trained based off of the dataset read directly from SQL.

First of all we need to consider the likely pointers that we can use to identify that a particular job advert is either Real or Fake. Some of them include:

* Company Name: Ensuring the company name is a registered and recognized company in the country. If the company name is not recognized, it is likely a fake advert.
* Company Address: Ensuring the address stated is the address of an actual place on the map. If the address is not locatable, it is likely a fake advert.
* Unprofessional Emails: If there are wrong grammatical constructs, spelling, capitalization, punctuation errors in the email/message sent, then it’s likely a fake advert.
* Job Description: If the job description is vaguely expressed then it’s likely fake.
* Job Requirement: If the job requirement is vaguely expressed then it’s likely fake.
* No prior application: If you didn’t really apply of the job and they contacted you then the job advert is likely fake.
* Interviews via Messaging Services: Many scams show the interview taking place online using an instant messaging service.
* Also if email/message does not include the company’s address and phone, then it’s most likely fake.
* If you’re required to provide confidential information then it’s likely a fake advert.

So from the above breakdown, we can get an idea of what our data is likely going to look like. The four (4) fundamental data columns required now are:

1. Company Name
2. Company Address
3. Company Phone/Email
4. Job Description

Machine Learning Model

The model could work in such a way that the description column would be extracted and used (sentiment analysis carried out on it). The TfidfVectorizer function in sklearn could be imported and utilized in analyzing words in the job description; this will help spot out some keywords that would be specified in the text. These keywords take the form of grammatical errors, unrecognizable words, wrongly spelt words and so on. If such keywords appear more than twice, then a notice is displayed that such job alert is fake.

Another way to do this is the adopt the capabilities of the googlemaps library. It is a library in python that utilizes google maps API to locate places and coordinates on the map. We could apply this running an indepth search of the company’s address column. If it is found, return Real else return Fake. We can do same for the company name.

Concerning company email, a simple method is to apply the string.split(‘@’) method to split/separate the email using @. If the string after the @ is not equal to one of a list of regular email service providers then it’s likely fake. This is a probabilistic approach and so we cannot just hypothesize that its fake. A more in-depth analysis has to be done.

Finally, we could carry out one-hot encoding on the four features listed above. Dummy variables can be created using the conclusions drawn from each of the columns but representing real with 1 and fake with 0. This will create a series of numeric features that can be passed on to a machine learning model, trained and deployed for use in the app/website.