How To Detect Job recruiting fraud

Team REAL
Danny, Allen, Tommy, Caroline, Mochi

TABLE OF CONTENTS

TABLE OF CONTENTS

Have you ever seen this?

From: *[Redacted] <[redacted]@gmail.com>*
Subject: *JOB POSITION OPPORTUNITY FOR STUDENT*

Job Scam!

Dear Student,

I'm very happy to inform you about the job opportunity in conjunction with your school (The University of Houston) we got your mail from your school data base. Our reputable company (CiscoSystems Company) is running a student empowerment program. This program is to help the hardworking student to secure a work at home job, this will not stop you from your daily works and your school activities. All you need is jst an hour or two to carry out the job weekly. Your wages will be $350 USD per week.

Kindly get back to us with your PHONE NUMBER AND PERSONAL EMAIL IF YOU ARE INTERESTED IN THE JOB POSITION

PHONE NUMBER:
PERSONAL EMAIL:

Regards

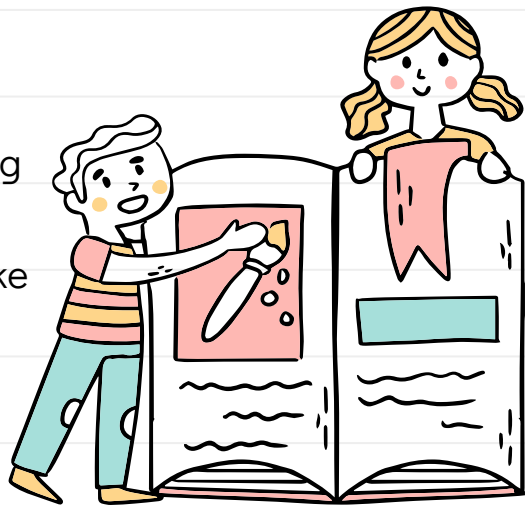
[Redacted]
Recruiting Manager
CiscoSystems

Background

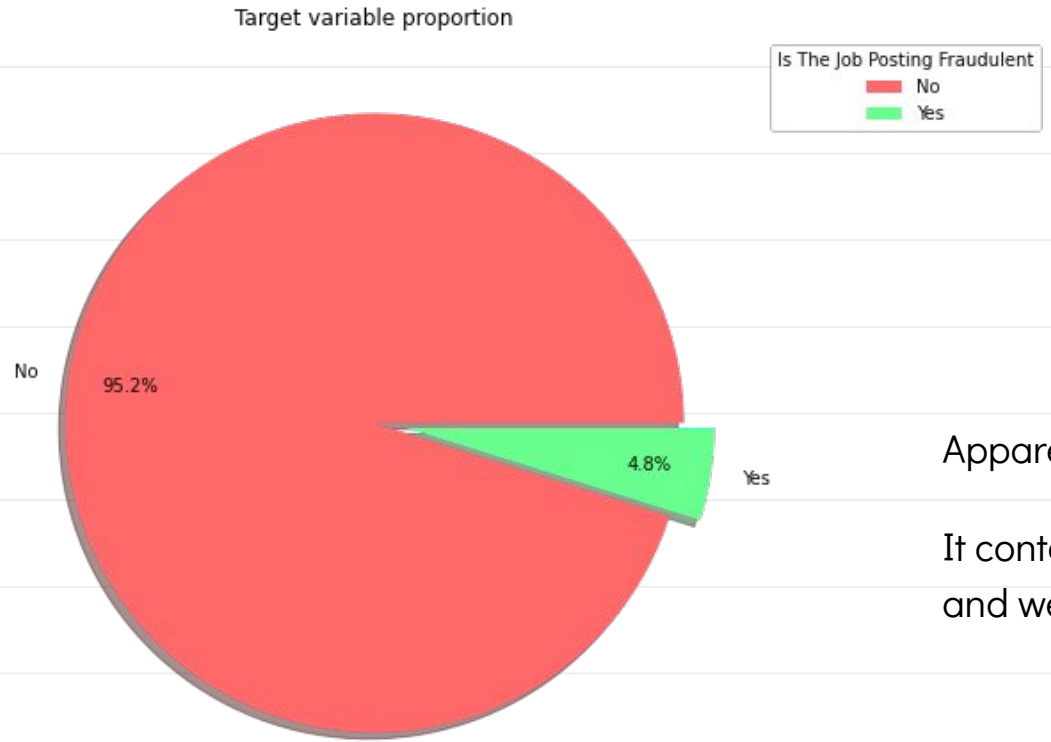
- The problem motivates us is the possibility of **identifying fake job recruiting news**
- People who posted fake job recruiting news may aim at
  - Collecting personal information (even sell it)
  - Providing lucrative job opportunities and asking for money

The methods we are considering includes but not limited to:

- Recognition of fake job posting by deep learning (NN) skills, including transferring text into code, model training & testing ect.
- Utilize LIME, sentiment analysis to check the similarities between fake ones



Descriptive Statistics
- Is this dataset balanced?



17,879
postings

866
fake

Apparently, this dataset is **unbalanced**.

It contains only 4.8% fake job recruiting news, and we should keep that in mind.

The Employment Scam Aegean Dataset form The University of the Aegean (http://emscad.samos.aegean.gr/)

TABLE OF CONTENTS

Data-PreProcessing

17,879 postings

600 tokens

Target

Fraudulent

Categorical Variables
Telecommuting
Has_company_logo
Has _questions
Employment_type
Required_experience
Required_education

Text Variables
Title
Department
Company_profile
Description
Requirements
Benefits
Industry
Function

Special Variables
Salary_range
Loation

- 6 Categorical Variables - Give scales or get dummy
- 8 Text Variables - Combine and apply **TextVectorization()**
- 2 special variables
  - Salary range - Separate the range & get upper and lower bound
  - Location - Combine location with Text

- For the missing values, we fill in numerical ones as -1, text ones with ‘unknown’

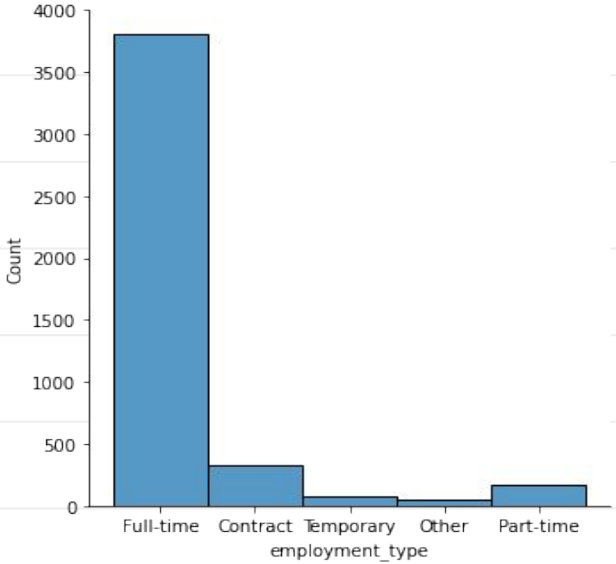
EDA - Map of location

North America

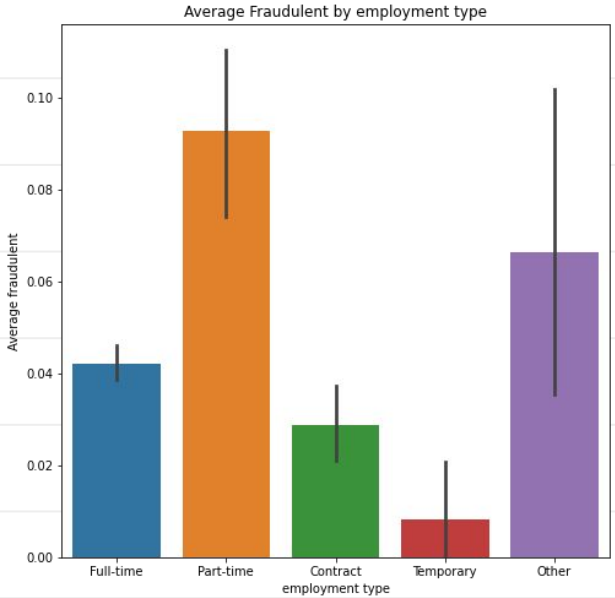
Europe

EDA – Some major categorical Features

Employment type

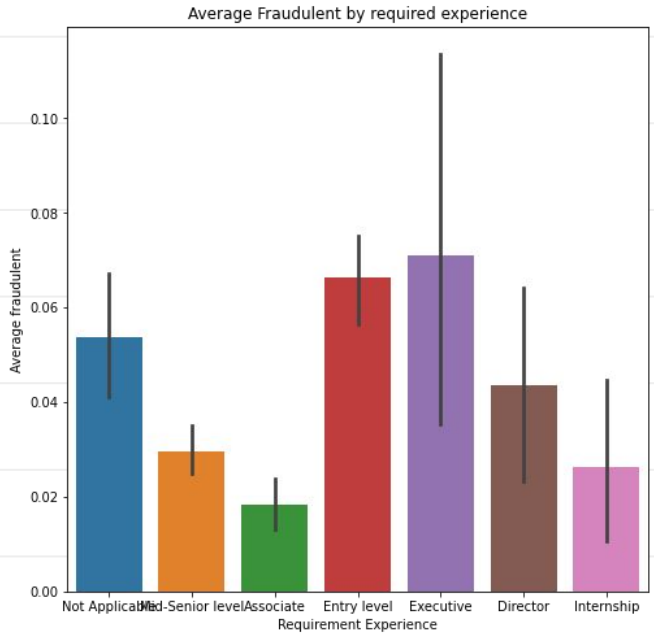


Avg Fraudulent by Employment Type



EDA – Some major categorical Features

Avg Fraudulent by Required Experience



Avg Fraudulent by Requirement of Education

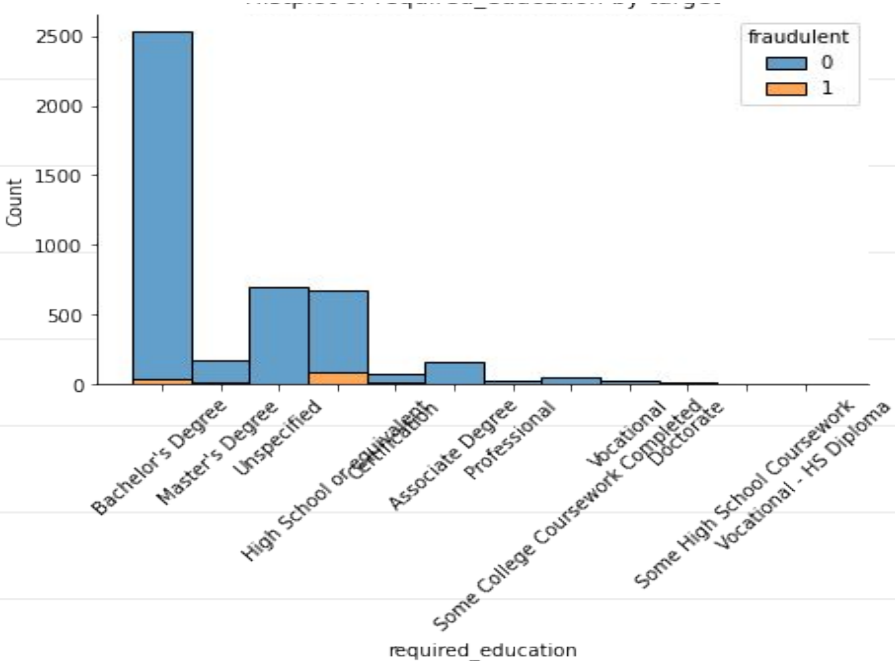


TABLE OF CONTENTS

How we prepare the layers for our model

How we deal with text

- First, we used **TextVectorization() layer** to preprocess our text;
- Then we one-hot encoded these integer sequences but it crashed.

How we deal with numerical

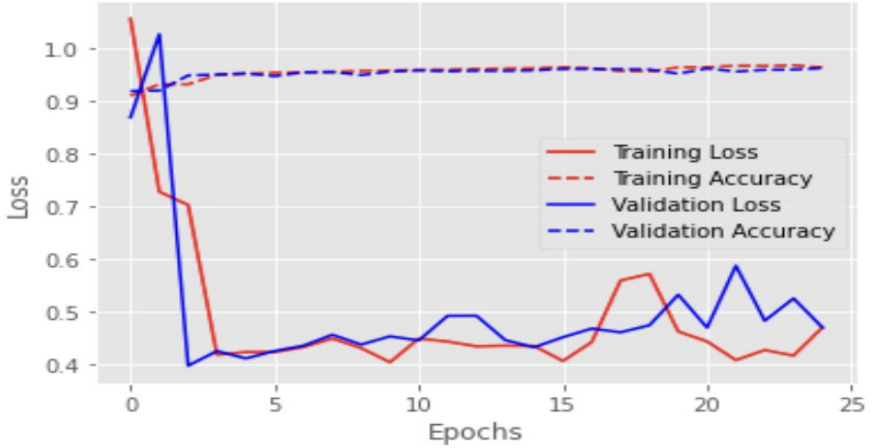
- After preprocessing, we added another layer to deal with all the numerical values.

NN ARCHITECTURE

MODEL SUMMARY

- **Embedding layer & Numeric input**
- **Loss function -** binary_crossentropy
- **Train-Test Split** - [:12516] (70/30%)
- **Cross validation** - k=4
- **Activation** - 'sigmoid', ' relu', 'linear'
- **Optimizer** - Rmsprop
- **Dropout** - 0.5
- **Training-Accuracy**: 97.91%
- **Validation-Accuracy**: 97.51%
- **Test-set Accuracy** - 94.89%

```
def build_model_embed():
    inputs = keras.layers.Input(shape=(1), dtype="string") # We take our strings as input
    processing = text_vectorization(inputs)

    # Truncates after 600 tokens, and pads up to 600 tokens for shorter reviews.
    # Mask zero means it will skip 0 tokens and will not pass them on.
    embedding = keras.layers.Embedding(input_dim=2000,output_dim=8,input_length=600, mask_zero=True)(processing)

    x = keras.layers.Bidirectional(keras.layers.LSTM(32))(embedding)
    x = keras.layers.Dropout(0.5)(x)
    outputs = keras.layers.Dense(1, activation="sigmoid")(x)

    # Define numeric input branch
    input_numeric = keras.layers.Input(numeric_cate_subset.shape[1],name="Numbers")
    x = keras.layers.Dense(3,activation="relu")(input_numeric)
    x = keras.layers.Dense(3,activation="relu")(x)
    numeric_output = keras.layers.Dense(5,activation="linear")(x) #Another option might be to have dense matrice

    merge = keras.layers.Concatenate()([outputs,numeric_output])
    x = keras.layers.Dense(5,activation="relu")(merge)
    final_output = keras.layers.Dense(1)(x)

    model = keras.Model(inputs=[inputs, input_numeric],outputs = final_output)
    model.compile(optimizer="rmsprop",loss="binary_crossentropy",metrics=['accuracy'])
    return model
model = build_model_embed()
```



Pre-trained embeddings: GLOVE

Model summary

- **Global Vector Representation**
- **Loss Function -** binary_crossentropy
- **Train-Test Split** - [:12516]
- **Cross validation** - k=4
- **Training-Accuracy**: 95.09%
- **Validation-Accuracy**: 95.24%
- The resulting model is about 84.06% accurate in the holdout sample.

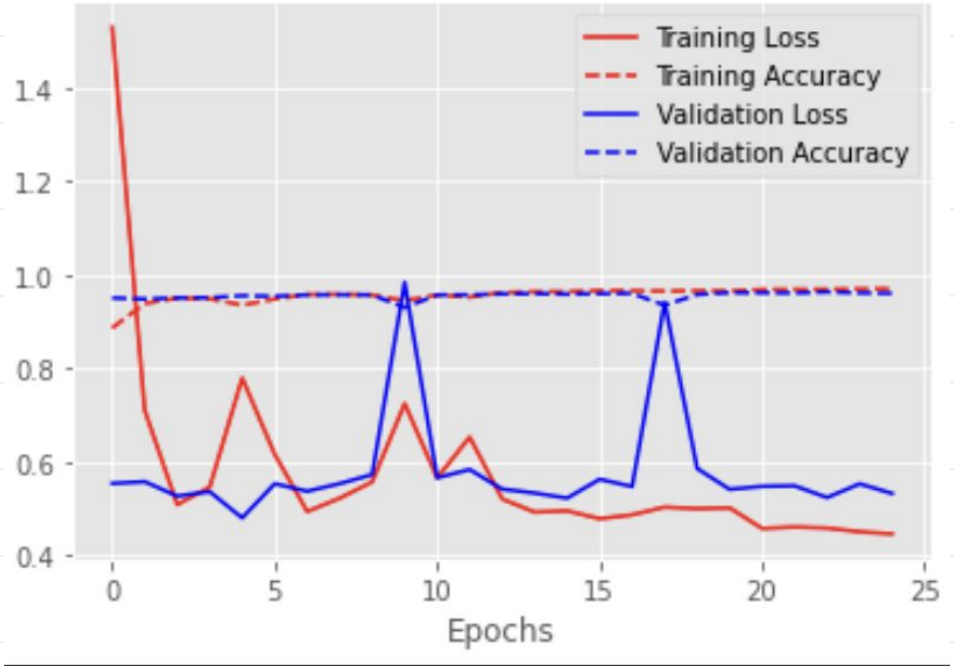
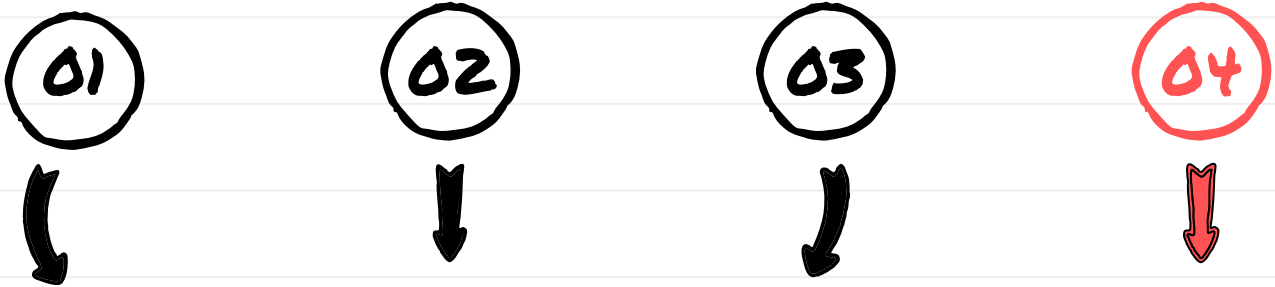


TABLE OF CONTENTS



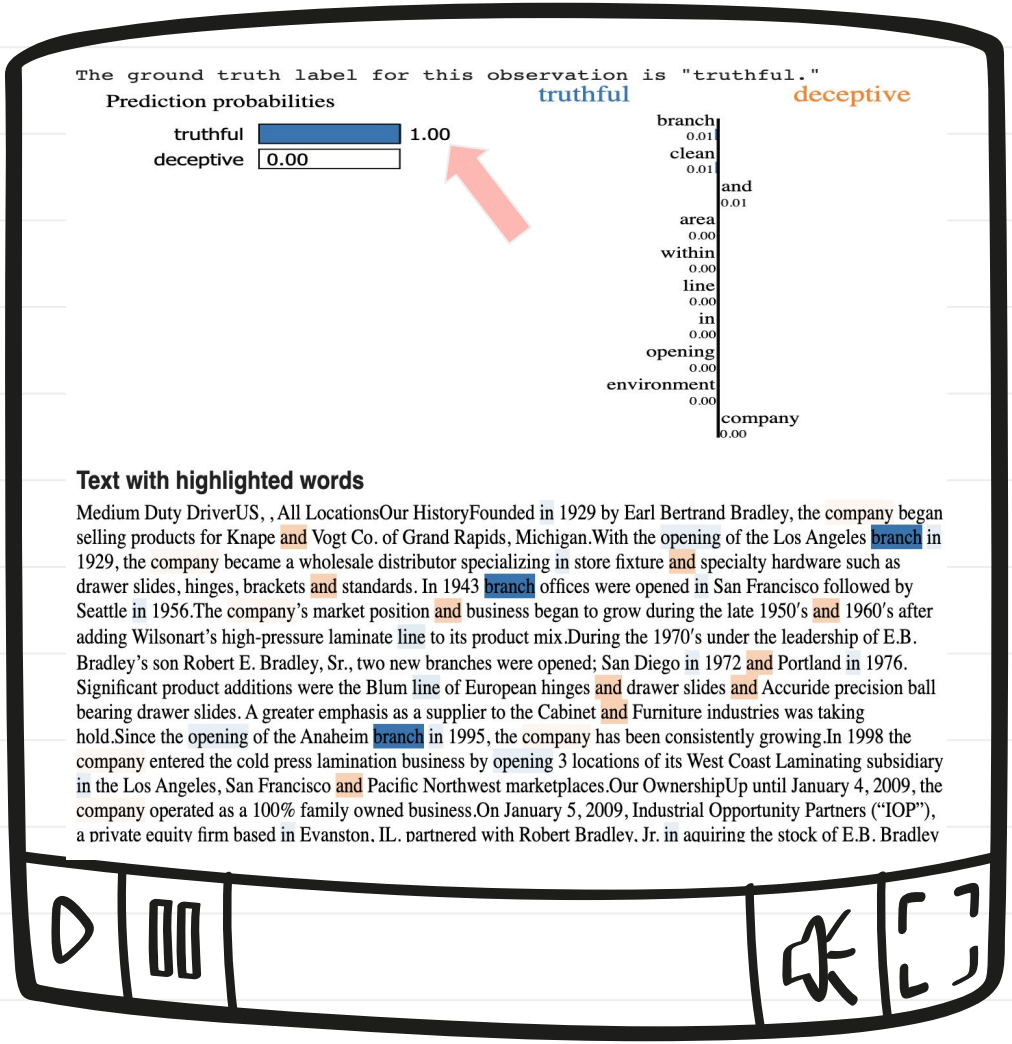
LIME

After doing a 20/80 test-train split, we trained our model processing text.

Accuracy = 98.34%

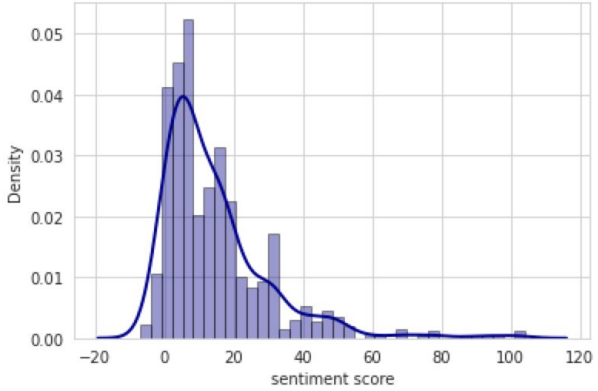
Randomly calling the one observation out, we tried multiple obs, we almost have 100% sure of predicting real job postings, around 99% probability of predicting fake ones.

Sample results showed as listed..

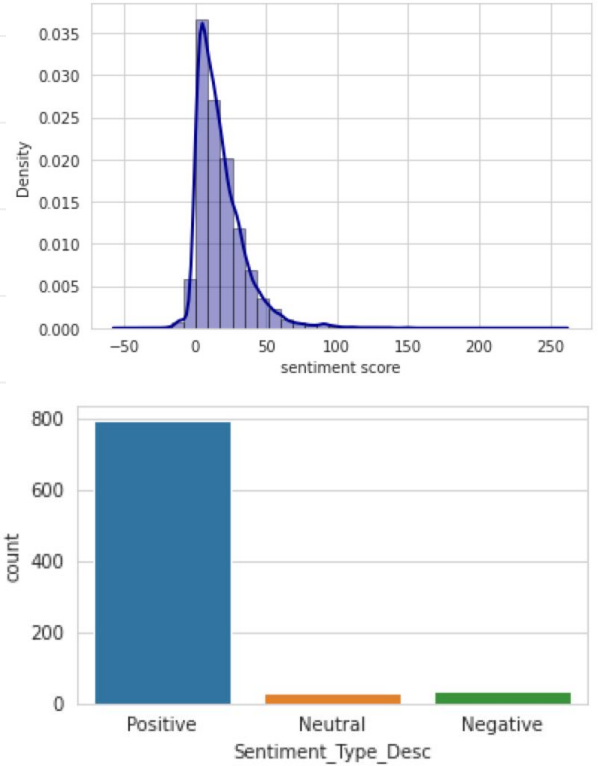


Sentiment Analysis

Fraud / Avg = 14.60



Non-Fraud / Avg = 17.98



Word cloud

Company File

Benefits

It turns out that indeed our guesses are partially right. The fake ones do tend to promise bonus and benefits, supply with online training, guarantee work life balance.

Word cloud

Job Description

Job Requirements

The fake job recruiting postings tend to describe their projects or products, we may assume they are just empty talk. For the requirements, they tend to stress experience - something can not be quantified.

Conclusion

Model 1

Our self designed model has **validation accuracy around 97%.**

Model 2

The pre-trained model has **validation accuracy around 95%.**

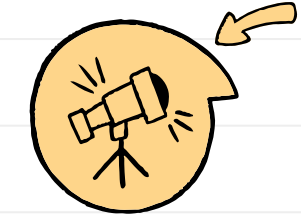
LIME

Only focused on text, the accuracy among test set is around 98%.

Sentiment Word Cloud

Fake ones tend to talk big, guarantee great welfare.

Suggestions

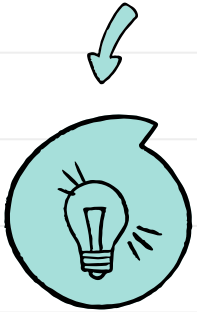


Model

We tried to implement one-hot encoding processing the integer sequences, only it got crashed.

But for the first glance, the performance after one-hot is better.

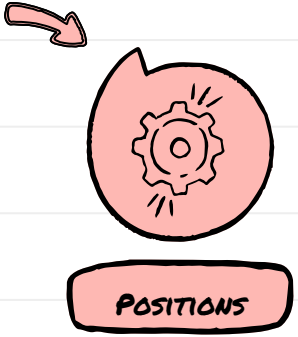
For the embeddings, we can increase the depth of it (utilize more tokens).



Job seekers

Fake job recruiting postings tend to focus on entry level or executive level; part-time job.

Be cautious when applying for these types of positions.



Positions

After doing sentiment analysis and word cloud towards fake ones, we figured out that they tend to talk big about themselves, but avoid some essential information.

When seeing great welfare and work environment, be cautious.

"Think positive.
Be cautious."

— Someone Famous 😉

THANKS FOR LISTENING!

- Team REAL -