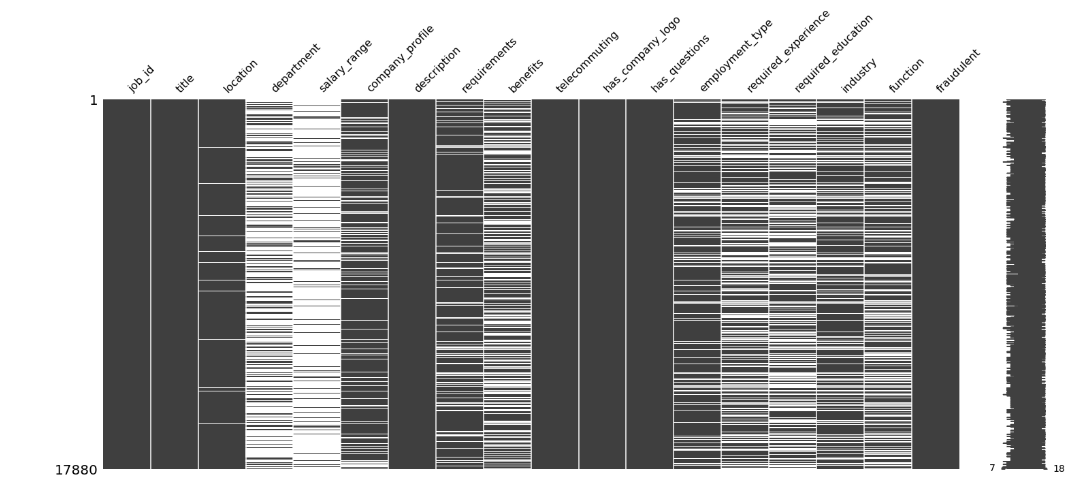
**Study on Kaggle Work**

For a comparison study, we have selected Kaggle work: [NLP(98%acc.) EDA with model using Spacy & Pipeline](https://www.kaggle.com/code/shivamburnwal/nlp-98-acc-eda-with-model-using-spacy-pipeline). This work is done by Shivam, Burnval. This code got 3rd most number of votes compared to 110 other Kaggle user works.

1. **Import Required libraries:** Author has imported necessary libraries to perform data analysis.
2. **Pre-Processing the data**:
3. Author has first identified the missing values using **missingno** library and plotted the result.
4. 
5. Author has identified the target variable = ‘**fraudulent’**
6. In the next step, all the Nan Values has been filled in required columns and unnecessary columns has been deleted from the dataset.
   1. A picture containing text

      Description automatically generated
7. As a part of cleaning the text data, Author had combined all the columns and stored in a new column with name: **data[‘text’]** and deleted all other columns which means a new data model was created with only 2 features
   1. Target feature
   2. All features combined together (data[text])
8. **Feature Extraction:**

* Author has plotted all the words in real jobs and fake jobs using word cloud and stop words.

Graphical user interface, text, application

Description automatically generated

Vii Example of one such plot is displayed below.

Text

Description automatically generated

1. **Cleaning the data model:**
2. 2 functions were created such that it accepts a sentence as input and processes the sentence into tokens, performing lemmatization, lowercasing, and removing stop words.

Graphical user interface, text, application, email

Description automatically generated

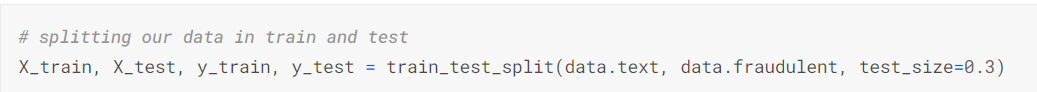
Graphical user interface, text, application

Description automatically generated

ii. After creating the functions, Author has performed cleaning the dataset.

1. **Training the model**:

Author utilized 30% of data for testing and 70% of data for training.



1. **Creating Classifier Models:**
2. Post creating the data Model, User has created Classifier model which accepts the input parameters of data model.
3. List of Classifier Models utilized were:
   1. **Logistic Regression:**
   2. **Random Forest Classifier**
   3. **Simple Vector Machine (SVM) Classifier**
   4. **XGBoost Classifier**
4. Screenshot below shows the implementation of classifier model by passing the data model using pipeline.

Text

Description automatically generated

1. **Evaluation Metrics:**
2. Author primarily used Accuracy and Recall as evaluation metrices for the classifier models. Screenshot below shows the Evaluation metrices.

Graphical user interface, text, application

Description automatically generated

1. **Visualization and interpretation:** User only used confusion matrix to plot the model evaluation scores.

**Graphical user interface

Description automatically generated**

**Comparative evaluation of Milestone-3 for Template Code:**

1. **Importing Required Libraries**: All the required libraries required to perform various data processing and analytical operations were imported.
2. **Pre-processing:**
3. **Checking for missing values**: Using SimpleImputer function on the numerical columns filled the missing values.
4. **Checking for duplicates**: All the duplicate rows that were present in dataset were dropped.
5. **Encoding categorical variables**: This involves converting categorical data into numerical values.
6. **Drop Columns with low variance**: A new function was created to delete feature which have low variance to avoid false predictions for machine learning algorithms.
7. **Clean\_dataset Function**: This function takes the DataFrame 'df' as input and returns a cleaned version of the DataFrame by removing rows with null, NaN, and infinity values.
8. **Exploratory Data Analysis:**
9. **Splitting data into training and testing sets:** In this project, we used 70% of the data for training and 30% of the data for testing.

Text

Description automatically generated

1. **3.2. Class distribution:** In this dataset, we have two classes - "**Real**" and "**Fake**". We can use a bar chart to visualize the class distribution.

Chart, bar chart

Description automatically generated

1. 3.3. **Correlation between features:** A correlation matrix is a table that displays the pairwise correlations between all the variables in a dataset.

Chart

Description automatically generated

1. **Skew of Univariate Distributions:** Skewness can be positive, indicating a longer tail on the right side of the distribution, or negative, indicating a longer tail on the left side.

Graphical user interface, diagram, engineering drawing

Description automatically generated

1. **Feature selection:** The template code aimed to perform feature selection using the SelectKBest method with **chi-squared (chi2) as the score function**. The result is the top 5 features selected by the SelectKBest method with chi2 as a score function and a bar chart was plotted.

Chart, bar chart

Description automatically generated

1. **Feature importance:** The code trains an **ExtraTreesClassifier model** to identify the **top 10 most important features** of a dataset.

Chart, bar chart

Description automatically generated

1. **Classifier Models:** The following were the list of classifiers for which the cross-validation technique is applied to perform model training and evaluation of each of the models.

* Linear Regression
* Logistic Regression
* k-Nearest Neighbors
* Gaussian Naive Bayes
* Perceptron
* Linear SVC
* Stochastic Gradient Descent
* Decision Tree
* Random Forest
* Linear Discriminant Analysis
* Ada Boost - Ensemble
* Gradient Boosting
* ExtraTrees
* Extreme Gradient Boosting

1. **Model Execution:** We used **cross-validation technique** which is a part of a model execution script, that executes all the classifier models. The model uses X, `training\_scores\_encoded` data with the defined `KFold` function as parameters.
2. **Model Execution Times:** Below image represents the execution time consumed by each of the classifiers for cross-validation technique.

Chart, bar chart

Description automatically generated

1. **Model Evaluation Metrics:**
2. **Accuracy:** Accuracy of all the classifier models has been calculated as below.

Table

Description automatically generated

1. **AUC Score:**

* AUC Score for Extreme Gradient Boosting is 0.99 &
* AUC Score for Extra Trees Classifier is: 0.99

1. **ROC Curve:** Below plots show the **ROC Curve of** **top 2 classifiers that got high accuracy**.

Graphical user interface

Description automatically generated

A picture containing graphical user interface

Description automatically generated

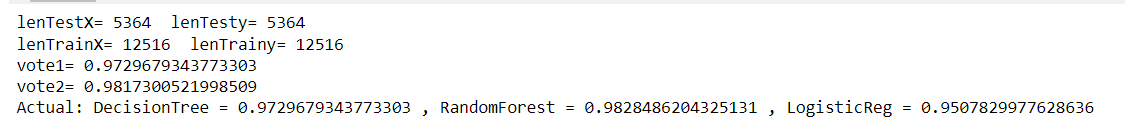
1. **Ensemble modeling using Max Vote:** It is a technique in which multiple models are trained and their predictions are combined to make the final decision.

* Decision Tree is plotted using Ensemble model voting classifier on decision tree classifier.

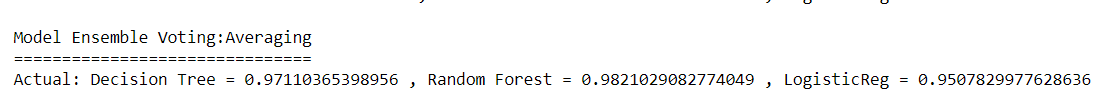
Chart, scatter chart

Description automatically generated

* Accuracy for Ensemble modelling max-vote is shown below.



* Accuracy for Ensemble modelling Averaging is shown below.



1. **Application of Deep learning method: LSTM**

* **LSTM Model Training Validations:** The plot in the left side represents the LSTM Model Accuracy on both test and train results, while the plot on the right represents the Model Loss for both Test and Train results.

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

* **LSTM Model Evaluation:** The evaluation metrics provided in the output are precision, recall, and F1-score, which are common measures used to evaluate the performance of a binary classification model

Chart

Description automatically generated

* **Confusion Matrix for LSTM Model:**

Chart

Description automatically generated

**Conclusion of Comparison Study:**

|  |  |  |
| --- | --- | --- |
| S.No | **Kaggle User's Efforts** | **Our Team's Efforts** |
| 1 | Author has prepared Data Model with only target feature and all other features merged to a single feature | Strict Data Pre-processing using multiple functions |
| 2 | Feature Selection and Feature importance is not properly processed | Feature selection and Feature importance is achieved by chi squared as score function |
| 3 | Pre-processing of Numerical Columns is not clear | SimpleImputer function is used to impute all numerical columns |
| 4 | Very few classifiers models were used to evaluate the dataset | A wide range of classifier models were used to evaluate dataset |
| 5 | Author has created custom Classifier Model using Spacy and pipelines | We used standard Cross-Validation technique on classifier models |
| 6 | Achieved good accuracy in most classifiers | Achieved >95% accuracy score for most of the classifiers |
| 7 | Less interpretation of Model evaluation metrics | Wide range of model evaluation plots |
| 8 | Model Execution time was not calculated | Model execution Time was calculated and plotted |
| 9 | Model Evaluation Metrics were only Precision and Accuracy | Accuracy, Precision, F1 Score, Execution Time and Recall were used |
| 10 | AUC Score or ROC Curve is not plotted | AUC Score was calculated, plotted ROC Curve |
| 11 | Ensemble Modeling/ Deep learning methods were not used | Ensemble Modeling/ Deep learning methods were used |