**Resume Screening/Parsing**

**Aim -** Understanding the implementation of machine learning algorithm and NLP (Natural Processing Language) using python

**Business Problem-** Companies often receive thousands of resumes for each job posting and employ dedicated screening officers to screen qualified candidates. Therefore, Screening and parsing the resumes of the candidate/s could help companies/hiring mangers in effective and time-saving selection process for interviews.

**Learning Outcomes-** Exercise helps candidates to understand the end-to-end process of transfer learning, natural language processing and prediction modeling implementation on image data, data augmentation.

**Initial Skill Requirements-** HR domain Knowledge, NLP, Python Machine learning algorithm to test & train the data.

**Data Information-** Here, we have taken the data in a CSV format which has categories and resume skills required.

***Steps involved***

**Step 0**: Domain knowledge and data information understanding

Description**-** Domain knowledge is the initial key to understand the problem and figure out the initial data required for solving the problem statement.

**Step 1**: Install and Load all required python libraries (for e.g., import NumPy as np, import pandas as pd, import seaborn as sns, import nltk and so on).

**Step 2**: Performing Exploratory Data Analysis

Description- The step helps to know the data better and have a visual representation of it.

Steps to follow-

* Check the categories of resumes in dataset.
* Have a look at the distinct categories of resume and the number of records belonging to each category
* Visualize the number of categories in the dataset.
* Check the distribution of categories.

Expected Outcome- EDA should give the clear idea about the data.

**Step 3**: Cleaning the data by creating a function.

Description- Function will remove the URLs, hashtags, mentions, special letters, and punctuations.

Steps to follow-

* Import re.
* Use the sub function.

Expected Outcome- It will clean the dataset by removing un-necessary fields.

**Step 4**: Creating a word-cloud using the WordCloud library.

Description- Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud

Steps to follow-

* Import nltk & wordcloud- the necessary libraries.
* Select the text & amount of text as stopwords, and create an array.
* Create a loop that will clean and check the tokenize words in the range iterations.
* Use frequency distributions to count number of times the outcome occurs.
* Build the wordcloud.

Expected Outcome- WordCloud should be built.

**Step 5**: Pre-processing data & Encoding Category

Description- we will work with the preprocessed and converted the ‘cleaned resume’ column into vectors. We will encode the ‘Category’ column using Label Encoding. Even though the ‘Category’ column is ‘Nominal’ data we are using LabelEnconding because the ‘Category’ column is our ‘target’ column.

Expected Outcome- By performing LabelEncoding each category will become a class and we will be building a multiclass classification model.

**Step 5**: Building Model and displaying classification report.

Description- We will be using the ‘One vs Rest’ method with ‘KNeighborsClassifier’, random forest and ANN to build this multiclass classification model.

Steps to follow-

* Check the library/package imports.
* Split the data into test & train data sets.
* Use the OneVsRest method with KNeighborsClassifier and display report.
* Use the OneVsRest method with Random Forest and display report.
* Use the OneVsRest method with ANN and display report.

Expected Outcome- Performance classification report will be displayed.

**Step 6**: Resume Parsing

Description- we will be importing one/multiple resumes and extract the summarized information into a data frame.

Steps to follow-

* Check the library/package imports.
* Import the files.
* Use the OneVsRest method with KNeighborsClassifier and display report.
* Use the OneVsRest method with Random Forest and display report.
* Use the OneVsRest method with ANN and display report.

Expected Outcome- Performance classification report will be displayed.

Steps For model making in ANN-

1. Data Preprocessing
   1. Import the Libraries-
   2. Load the Dataset
   3. Split Dataset into X and Y
   4. Encode Categorical Data-
   5. Split the X and Y Dataset into the Training set and Test set
   6. Perform Feature Scaling
2. Build Artificial Neural Network
   1. Import the Keras libraries and packages
   2. Initialize the Artificial Neural Network
   3. Add the input layer and the first hidden layer
   4. Add the second hidden layer
   5. Add the output layer
3. Train the ANN
   1. Compile the ANN
   2. Fit the ANN to the Training set
4. Predict the Test Set Results-
5. Make the Confusion Matrix

Steps for model making in Random Forest-

* 1. Import the Libraries-
  2. Load the Dataset
  3. Split Dataset into X and Y
  4. Encode Categorical Data-
  5. Split the X and Y Dataset into the Training set and Test set
  6. Perform Feature Scaling
  7. Fit Random Forest regressor to the dataset
  8. Predicting a new result
  9. Visualizing the result

Steps for model making in knearestneighbor-

* The k-nearest neighbor algorithm is imported from the scikit-learn package.
* Create feature and target variables.
* Split data into training and test data.
* Generate a k-NN model using neighbors value.
* Train or fit the data into the model.
* Predict the future.