Research on Employee Turnover, Twitter Sentimental analysis & Fake News Detection

DATA MINING & MACHINE LEARNING

Aniket Kutty Shetty
MSc. Data Analytics (2024-2025)
National College of Ireland
Dublin, Ireland
x23177861@student.ncirl.ie

So in this research we would be studying about different datasets such as Employee Turnover, Twitter Sentimental Analysis, Fake News detection. The objective of this research is to create prediction models that are unique for each dataset using machine learning techniques like Logistic Regression and Decision Tree classifiers. These models will help us to get crucial insights for misinformation avoidance, organizational management, and social sentiment monitoring.

Dataset 1 – Employee Turnover

In this data set we are going to study inadept about the employee turnover, and we will build a model that predicts whether the employee will leave the company, or no?
It will be great to understand the causes that lead to an employee's departure if we can anticipate when they are most likely to go. Increasing staff retention will be advantageous to the business since it takes time and money to identify, interview, and hire new professionals.

Dataset 2- Twitter Sentimental Analysis

Using the Model Twitter Sentiment analysis, we can monitor how consumers perceive brands, spot patterns in social media discussions, and get insight into the opinions of the general public on a variety of topics. We will be able to distinguish between a serious/negative tweet and a positive one.

Dataset 3: Fake News Detection

The goal of fake news detection is to locate bogus articles and WhatsApp forwards.
This model will help people who easily trust all kind's fake WhatsApp forwards they receive or news they might read. This approach aims to inform them about the news they are receiving because the fake news can easily influence the naïve minds.

How Datasets differ from each other?

Datasets on employee turnover include details on workers demographics, performance evaluations, working hours, promotion, department etc while the twitter sentimental analysis consists of tweets from social media which indicates the sentiments of the tweets whether positive or negative**.** On the other hand Fake News detection consists of the textual data from various news articles and a label which indicates whether the news is fake or true. The news includes headlines, body, URL's, metadata etc. The Nature of the all the three datasets are different from one another.
In employee turnover dataset, each variables was checked if they influence the target variable, removed outliers, duplicates etc. This dataset is highly imbalanced which makes it different than the other two datasets and using Smote technique we have balanced this dataset.
Whereas in Twitter dataset we made use of NLTK techniques to remove stop words, use of porter stemmer to do stemming of words to its root words and apply tokenisation these characteristics make this datasets stands out from out two.
In Fake news detection dataset, cleaning & transformation is performed using regex patterns to remove URL's, punctuations, special characters, spaces etc. We have made use of vectorisation to convert the news into numeric form.

- **Tabular Format For 7 Characteristics of each of The Three Datasets Employed.**

Characteristic	Dataset 1	Dataset 2	Dataset 3
1. No of Independent Variables	9	5	4
2. No of dependent Variables	1(left)	1 (tweets)	1 (label)
3. Number of Records	14999 rows, 10 columns	1600000 rows, 6 columns	44878 rows & 5 columns
4. Data types of combination			
a. Binary	'work_accident', 'left', 'promotion_last_5years'	'target' (converted into binary)	'label'
b. Nominal	'Department'	'user'	'subject'
c. Categorical	'salary'	None	None
d. Numerical	'satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours', 'time_spend_company'	'id', 'date' ,'flag'	'date'
e. Textual	None	'tweets'	'title', 'text'
5. Data Cleaning	Yes	Yes	Yes
a. Irrelevant Variables Removed	1 - 'satisfaction_level'	4 - 'id', 'user', 'flag' 'date'	3- 'title','subject','date'
b. Duplications Removed	3008	No Duplicates	208
c. Missing Values	No missing values	No missing values	No missing Values
d. Outliers Filtered	1 - "tenure"	No Outliers	No Outliers
6. Data Normalization	3 columns a) Feature enginerring on Department column b) Converted monthly_hours to binary. c) Encoded the categorical column "Salary" into ordinal numeric category	"tweets": a) Removal of All the non-alphanumeric, special characters. b) Removal of Stop words c) Tokenization d) Stemming e) Vectorizations to convert Tweets into numerical.	"text": a) Removal of All the non-alphanumeric, special characters from. b) Remove URls, Digits, Newline Characters, Punctuation c) Applied Vectorizations
7. Data balancing characteristics and splitting	"left" – Used **SMOTE** fo r balancing of the target c olumn "left" { Employee Stayed: 83.1 5%. (11428 rows) & Employees left : 16.85 %. (3571 rows) } Data Split 30% test Data 70% traindata	"tweets" The tweets are equally balanced. {Positive Tweets: 8000 00 Negative Tweets: 8000 00} Data Split 30% test Dat a 70% train data	"text" The news data are somewhat balanced {True News 21417 Fake News – 23481} Data Split 30% test Data 70% train data

DATASET 1: Employee Turnover

- **Detailed Explanation**

i) Number of Independent Variables – 9, Dependent variables – 1

ii) Number of records: 14999 Rows and 10 columns
In this total 3571 are the total employees who left and 11428 stayed with the company. This data is highly biased towards the employees who stayed. To deal with such imbalance data set we are using **SMOTE** for balancing the data.

iii) Data types of combination :
1) Binary – 'Work_accident', 'left', 'promotion_last_5years'
2) Nominal – Department'
3) Ordinal - 'salary'
4) Numerical – 'satisfaction_level', 'last_evaluation', 'number_project', 'average_montly_hours', 'time_spend_company'

```
Data columns (total 10 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   satisfaction_level     14999 non-null  float64
 1   last_evaluation        14999 non-null  float64
 2   number_project         14999 non-null  int64
 3   average_montly_hours   14999 non-null  int64
 4   time_spend_company     14999 non-null  int64
 5   Work_accident          14999 non-null  int64
 6   left                   14999 non-null  int64
 7   promotion_last_5years  14999 non-null  int64
 8   Department             14999 non-null  object
 9   salary                 14999 non-null  object
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

Fig above shows detailed information Data Types of the Columns:

iv) Summary of each variable: min, max, mean, median, and quartiles:

The below fig shows us the summary of each variable. The mean of Average monthly hours is 201 whereas minimum hour is 96 and maximum is 310. The mean of number of projects column is 3.8 i.e Average number of projects worked by employees is close to 4. Maximum is 7 and min is 2

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.021268
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.144281
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.000000
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.000000
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.000000
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.000000

v) Data Cleaning & Transformation:

In the dataset the pre-processing steps involved lot of data cleaning steps & transformations.
First, I started by checking if we had any null or missing values in our dataset luckily, we didn't have any missing values.

- Duplicate removal

I looked for duplicates in the data and found 3008 duplicates which is 20% of the data. The records when observed didn't seem to be a genuine entry because how likely is it that two workers self-reported matching answers in every column? The duplicate entries were dropped, and a new data Frame was created.

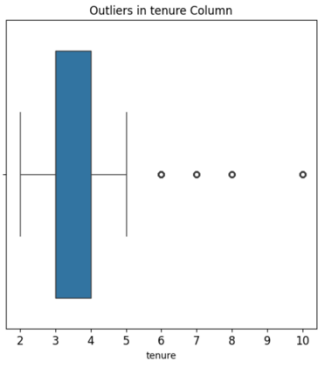
Lets Check for the duplicates in our datasets

```
# Check for duplicates
employeesData.duplicated().sum()
# lets examine a few rows that have duplicates.
```

```
3008
```

- Outlier's Detection and removals technique

When we plotted a boxplot to check for outliers, it was observed that tenure column had the most outliers and since the logistic regression is highly sensitive to the outliers, we removed the outliers from tenure column. The Number of data containing outliers in Tenure was 824.




```
Lower limit: 1.5
Upper limit: 5.5
The count of rows in the data that have tenure outliers 824
```

We made a use of IQR- Interquartile Range for removal of Outliers.
The Interquartile Range, or IQR, is a measure of statistical dispersion. It represents the range within which the middle 50% of the data falls. To calculate the IQR, you need to find the difference between the 75th percentile (Q3) and the 25th percentile (Q1).
IQR = Q3 — Q1.

We set two limits in order to use the IQR approach to find outliers.
Lower Limit -> Q1 – 1.5*IQR. We got 1.5.
Upper Limit -> Q3 + 1.5 * IQR we got 5.5

Any data point that was below the lower bound 1.5 were removed and the data points which were above the 5.5 upper limit was removed.

vi) Feature Engineering.

Before building the model, we encoded the non-numeric variables, as model can understand only numeric data.

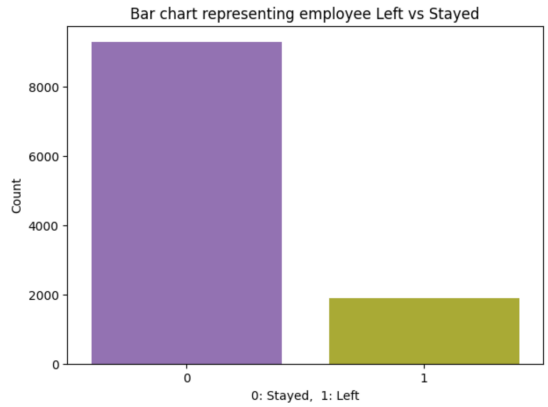
a) Since the **Department column** is categorical(nominal) we applied pandas get_dummies function to convert it into indicator in a binary for modelling.
b) **Salary** is ordinal i.e. low, medium, high and we converted the levels to numbers 0–2.
c) We also converted the average_monthly_hours which was numerical into binary format which will indicate whether the employee has overworked monthly.

The threshold value was considered as 175hrs. If the value was > 175hrs then overworked is True i.e 1 and value was <= 175 then overworked is False i.e 0.

d) Since the satisfaction_level had the lowest correlation with other independent and dependent columns we dropped the satisfcation_level column.

vii) Over Sampling using SMOTE

The below Chart represents Bar chart which shows employee left vs Stayed. It shows how our data is highly biased. We made a use of **SMOTE** technique to deal with such problem and balanced the data.



- **Model Building & its Metrics**

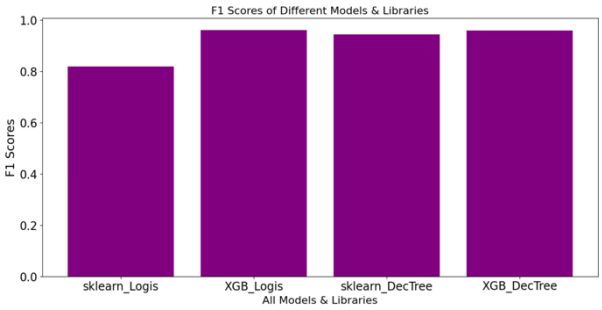
Data split - 30% Testing and 70% Training.

Logistic Regression using – sklearn Package.
Accuracy: 80.17%
F1 Score: 81.80%

Logistic Regression using – XGBoost Package
Accuracy: 96.07%
F1 Score: 96.00%

Decision Tree using – sklearn Package.
Accuracy: 94.63%
F1 Score: 94.46%

Decision Tree using – XGBoost Package
Accuracy: 96.00%
F1 Score: 95.93%



RESEARCH QUESTION 1

1) So, we can clearly observe that the performance measures of the both the models are **different**. When we use Logistic regression using sklearn package vs the XGBoost package, XGBoost Package outperforms in this case with accuracy of 96.07%.

2) Also, When we implement Decision tree and using Sklearn & XGboost's then there is slight difference In their accuracies and F1 scores of 1% or 2%. XGBoost gives the highest accuracy and F1 score wins in this case as well.

RESEARCH QUESTION 2

- The reason XGboost library is performing better when implementing Losistic regression is because**,**

- It uses gradient boosting techniques internally- (i.e it combines multiple weak models in sequence and train in sequencial order to correct the mistakes made by the previous models)
- XGboost has a better and more complex algorithm than traditional sklearn's logistic regression.
- XGboost can handle outliers and missing data more effectively.

➢ Although there is no major difference in the performance metrics of Decision tree while using Sklearn and XGboost but the slight difference is because,
- When XGboost is used instead of scikit-learn's default parameters, the model performs better since it makes use of hyperparameter optimization techniques like regularization and gradient boosting.
- XGBoost combines several weak learners (decision trees) into a single strong learner by applying ensemble techniques like gradient boosting.
- XGBoost's decision tree implementation provides enhanced feature importance analysis and can capture complex relationships between features more effectively than an sklearn.
- To prevent overfitting, the decision tree implementation in XGBoost usually uses pre-built regularization techniques like shrinkage and column subsampling. These regularization techniques may outperform sklearn's decision tree in terms of generalization, especially if the dataset is complex like ours or prone to overfitting.

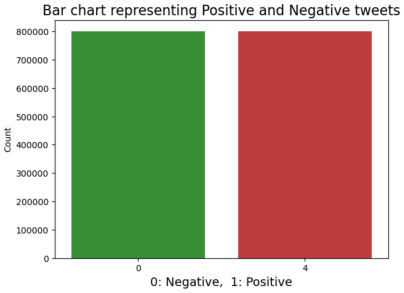
conclusion Dataset-1

Overall, when we compare all the above implementation for Dataset 1: Employee Turnover, when using XGBoost Library Decision tree gives performs better than other implementation. Another reason for us getting a good score is use of feature engineering. From the Visualisation done in the code we can say that employees are actually overworked here, and the recommendation would be to give employees fixed hours of work per month and that hours shouldn't be exceeded so that they have balanced life.
Another recommendation would be that employees who are working for at least 4 years should get promotion and investigation should take place why employees working for 4 years have low satisfaction level. The higher management should make discussion on companies work culture and make policies so that people get timely promotion on their evaluations basis, they should take measures on improving the satisfaction score of the employees and try to retain the employees.

DATASET 2: Twitter Sentimental Analysis.

i) Number of Independent Variables – 5, Dependent variables – 1

ii) Number of records: 1600000 rows, 6 columns
In this total 1600000 are the total tweets where 800000 is a Positive tweet and 800000 negative tweets



The Above Fig represents the quantity of positive and negative tweets. As we can see that class is equally divided. We don't need to perform any balancing.

iii) Datatypes
a. Binary: 'target' (converted into binary)
b. Nominal: 'user'
c. Ordinal: None
d. Numerical: 'id', 'date' ,'flag'
e. Textual: 'tweets'

iv) Summary of Data:

	target	id
count	1.600000e+06	1.600000e+06
mean	2.000000e+00	1.998818e+09
std	2.000001e+00	1.935761e+08
min	0.000000e+00	1.467810e+09
25%	0.000000e+00	1.956916e+09
50%	2.000000e+00	2.002102e+09
75%	4.000000e+00	2.177059e+09
max	4.000000e+00	2.329206e+09

There was no missing values and outliers in this data. We dropped the columns such as id', 'user', 'flag' 'date' which had no relation in prediction of the twitter sentiments.
We have made a use Natural Language Toolkit package to remove the stop words, performed stemming of words to its root words and tokenizing of the words.

➢ Model Building & its Metrics.

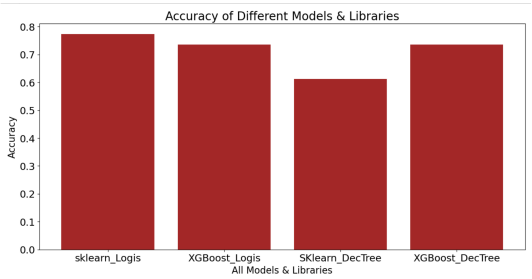
Data split - 30% Testing and 70% Training.

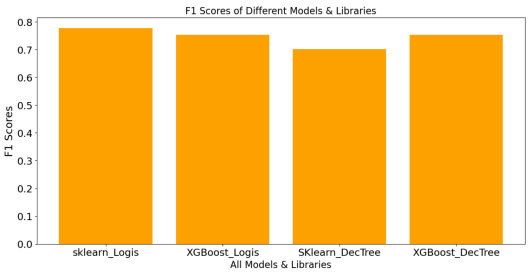
Logistic Regression using – sklearn Package.
Accuracy: 77.27%
F1 Score: 77.68%

Logistic Regression using – XGBoost Package.
Accuracy: 73.58%
F1 Score: 75.31%

Decision Tree using – sklearn Package.
Accuracy: 61.16%
F1 Score: 70.12%

Decision Tree using – XGBoost Package
Accuracy: 73.58%
F1 Score: 75.31%





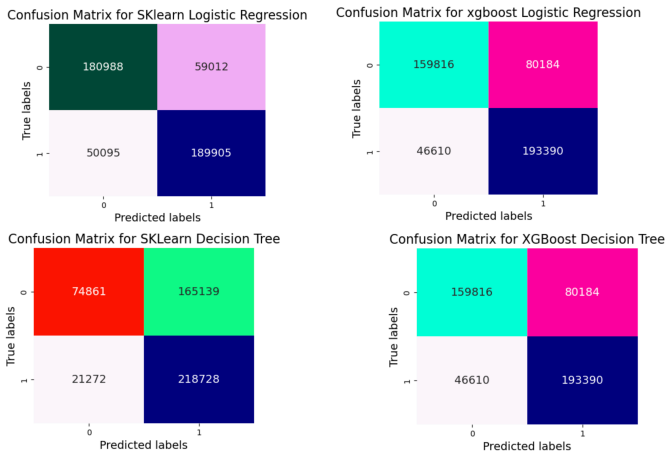
RESEARCH QUESTION 1

1) So, we can clearly observe that the performance measures of the both the models are different. The Above Figure shows the Bar chart for accuracies and F1 score. From the above metrics it is evident that Logistic Regression model using Sklearn library outperformed the library XGBoost with accuracy of 77.27%
2) On the other hand the 2nd algorithm i.e. Decision Tree when implemented with sklearn provide low accuracy of 61.16% compared to Decision Tree using XGBoost (accuracy -73.58%)

RESEARCH QUESTION 2

1) The reasons Logistic regression using sklearn performed better than XGboost package.
    - As we know that in twitter Sentimental analysis, we only need 2 columns i.e. 'tweets' and 'label'. XGBoost package is used for large-scale datasets and high-dimensional feature space where as Sklearn library is designed to be simple and suitable for data with less dimensions. This could be one reason Sklern's logistic regression outperformed xboost's logistic regression
    - Logistic regressions are less complex and effective when features are well selected & engineered compared with XGboost library.
    - The XGboost & Decision tree model's performance can be sensitive to the hyperparameter like Tree depth, learning rate, and the number of estimators. If it is not tuned properly the performance may be affected. While Logistic regression need less tuning to perform better.

2) The reasons decision tree using XGboost performed better than Sklearn because:
    - When compared to decision trees from Sklearn, this ensemble technique tends to reduce bias and variation and can result in enhanced prediction accuracy. XGBoost is a gradient boosting technique, which repeatedly creates an ensemble of weak learners (decision trees).
    - XGBoost provides greater hyperparameter tweaking freedom. Tree depth, learning rate, and the number of estimators are examples of hyperparameters that may be fine-tuned to greatly affect model performance. Compared to sklearn's Decision Tree, which has fewer hyperparameters to tune, XGBoost's capacity to efficiently adjust these hyperparameters may provide models that perform better.



The above figure shows us the confusion matrix for the all the models.

Conclusion - Twitter Sentimental analysis

After creating a Model on Twitter sentimental analysis, the Best performing algorithm is logistic model implemented using sklearn library with an accuracy of 77.27% and F1 score of 77.68%.

Thus, we can further improve the model performance and can use the following model for understanding the sentiments of the tweets which can help organisation monitor how consumers perceive brands, spot patterns in social media discussions and target right audience at the right times. Also this model can help understand human sentiments on election, including which party they prefer more etc. This can help parties to make improvements in their weaknesses and be a better political candidate for the public.

DATASET 3: Fake News Detection:

- Number of Independent Variables – 4, Dependent variables – 1
- Number of records: 44878 Rows and 5 columns
  In this total 21407 are the True news and 23471 Fake news and the data is balanced
- Data types of combination:
  Binary – 'label'
  Nominal – 'subject'
  Ordinal - None
  Numerical – 'date'
  Textual – 'title', 'text'

  There were no outliers and missing values In the data. We found 208 duplicates and we decided to drop it as it was small number.
  We have done data cleaning and transformation by replacing all the special characters, punctuations, spaces, URLs in the data.
  Finally we dropped the columns such as 'title', 'subject', 'date' which had no relation in prediction of the fake news vs true news. Then we converted the "texts" to numerical by using the vectorization method which is needed for model building

➢ Model Building & its Metrics

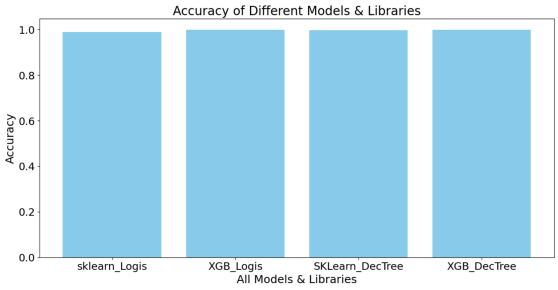
Data split - 30% Testing and 70% Training.

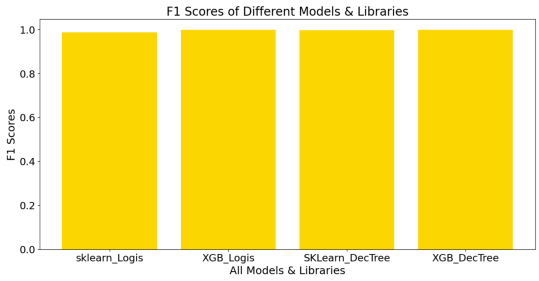
Logistic Regression using – sklearn Package.
Accuracy: 98.81%
F1 Score: 98.81%

Logistic Regression using – XGBoost Package.
Accuracy: 99.74%
F1 Score 99.74%

Decision Tree using – sklearn Package.
Accuracy: 99.59 %
F1 Score: 99.59 %

Decision Tree using – XGBoost Package
Accuracy: 99.74%
F1 Score: 99.74%





Research Question 1

After the evaluation we can conclude that all models performed almost **equally the same**. Using the sklearn or XGBoost package, both logistic regression and decision tree models obtained accuracy and F1 scores over 98%. All methods and implementations have given a high performance, as shown by the above bar chart evaluation metrics.

Research Question 2

The dataset was well-prepared, cleaned, balanced this is the reasons it led to consistent performance across different models. The reason could also be that dataset is not very complex and straightforward so the tasks for the logistic regression and decision trees models were not complicated. The hyperparameters were well tuned and the variations in the data sets were not much.

In conclusion this model will help prevent misinformation from reaching the naïve minds. This will keep integrity of the information and help build a more informed and discerning online community.