Identifying Fraudulent Job Postings

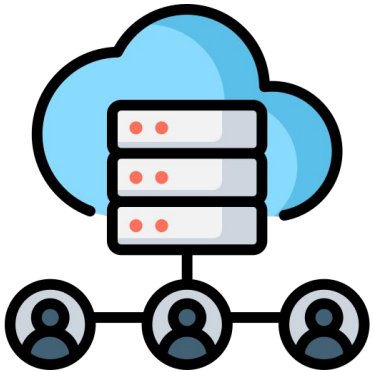
INFO 251 Applied Machine Learning
Morris Chang, Gina Shuyao Wang, Eric Yuxin Miao

Table of Contents

Introduction & The Problem

- Increase usage of online employment websites have lead to increase in fraudulent job postings.
- Fraudulent job posting have two main goals:
  - Acquire confidential personal information.
  - Solicit unlawful payments
- GOAL: Develop a Natural Language Processing (NLP) model that is able to detect fake job postings based on the textual description of the jobs.
- Utilize different kind of machine learning models and algorithms to identify patterns or anomalies.

Data Acquisition



- Employment Scam Aegean Dataset (EMSCAD)
- Published by the University of the Aegean's Laboratory of Information & Communication Systems Security
- Publicly Available Dataset that contains 17,880 real-life job postings online.
    - 17,014 Legitimate Job Postings and 866 fraudulent job postings.
- EMSCAD records were manually annotated and classified into two categories between 2012-2014.

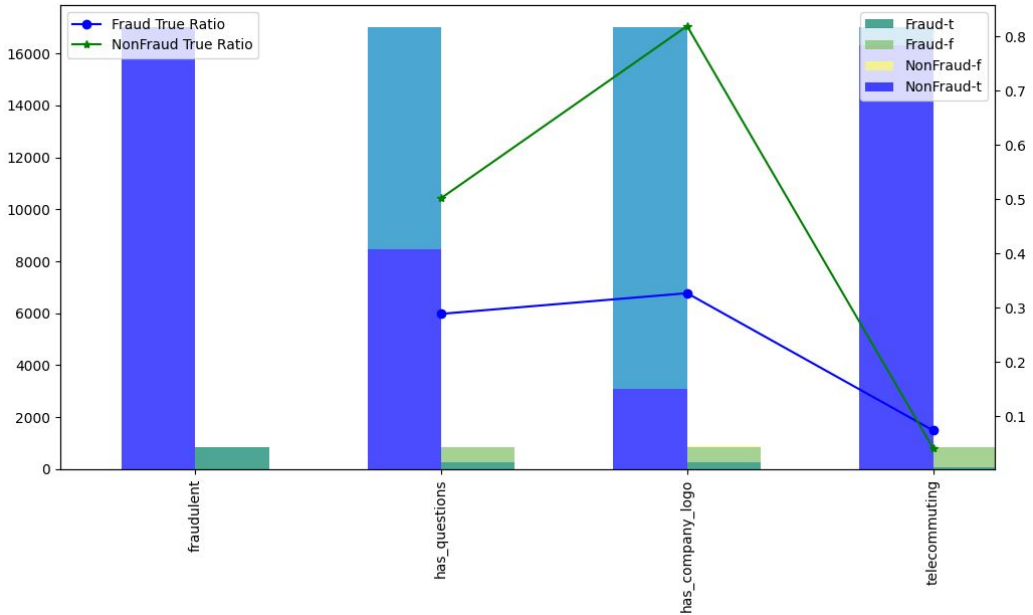
EDA

- Overall binary and nominal features does not show strong relationships.
- Take a closer look at the string and text entries.

Type	Name
String	Title
	Location
	Department
	Salary range
HTML fragment	Company profile
	Description
	Requirements
	Benefits

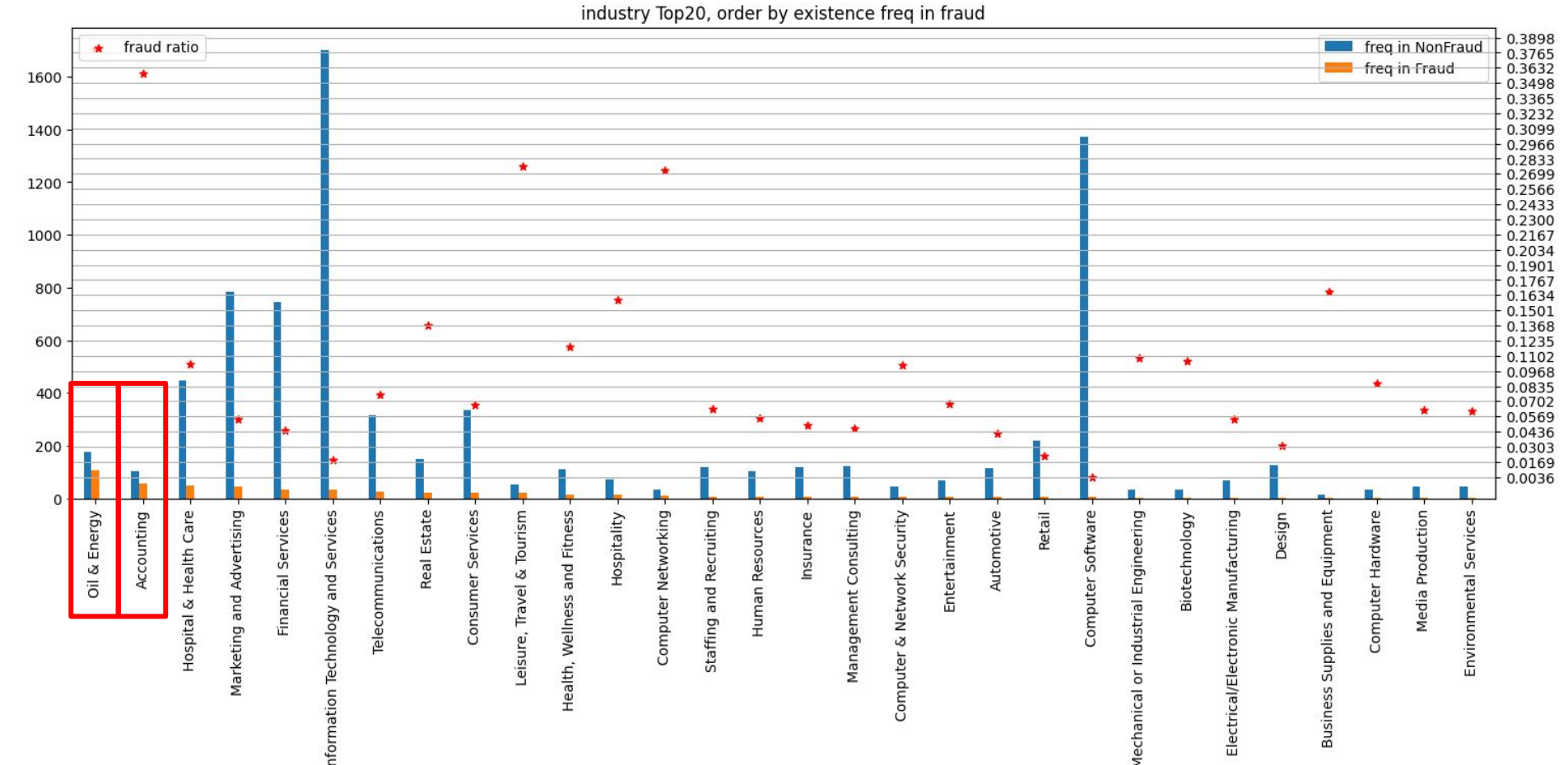
Type	Name
Binary	Telecommuting
	Company logo
	Questions
	Fraudulent
Nominal	Employment type
	Required experience
	Required education
	Industry
	Function

EDA & Data Preprocessing

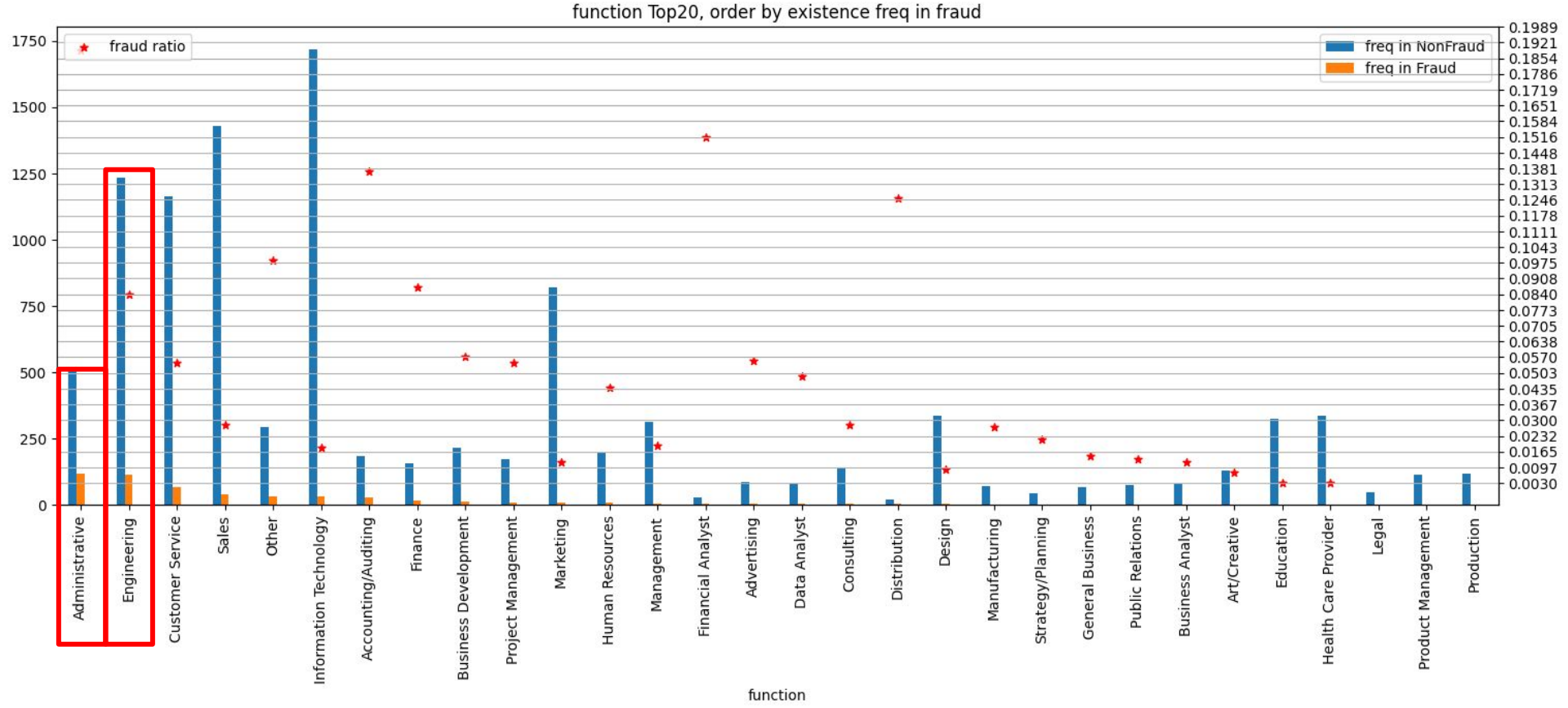


- Binary/Categorical values does not show strong relationship.
- Fraudulent postings distribution are similar across the 3 features.

Specific Industries have higher rate of fraudulent postings



Specific Job Functions have higher rate of fraudulent postings



Data Preprocessing

- Base on HTML/text features
- Concatenate String & Nominal features if not existed before
- Remove
  - Stop words
  - Punctuations
  - Etc.
- Keep
  - Numbers
  - Text
- Fraudulent
  - T == 1
  - F == 0

Type	Name
String	Title
	Location
	Department
	Salary range
HTML fragment	Company profile
	Description
	Requirements
	Benefits

Type	Name
Binary	Telecommuting
	Company logo
	Questions
	Fraudulent
Nominal	Employment type
	Required experience
	Required education
	Industry
	Function

Key metrics

1. **Recall: Reduce the false negative**
    a. Detect as many scams as possible
    b. Do the best to prevent job seekers from being scammed
2. **Precision: Reduce the false position**
    a. Detect as precise as possible
    b. Do the best to distinguish regular post from scams and prevent positions from being classified as scams and missed

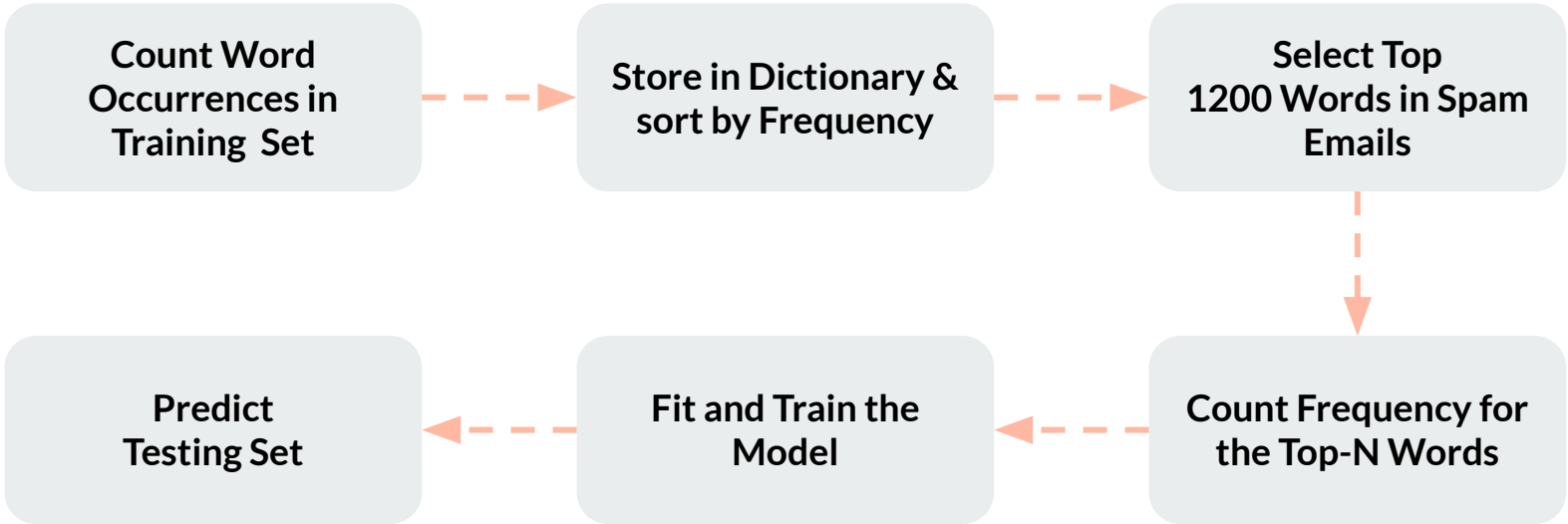
Models

1. Bag-of-words
    a. Traditional Model
        i. Logistic Regression
        ii. Random Forest
2. Neural Networks
    a. LSTM
    b. BERT

Bag-of-words

- The text is represented as a bag of its words and its frequency, and it disregards grammar and order .
- Two approaches:
  - Custom Top n-words Model
    - From 50 - 1200 words
  - CountVectorizer
    - n-gram (1, 2, 3, 4)

Top n-words Model

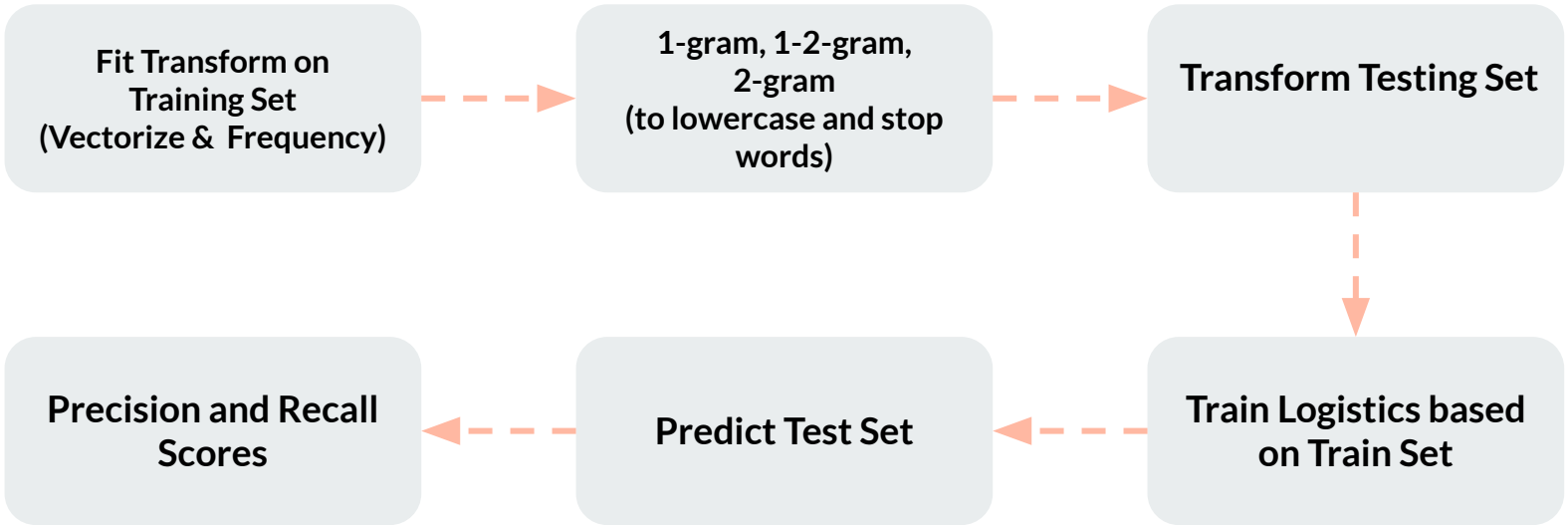


Top n-words

- Testing Top 50, 100 to 1200 words by 100 interval.
- Precision and Recall improves slower after 900 words.
- Many of those words are also present in the non-fraudulent set.
- Questions raised:
  - More words or preserve sequence?

	Accuracy	Precision	Recall	F1
50	0.949888	0.013453	0.428571	0.026087
100	0.958166	0.237668	0.757143	0.361775
200	0.965324	0.394619	0.814815	0.531722
300	0.967338	0.488789	0.773050	0.598901
400	0.971812	0.556054	0.821192	0.663102
500	0.974497	0.654709	0.797814	0.719212
600	0.975168	0.672646	0.797872	0.729927
700	0.975615	0.663677	0.813187	0.730864
800	0.975615	0.681614	0.800000	0.736077
900	0.976734	0.695067	0.811518	0.748792
1000	0.977629	0.699552	0.825397	0.757282
1100	0.978747	0.708520	0.840426	0.768856
1200	0.979418	0.726457	0.839378	0.778846

CountVectorizer and Logistic Regression

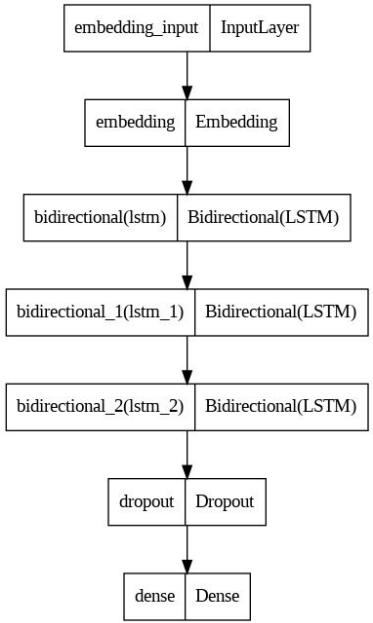


CountVectorizer & Logistic Regression

- Improvement in precision and recall.
- Best performance when 1-gram, and 2-grams are both considered.
- Over predicts when only 2-grams are considered, lead to high False positives.

	Accuracy	Precision	Recall	F1
(1, 1)	0.986130	0.762332	0.949721	0.845771
(1, 2)	0.987025	0.748879	0.988166	0.852041
(2, 2)	0.984116	0.681614	1.000000	0.810667

Neural Networks - LSTM



Hyperparameters

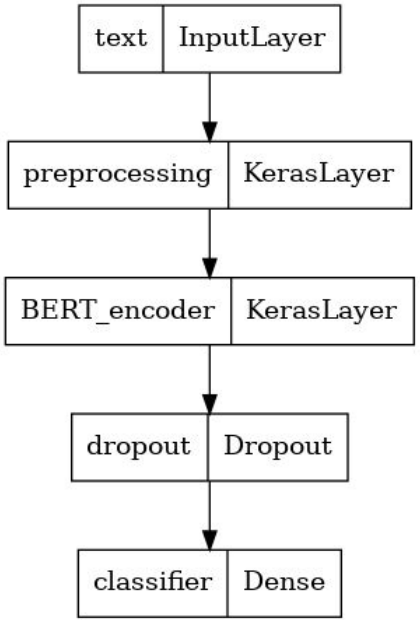
1. Embedding vector size
    a. 50 to 200
2. LSTM layer number
    a. 1 to 3
3. Class weight
    a. Class 0: 1.0
    b. Class 1: 1.0 to 3.0

Neural Networks - LSTM results

- Best Parameters:
  - Embedding dimension = 50
  - Layer of LSTM = 3
  - Class Weight: {0: 1.0, 1: 2.0}

model	embedding	LSTM layer	class_1	recall	precision	accuracy	f1
0	50	1	1	0.980	0.614	0.751	0.755
1	50	1	1.5	0.979	0.677	0.766	0.801
2	50	1	2	0.978	0.655	0.749	0.784
3	50	1	3	0.977	0.695	0.751	0.812
4	50	2	2	0.977	0.646	0.737	0.778

Neural Networks - BERT



Hyperparameters

1. Pre-trained BERT model
   a. 3 Models
2. Dropout rate
   a. 0.1 to 06
3. Class weight
   a. Class 0: 1.0
   b. Class 1: 1.0 to 3.0

Neural Networks - BERT results

- Best Parameters:
  - Pre-trained Model: small_bert/bert_en_uncased_L-4_H-512_A-8
  - Dropout rate: 0.2
  - Class Weight: {0: 1.0, 1: 2.0}

model	dropout_r	class_0	class_1	recall	precision	accuracy	f1
0	0.1	1	1	0.722	0.953	0.984	0.646
1	0.1	1	1.5	0.789	0.912	0.986	0.656
2	0.1	1	2	0.789	0.903	0.985	0.672
3	0.1	1	3	0.735	0.965	0.985	0.629
4	0.1	1	5	0.717	0.976	0.985	0.623
5	0.2	1	1	0.785	0.951	0.987	0.679
6	0.2	1	1.5	0.812	0.923	0.987	0.691
7	0.2	1	2	0.821	0.943	0.989	0.694

Model Results

Model	Precision	Recall	Accuracy	F1-Score
Top n-words Logistics Regression	0.726	0.839	0.979	0.778
CountVectorizer Logistics Regression	0.749	0.988	0.988	0.852
LSTM	0.637	0.953	0.980	0.763
BERT	0.943	0.821	0.989	0.694

Improvement with Traditional Model

- **Created a balanced dataset**
  - **Regular: 1000**
  - **Scam: 866**
- CountVectorizer Fit_transform to the sampled training and transform the testing data.
- Applied the dataset to various kind of models.

	Accuracy	F1_score	Precision	Recall
LogisticRegression() _ Train Details	100.0	100.0	100.0	100.0
LogisticRegression() _ Test Details	91.863	91.593	90.393	92.825
KNeighborsClassifier() _ Train Details	74.196	77.592	64.566	97.201
KNeighborsClassifier() _ Test Details	71.306	76.325	62.974	96.861
SVC() _ Train Details	97.212	97.002	95.897	98.134
SVC() _ Test Details	90.15	89.64	90.045	89.238
DecisionTreeClassifier() _ Train Details	100.0	100.0	100.0	100.0
DecisionTreeClassifier() _ Test Details	85.439	85.153	82.979	87.444
RandomForestClassifier() _ Train Details	100.0	100.0	100.0	100.0
RandomForestClassifier() _ Test Details	91.863	91.284	93.427	89.238
MultinomialNB() _ Train Details	96.212	95.875	95.95	95.801
MultinomialNB() _ Test Details	89.507	89.087	88.496	89.686

Improvement with Traditional Model

- Hyperparameter Tuning:

  Using Grid Search and Cross Validation to find the best parameters for logistic regression, and random forests

- ROC threshold

	Accuracy	F1_score	Precision	Recall
Logistic Regression() _ Train Details	0.981	0.979	0.976	0.983
Logistic Regression() _ Test Details	0.931	0.927	0.914	0.94
RandomForestClassifer() _ Train Details	0.987	0.986	0.976	0.997
RandomForestClassifer() _ Test Details	0.944	0.939	0.957	0.921

- **Logistic Regression:**
  - Best Parameters: {'C': 0.01, 'penalty': 'l2', 'solver': 'liblinear'}
- **Random Forest:**
  - Best Parameters: {'max_depth': 30, 'min_samples_split': 10, 'n_estimators': 50}

Neural Networks - Limitations

- Limited computational resources for BERT:
  - 1 hour/epoch locally
  - 15 min/epoch on Google Colab with GPU but with limitation
  - Only tested 8 sets of parameters
- More data and more complex structure is required
  - BERT reduces false positives
  - LSTM reduces false negative
  - Cannot achieve both in a single model
  - Need more complex structure to capture more relationships and more data to train the model.

Challenge, Impact and Evaluation

- Imbalance raw data create difficulty to train models.
- The choice to prioritize either recall and/or precision?
- More computational resources are required to speed up the training process
- More data is required to fine tune the model
- NN model still need more training/hyperparameter tuning
- Would apply to a certain range of job postings, but may be limited in impact.
- The format and content of online job postings have changed over the years,

Q&A

Model Results

Model	Precision	Recall	Accuracy	F1-Score
Top n-words Logistics Regression	0.726	0.839	0.979	0.778
CountVectorizer Logistics Regression	0.749	0.988	0.988	0.852
LSTM	0.637	0.953	0.980	0.763
BERT	0.943	0.821	0.989	0.694