CS 657 Massive Mining Datasets Assignment -2

Sai Sahana Bhargavi - G01330358

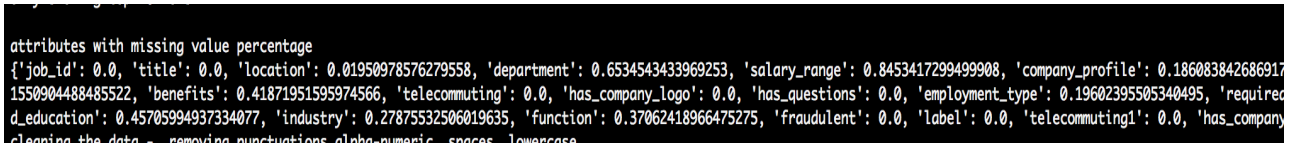
Introduction: The task is to create a system that automatically flags suspicious job postings.The dataset contains 2020 US Election Tweets Analysis of hashtag_joebiden.csv 18K job descriptions out of which about 800 are fake. The data consists of both textual information and meta-information about the jobs.

Preprocessing Data :

- Created spark session to read the data from the input file into a dataframe

```
spark = SparkSession.builder.appName("Predicting Fake Job Postings").getOrCreate()
```

- Encoded the label- “fraudulent” column using StringIndexer() and deleted the invalid data in the label fraudulent values either than 0 or 1 using filter().
- Computed the missing null values percentage of each attribute and dropped the columns that are having more than 1 % of missing null values.



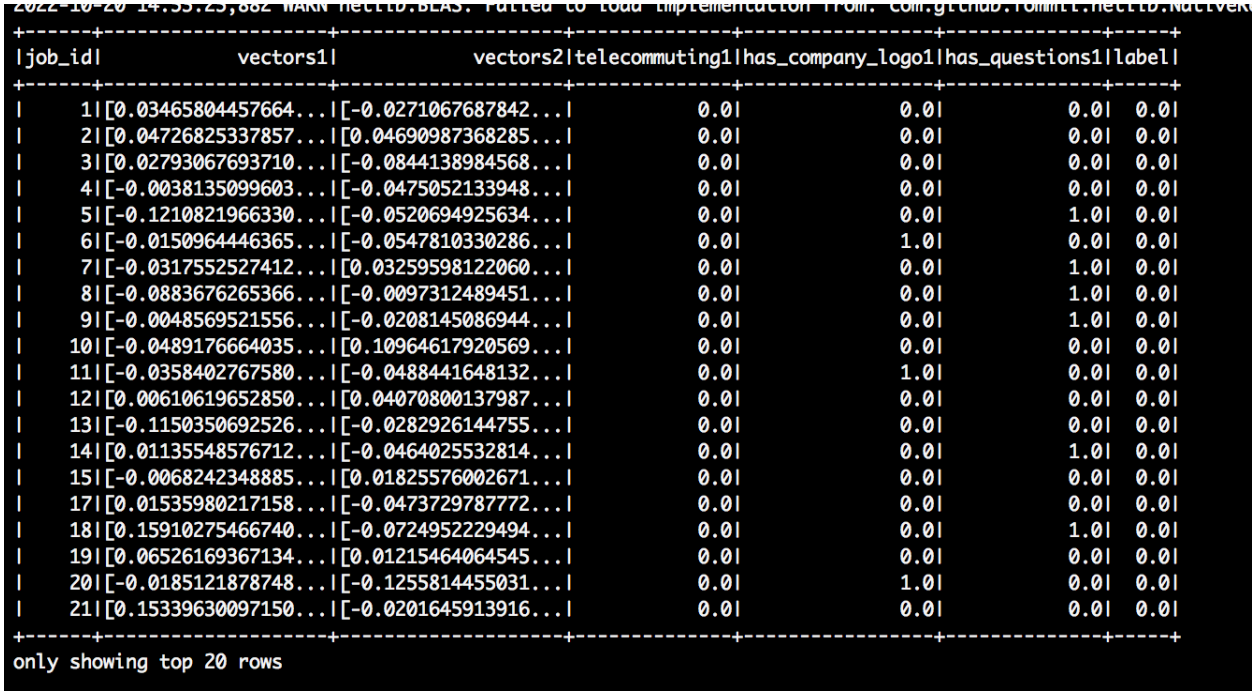
- Cleaned the data present in ‘title’,’description’ columns by removing punctuation marks, alpha-numeric characters, spaces and lowering the cases using regexp_replace() included with parameters for respective functionalities.

cleaning the data - removing punctuations,alpha-numeric, spaces, lowercase

job_id	telecommuting1	has_company_logo1	has_questions1	label	text	text1
1	0.0	0.0	0.0	0.0	marketing intern	food52 a fast gro...
2	0.0	0.0	0.0	0.0	customer service ...	organised focused...
3	0.0	0.0	0.0	0.0	commissioning mac...	our client locate...
4	0.0	0.0	0.0	0.0	account executive...	the company esri ...
5	0.0	0.0	1.0	0.0	bill review manager	job title itemiza...
6	0.0	1.0	0.0	0.0	accounting clerk	job overviewapex ...
7	0.0	0.0	1.0	0.0	head of content m f	your responsibili...
8	0.0	0.0	1.0	0.0	lead guest servic...	who is airenvy he...
9	0.0	0.0	1.0	0.0	hp bsm sme	implementation co...
10	0.0	0.0	0.0	0.0	customer service ...	the customer serv...
11	0.0	1.0	0.0	0.0	asp net developer...	position url_86fd...
12	0.0	0.0	0.0	0.0	talent sourcer 6 ...	transferwise is t...
13	0.0	0.0	0.0	0.0	applications deve...	the applications ...
14	0.0	0.0	1.0	0.0	installers	event industry in...
15	0.0	0.0	0.0	0.0	account executive...	are you intereste...
17	0.0	0.0	0.0	0.0	hands on qa leader	we are looking fo...
18	0.0	0.0	1.0	0.0	southend on sea t...	government fundin...
19	0.0	0.0	0.0	0.0	visual designer	kettle is hiring ...
20	0.0	1.0	0.0	0.0	process controls ...	experienced proce...
21	0.0	0.0	0.0	0.0	marketing assistant	intellibright is ...

only showing top 20 rows

- Converted the processed text in 'title','description' by breaking it down into words using feature extractor Tokenizer() , removed stop words using StopWordsRemover() and converted into vector format using feature transformer Word2Vec().



```
2022-10-20 14:55:23,682 WARN netlib.BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeR
+------+--------------------+--------------------+-------------+----------------+-------------+-----+
|job_id|            vectors1|            vectors2|telecommuting1|has_company_logo1|has_questions1|label|
+------+--------------------+--------------------+-------------+----------------+-------------+-----+
|     1|[0.03465804457664...|[-0.0271067687842...|          0.0|             0.0|          0.0|  0.0|
|     2|[0.04726825337857...|[0.04690987368285...|          0.0|             0.0|          0.0|  0.0|
|     3|[0.02793067693710...|[-0.0844138984568...|          0.0|             0.0|          0.0|  0.0|
|     4|[-0.0038135099603...|[-0.0475052133948...|          0.0|             0.0|          0.0|  0.0|
|     5|[-0.1210821966330...|[-0.0520694925634...|          0.0|             0.0|          1.0|  0.0|
|     6|[-0.0150964446365...|[-0.0547810330286...|          0.0|             1.0|          0.0|  0.0|
|     7|[-0.0317552527412...|[0.03259598122060...|          0.0|             0.0|          1.0|  0.0|
|     8|[-0.0883676265366...|[-0.0097312489451...|          0.0|             0.0|          1.0|  0.0|
|     9|[-0.0048569521556...|[-0.0208145086944...|          0.0|             0.0|          1.0|  0.0|
|    10|[-0.0489176664035...|[0.10964617920569...|          0.0|             0.0|          0.0|  0.0|
|    11|[-0.0358402767580...|[-0.0488441648132...|          0.0|             1.0|          0.0|  0.0|
|    12|[0.00610619652850...|[0.04070800137987...|          0.0|             0.0|          0.0|  0.0|
|    13|[-0.1150350692526...|[-0.0282926144755...|          0.0|             0.0|          0.0|  0.0|
|    14|[0.01135548576712...|[-0.0464025532814...|          0.0|             0.0|          1.0|  0.0|
|    15|[-0.0068242348885...|[0.01825576002671...|          0.0|             0.0|          0.0|  0.0|
|    17|[0.01535980217158...|[-0.0473729787772...|          0.0|             0.0|          0.0|  0.0|
|    18|[0.15910275466740...|[-0.0724952229494...|          0.0|             0.0|          1.0|  0.0|
|    19|[0.06526169367134...|[0.01215464064545...|          0.0|             0.0|          0.0|  0.0|
|    20|[-0.0185121878748...|[-0.1255814455031...|          0.0|             1.0|          0.0|  0.0|
|    21|[0.15339630097150...|[-0.0201645913916...|          0.0|             0.0|          0.0|  0.0|
+------+--------------------+--------------------+-------------+----------------+-------------+-----+
only showing top 20 rows
```

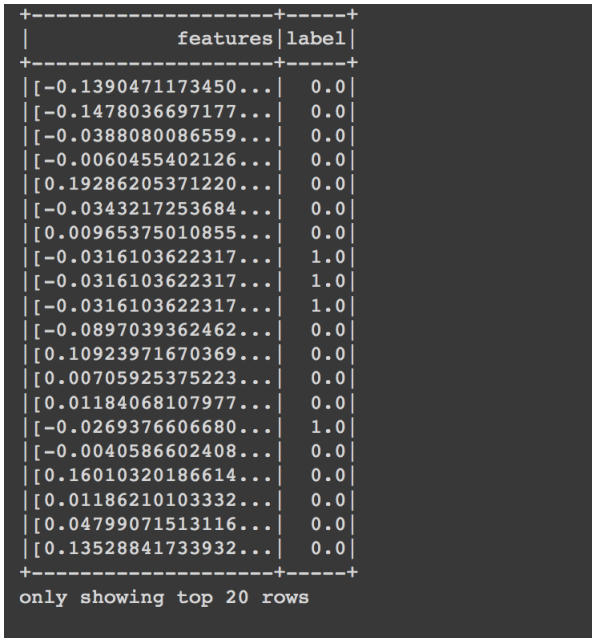
- As the data is highly balanced, the majority records, here 0's are undersampled -reduced to the number equivalent to minority records 1's using sampleBy().
- Converted the set of features into a single vector list using VectorAssembler () and transformed into two columns "features", "label".
- Performed a random split (70%,30%) of the data into training and test using randomSplit().

CrossValidation:

- K-fold cross validation performs model selection by splitting the dataset into a set of non-overlapping randomly partitioned folds which are used as separate training and test datasets.with k=10 folds, K-fold cross validation will generate 10 (training, test) dataset pairs, each of which uses 9/10 of the data for training and 1/10 for testing. Each fold is used as the test set exactly once.

  CV=CrossValidator(estimator=classifier,estimatorParamMaps=paramGrid, evaluator=evaluator,numFolds=folds)

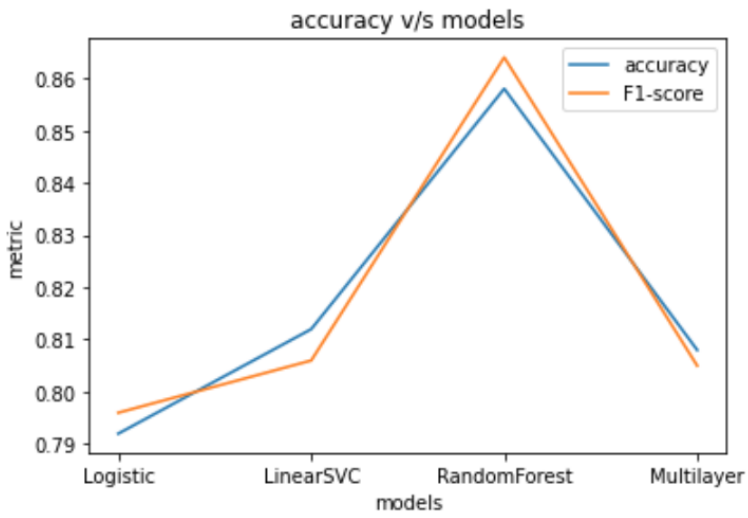
  fitModel = CV.fit(train)



```
+--------------------+-----+
|            features|label|
+--------------------+-----+
|[-0.1390471173450...|  0.0|
|[-0.1478036697177...|  0.0|
|[-0.0388080086559...|  0.0|
|[-0.0060455402126...|  0.0|
|[0.19286205371220...|  0.0|
|[-0.0343217253684...|  0.0|
|[0.00965375010855...|  0.0|
|[-0.0316103622317...|  1.0|
|[-0.0316103622317...|  1.0|
|[-0.0316103622317...|  1.0|
|[-0.0897039362462...|  0.0|
|[0.10923971670369...|  0.0|
|[0.00705925375223...|  0.0|
|[0.01184068107977...|  0.0|
|[-0.0269376606680...|  1.0|
|[-0.0040586602408...|  0.0|
|[0.16010320186614...|  0.0|
|[0.01186210103332...|  0.0|
|[0.04799071513116...|  0.0|
|[0.13528841733932...|  0.0|
+--------------------+-----+
only showing top 20 rows
```

-
- CrossValidator() performs cross validation for the given 'classifier' model using the corresponding parameters given in 'paramGrid' and here,evaluates using BinaryClassificationEvaluator() and further it fits the training data to the model and returns the model tuned with respective parameters.
- Here, we are using the classifiers as LogisticRegression, RandomForestClassifier, LinearSVC and MultilayerPerceptronClassifier.
- Here, The best parameters of each model are retrieved using fitModel.bestModel and the specific parameters using getMaxIter() for logistic regression, LinearSVC, MultilayerPerceptron Classifier and getMaxDepth() for RandomForestClassifier.
- After the above cross-validation procedure, the tuned model is used to predict the test data and accuracies, F1-score are determined for each classifier.

Classifier	**accuracy**	**F1-score**	**Best-Parameter**
LogisticRegression	0.834	0.842	MaxIterations= 10
LinearSVC	0.831	0.843	MaxIterations= 15
RandomForestClassifier	0.858	0.864	MaxDepth=10
MultilayerPerceptronClassifier	0.808	0.808	MaxIterations=100



Conclusion: This assignment helped us to learn how to deal with missing attribute columns and how the textual-data is pre-processed and especially dealing with Imbalanced data using sampling and how the different classifier models can be tuned with cross-validation technique and to get improved accuracy and F1-score.From the mentioned observations, among the tested classifier models, Random Forest performs well with highest accuracy as 0.858 and F1-score as 0.864.This gave an insight on utilizing the Pyspark ML libraries.