NLP Report

Prepared by Team 4 (AI):

- Mark Yousri Mounir (2020170131)
- Abdelrahman Maged Zaki (20201701722)
- Adham Khaled Ali (20201701704)

First: Data Preprocessing:

The default pipeline for the clean method is the following:

- fillna(s) Replace not assigned values with empty spaces.
- lowercase(s) Lowercase all text.
- remove_digits() Remove all blocks of digits.
- remove_punctuation() Remove all string.punctuation (!"#$%&'()*+,-./:;<=>?@[]^_`{|}~).
- remove_diacritics() Remove all accents from strings.
- remove_stopwords() Remove all stop words.
- remove_whitespace() Remove all white space between words.

The Hero Library was used for the data cleaning process.

The following columns were dropped due to high number of null values:

- salary offered for the job
- department
- benefits
- experience required

Stop words were removed using the hero.top_words() to count how much certain words were used, in conjunction with hero.clean to remove those words.

Categorical columns were label encoded using the sklearn library.

Nulls were filled using KNN Imputers instead of filling it manually with mean and median methods.

nltk.tokenize library was used to tokenize the words for lemming process.

nltk.stem library was used in conjunction with the tokenized words to lemmatize them.

Second: Feature Extraction:

TFIDF was used to check the frequency of words and determine how relevant they are in a document.

nGrams was used to extract sequence of words that can be relevant.

Third: Models Used:

- Logistic Regression
- AdaBoost
- Multilayer Perceptron
- XGBoost

Forth: Hyperparameter Tuning:

For Logistic Regression: after trial and error it was concluded that the following hyperparameters were optimal for the current case.

(max_iter=3000, random_state=83)

For AdaBoost: the n_estimators were tuned and tested and results shows that 60 strikes good balance between validation and training accuracy. (n_estimators=60)

For MultiLayer Perceptron: three hidden layers were used and after some trial and error we found great accuracy using the following layer sizes.

(20, 5, 30)

For XGBoost: a choice of 50 estimators and a max depth of 5 achieved the best accuracy out of all models.
(n_estimators = 50 , max_depth = 5)

Fifth: Results

Logistic Regression:

	Precision	Recall	F1score	Support
0	0.95	1.00	0.97	4245
1	0.85	0.09	0.17	250
Accuracy			0.95	4495
Macroavg	0.90	0.55	0.57	4495
Weightedavg	0.94	0.95	0.93	4495

MLP:

	Precision	Recall	F1score	Support
0	0.96	0.99	0.98	4245
1	0.68	0.33	0.45	250
Accuracy			0.95	4495
Macroavg	0.82	0.66	0.71	4495
Weightedavg	0.95	0.95	0.95	4495

XGBoost:

	Precision	Recall	F1score	Support
0	0.98	1.00	0.99	4245
1	0.94	0.63	0.75	250
Accuracy			0.98	4495
Macroavg	0.96	0.81	0.87	4495
Weightedavg	0.98	0.98	0.97	4495

Sixth: Final Test Accuracy

Logistic Regression: 92.34%

AdaBoost: 94.78%

Multilayer Perceptron: 80.55%

XGBoost: 94.81%