

# Indeed.com: Review Sentimental Analysis

Spring 2022

Instructor: Jingyi Sun

Members: Priya Gohil, Pawanneet Kaur, Viraj Jadhao, Sahil  
Mody

Group 10

Web Mining Final Report

## 1. INTRODUCTION:

Job searching is a time-consuming procedure for many people all around the world. When applying for a job, one must consider a variety of factors such as the work description, the satisfaction of present employees, the perks, the wage range, and so on. The goal of this project is to scrape the prominent Job-Hunting website "Indeed.com" based on the job designation to find mainly the list of companies, user ratings, reviews, and descriptions.

We have categorized the organizations depending on their field and domain, and we will assess existing employee work satisfaction based on the reviews scraped. The employee reviews posted on the job board were examined to derive useful insights and employee sentiments. This will assist individuals in picking the companies for which they desire to apply. We will do so by various EDA techniques ranging from Histogram, Pie charts, Bar plots and Word Cloud. Various Python libraries were used, including BeautifulSoup for Web Scraping; Seaborn and Matplotlib for EDA etc.

## 2. LITERATURE REVIEW:

Job board builders aren't the only ones that gain from data from job boards. Human resource experts, job seekers, to-be job hoppers, and recruitment and job market analysts are all keen for job data. If you're looking for employment, having a broad view of the market might help you negotiate more effectively. We investigated different references to understand the implementation of web mining and scrap indeed.com and thereafter implement the sentiment analysis on the reviews scraped.

- Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. *"Sarcasm as contrast between a positive sentiment and negative situation"*, In EMNLP 2013, pp. 704–714

- B. Agarwal, N. Mittal, "*Prominent Feature Extraction for Review Analysis: An Empirical Study*", In Journal of Experimental and theoretical Artificial Intelligence, 2014, DOI: 10.1080/0952813X.2014.977830
- M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, "*Lexicon-based methods for sentiment analysis*", Computational Linguistics, v.37 n.2, p.267-307, 2011

### **3. RESEARCH QUESTION:**

We've all experienced how difficult and stressful a job search can be. Indeed.com is a popular job-searching website. We aim to focus on popular jobs in the market through this project. The research part tries to comprehend and analyze the sentiments expressed by workers in job reviews. We will examine various work roles and assess the job satisfaction of various firms. This will give people a better idea of what the firm has to offer.

The source of our data will be Indeed.com. The research aspect aims to address the question, "How satisfied are the employees with the job?", "How can an employee use ratings and reviews of different companies to choose the next company he/she would want to work for?"

We will be looking at the different job positions (Pharmacist, Data Scientist) for different companies and determine which companies have more ratings and have an overall good or bad review in terms of these positions.

### **4. DATA COLLECTION:**





For the process of Data collection, we used the method of web scraping on the website - indeed.com. We focused mainly on the library - BeautifulSoup

to scrape the reviews of about 150 companies for the two designations, precisely, Data Scientist and Pharmacist.

The attributes scraped are - company name, designation, title, review, date, place and ratings

This scraped data of company reviews was used further to implement sentiment analysis. The two different types of pages were scraped, a list of companies falling under one position and the review page of each company showcasing the reviews description, ratings, date, place, designation, etc.

Page 1 scraped:

 [Find jobs](#) [Company reviews](#) [Find salaries](#)   

Company name or job title






City, state, or zip (optional)

Find Companies

data scientist

Los Angeles, CA

**Popular companies for data scientist in Los Angeles, CA**  
Based on reviews and recent job openings on Indeed

	Data Scientist	<a href="#">Reviews</a>	<a href="#">Salaries</a>	<a href="#">Jobs</a>
	<b>Meta</b> 4.1 ★ Arts, Entertainment & Recreation	<a href="#">Reviews</a>	<a href="#">Salaries</a>	<a href="#">Jobs</a>
	<b>Comcentric</b> 4.2 ★ Human Resources & Staffing	<a href="#">Reviews</a>	<a href="#">Salaries</a>	<a href="#">Jobs</a>
	<b>Deloitte</b> 4.0 ★ Management & Consulting	<a href="#">Reviews</a>	<a href="#">Salaries</a>	<a href="#">Jobs</a>
	<b>Accenture</b> 4.0 ★ Information Technology	<a href="#">Reviews</a>	<a href="#">Salaries</a>	<a href="#">Jobs</a>

We're a leading professional services firm, but what makes working at Deloitte unique? When you join us, you become connected. To inspiring people. To meaningful work with fascinating clients.

With 505,000 people serving clients in more than 120 countries, Accenture brings continuous innovation to help clients improve their performance and create lasting value across their enterprises.

Snapshot

Why Join Us

10.6K  
Reviews

112K  
Salaries

Benefits

90.5K  
Jobs

116  
Q&A

Interviews

79  
Photos

Follow

Reviews

Senior Technical Product Manager in New York, NY

5.0 ★ on April 18, 2022

**Great place to start your career as there are opportunities to learn and develop network.**

I have come across the most hard working and knowledgeable bunch of individuals and learned a great deal working side by side them. Folks are really smart and can articulate complex situations in a simple easy to understand language. It is a great place to work if you want to progress your career as a consultant. However, not the right place to advance product management career.

I am a Sr. Product Manager with more than seven years of experience. I was lucky to get great opportunities to express my passion for product management however from long term career perspective this is not the right place.

Consultant in Los Angeles, CA

4.0 ★ on May 5, 2022

**Good opportunities**


The work can be demanding, but great company to work for. There is a lot of travel involved, but they always provide plenty of notice with a predefined schedule.

Good work culture  
Training and improve your skills  
You get opportunity to work on different clients

What would you say about your employer?

Help fellow job seekers by sharing your unique experience.

Write a review



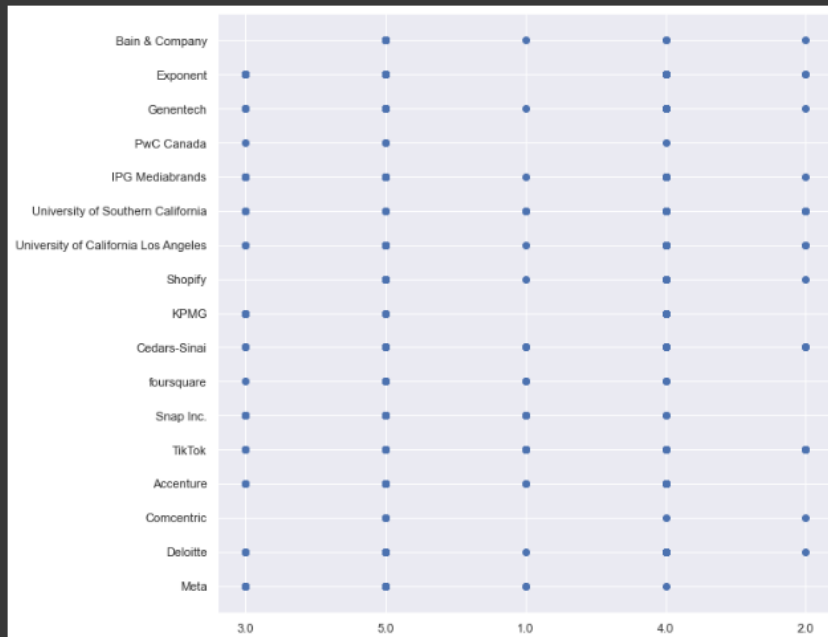
## 5. EXPLORATORY DATA ANALYSIS:

We attempted to create numerous plots for exploratory data analysis to efficiently analyze our data.

- Scatter Plot:

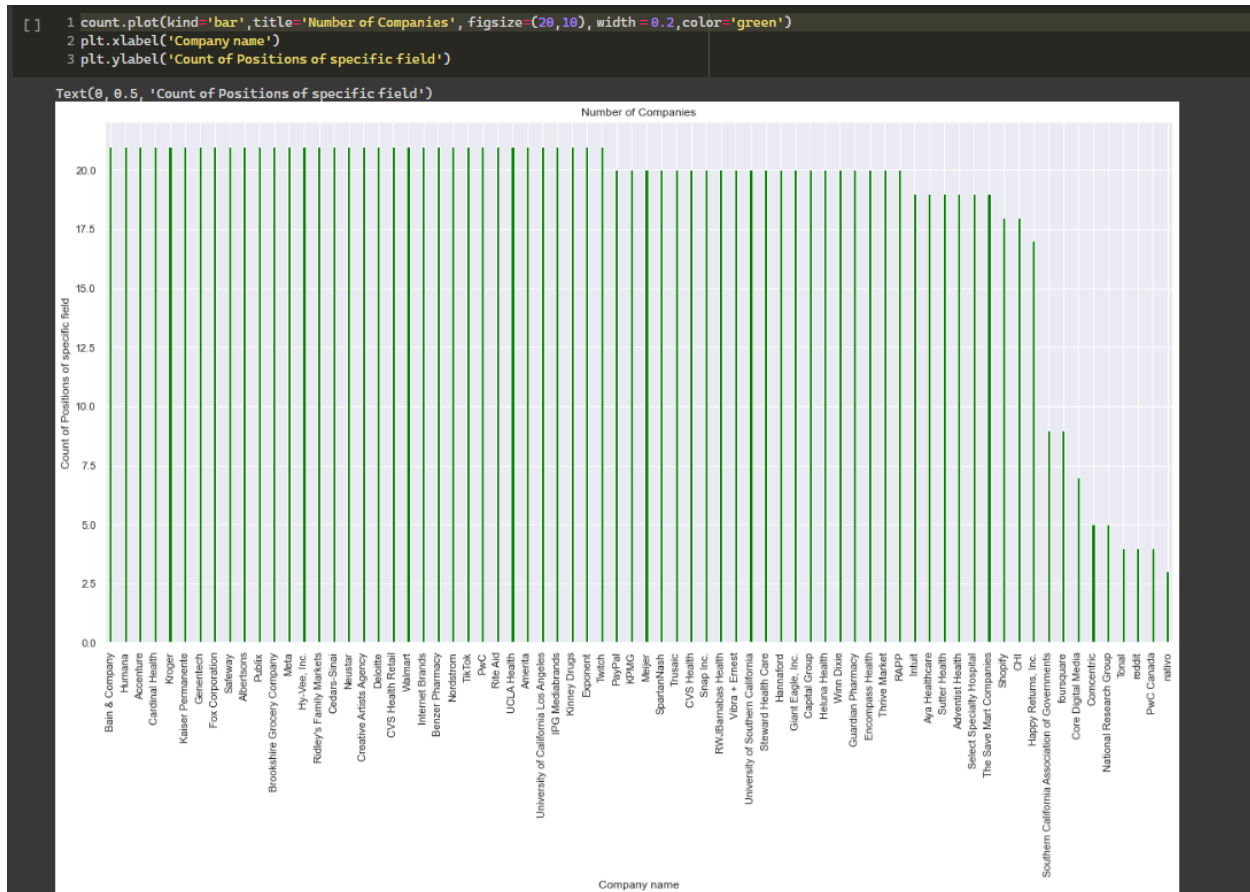
For the first part, we have made a Scatter Plot with the Ratings on the X-axis and names of the company on the Y-axis. From the Scatter Plot, we conclude that most of the Deloitte and Tik Tok have an above average rating while Shopify and KPMG, etc. have a low rating. Most of the IT companies have varied ratings from extremely low to extremely high. We have used the matplotlib library for this plot.

```
[ ] 1 import seaborn as sns
2 import matplotlib.pyplot as plt
3 import numpy as np
4 rating=[]
5 company=[]
6 for d in df['ratings']:
7     rating.append(d)
8 for e in df['company']:
9     company.append(e)
10 sns.set(color_codes=True)
11 plt.scatter(rating[1:300],company[1:300])
12 plt.show()
```



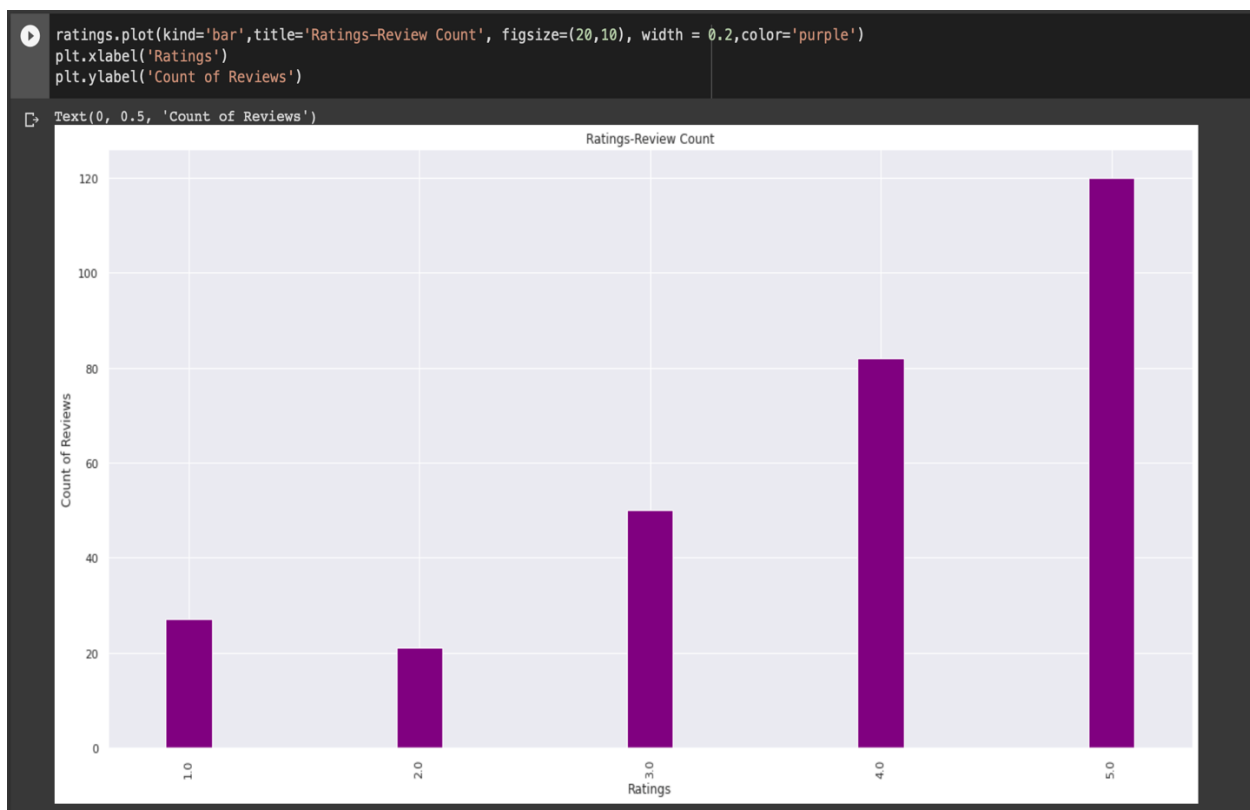
- Bar Plot (i):

The company name is on the x-axis, while the number of positions in a certain field is on the y-axis. This graph illustrates the number of available employment positions in each organization. After closely inspecting the plot, we can conclude that organizations such as Kroger, Meta, Tik Tok, PwC, and others have the most available jobs, i.e. 20-25 job opportunities per company. Companies such as KPMG, CVS, Shopify, and others have fewer job openings, ranging from 15 to 20. Nativo, on the other side, has the fewest available slots, namely three. For this sample, we used the matplotlib package.



- Bar Plot (ii):

The number of reviews is on the y-axis, while the ratings offered are on the x-axis in this graph. This graph depicts the number of reviews provided to a certain firm under a specific job satisfaction level. After extensively examining this chart, we can determine that individuals prefer to provide generally positive reviews rather than negative ones. Reviews with 5 stars are much more than reviews with a single star. This demonstrates that the majority of individuals are pleased with the work at hand.



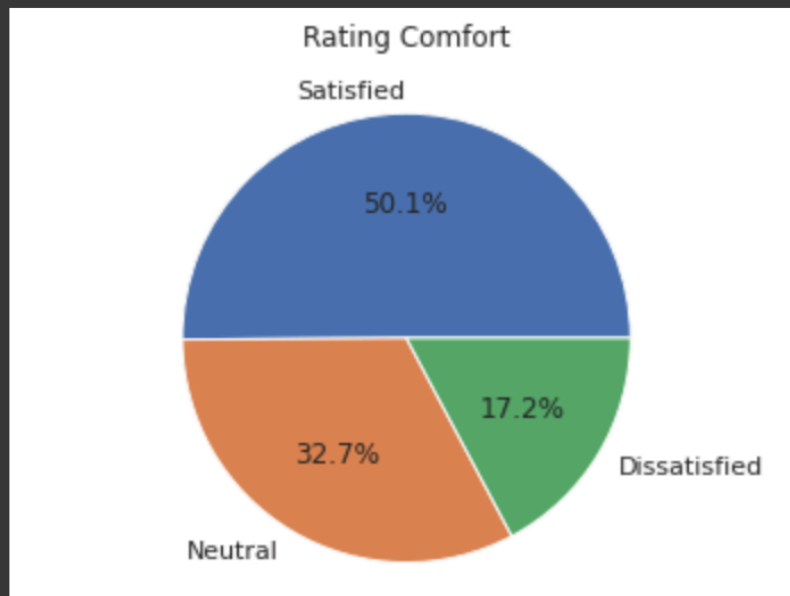
- Pie Chart:

Furthermore, we have plotted a Pie Chart to find out how many working adults are satisfied with the job that they have. From the plot we can see that the majority around 60% of the people have given a rating of above 3.5, which would mean they are satisfied. On the other hand, 10% of the people have given a rating of below 3 to their company, which would mean that they are extremely unsatisfied with their jobs. Again, we have used the matplotlib library of Python to plot this chart.





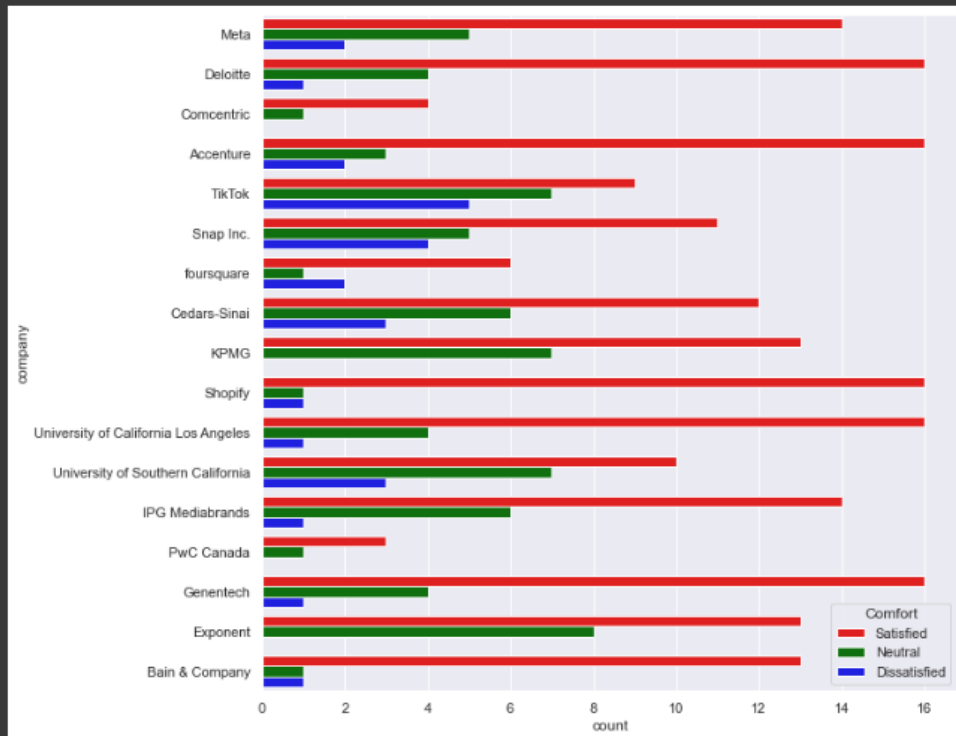
```
my_labels= ["Satisfied", "Neutral", "Dissatisfied"]  
plt.pie(train1, autopct='%1.1f%%', labels=my_labels)  
plt.title('Rating Comfort', pad=20)  
plt.axis('equal')  
plt.show()
```



- CounterPlot:

Next we have the counterplot, with the number of ratings obtained in each satisfaction category on the X-axis and the firm name on the Y-axis. To minimize data saturation, we only examined the top 20 firms with the most ratings in this graph. We look at how many people in a certain organization are pleased with their jobs. As a result, we can conclude that the majority of employees at organizations such as Deloitte, Accenture, Shopify, and others are happy with their careers. On the other hand Tik Tok has the highest number of unsatisfied employees. For this plot, we used the seaborn library.

```
[ ] 1 main_df=df[0:300]
2
3
4 import seaborn as sns
5 with sns.axes_style(style='whitegrid'):
6     sns.set(rc={'figure.figsize':(10,10)})
7     g = sns.countplot(y=main_df['company'],hue=main_df["Comfort"],
8                       data=main_df,palette=['Red', 'Green', 'Blue'])
```



- Word Cloud:

This word cloud is based on reviews submitted by workers of a certain organization. This word cloud highlights the most important talking topics for each organization. The larger the font size of the term, the more frequently that word is used while leaving a review. For example, we can see that the majority of individuals discuss the company's management, the workload of the job, other employees, the compensation provided, the work environment, and so on. This chart was created using the WordCloud library.

```
Text(0.5, 1.0, 'Trump Tweets Word Cloud')
```



- TextBlob helps the user by returning a polarity value.
- For our dataset, if the polarity value was less than the sentiment was classified as negative; if the polarity value was equal to zero, the sentiment was neutral and if the polarity was a positive value, the sentiment was assumed to be positive.

```
#functions that will assist in the sentiment analysis
def subjectivity(review):
    return TextBlob(review).sentiment.subjectivity
def polarity(review):
    return TextBlob(review).sentiment.polarity
def conclusion(val):
    if val<0:
        return 'negative'
    elif val==0:
        return 'neutral'
    else:
        return 'positive'
```

```
[ ] subjectivity_col = df1['review'].apply(subjectivity)
    polarity_col = df1['review'].apply(polarity)
    analysis_col = polarity_col.apply(conclusion)

df2 = {'Review': df1['review'], 'Subjectivity': subjectivity_col, 'Polarity': polarity_col, 'Sentiment': analysis_col}
sentiment_analysis = pd.DataFrame(df2)
sentiment_analysis.head()
```

	Review	Subjectivity	Polarity	Sentiment
0	Working at Facebook exposes you to problems se...	0.665344	0.166667	positive
1	I have joined the company since 2018 and enjoy...	0.472222	0.194444	positive
2	Great if you're looking for a big tech job, le...	0.347619	0.252381	positive
3	It is a great company to work for. I had a pre...	0.784694	0.352041	positive
4	I've been at Meta (previously Facebook) for a ...	0.367899	0.157681	positive

```
neg_num = sentiment_analysis[sentiment_analysis['Sentiment']=='negative'].Sentiment.count()
neu_num = sentiment_analysis[sentiment_analysis['Sentiment']=='neutral'].Sentiment.count()
pos_num = sentiment_analysis[sentiment_analysis['Sentiment']=='positive'].Sentiment.count()

print('Sentiment Breakdown: Company Reviews')
print('Negative Reviews: ', neg_num)
print('Neutral Reviews: ', neu_num)
print('Positive Reviews: ', pos_num)
```

```
☞ Sentiment Breakdown: Company Reviews
Negative Reviews: 250
Neutral Reviews: 41
Positive Reviews: 1040
```

## 7. MACHINE LEARNING MODELS:

- Logistic Regression:

```
[ ] 1 #Feature extraction, model selection and model training library
2 from sklearn.feature_extraction.text import HashingVectorizer
3 from sklearn.pipeline import Pipeline
4 from sklearn.linear_model import LogisticRegression
5 from sklearn.naive_bayes import GaussianNB
6 from sklearn.model_selection import train_test_split
7
8 #Libraries to check the model performance
9 from sklearn import metrics
10 from sklearn.metrics import classification_report
11 from sklearn.metrics import confusion_matrix

1 train_data = df['review']
2 targets = df['Sentiment']
3 X_train, X_test, Y_train, Y_test = train_test_split(train_data, targets, test_size=0.3)

[ ] 1 model = Pipeline([('vect', HashingVectorizer()),
2                       ('logreg', LogisticRegression(max_iter=1000, solver='saga'))
3                       ])
4 model.fit(X_train, Y_train)

Pipeline(steps=[('vect', HashingVectorizer()),
                 ('logreg', LogisticRegression(max_iter=1000, solver='saga'))])

[ ] 1 def check_model_metrics(model, test_data, test_targets):
2     y_pred = model.predict(test_data)
3
4     print("ACCURACY:")
5     print(metrics.accuracy_score(test_targets, y_pred)*100)
6
7     print("\nCONFUSION MATRIX")
8     print(confusion_matrix(test_targets, y_pred))
9
10    print("\nCLASSIFICATION REPORT")
11    print(classification_report(test_targets, y_pred))
```

```
[ ] 1 check_model_metrics(model, X_test, Y_test)
```

ACCURACY:  
80.0

CONFUSION MATRIX  
[[ 8 0 66]  
 [ 2 0 9]  
 [ 3 0 312]]

CLASSIFICATION REPORT

	precision	recall	f1-score	support
negative	0.62	0.11	0.18	74
neutral	0.00	0.00	0.00	11
positive	0.81	0.99	0.89	315
accuracy		0.80		400
macro avg	0.47	0.37	0.36	400
weighted avg	0.75	0.80	0.73	400

- Multinomial Naive Bayes:

Multinomial Naive Bayes algorithm is a probabilistic learning method that is mostly used in Natural Language Processing (NLP). It is popularly used for Text data classification, in this case, to classify whether a review was positive or negative.

This algorithm is based on the popular Bayes Theorem  $P(A|B) = P(A) * P(B|A)/P(B)$ .

One of the major advantages of this algorithm is that it is highly scalable and appropriate for large datasets. In this case, we decided to use Count Vectorizer and we noticed a significant change in the complexity of the program.

With the help of sklearn library, MultinomialNB, which has alpha as 1, also gave us an accuracy of 85.18%.

```
[ ] 1 from nltk.corpus import stopwords
    2 from nltk.stem.porter import PorterStemmer
    3 from sklearn.feature_extraction.text import CountVectorizer

[ ] 1
    2 def normalize_numbers(s):
    3     return re.sub(r'\b\d+\b', 'NUM', s)
    4
    5 cv = CountVectorizer(preprocessor=normalize_numbers, ngram_range=(1,3), stop_words='english')

▶ 1 X_train_counts = cv.fit_transform(X_train)
   2 X_test_counts = cv.transform(X_test)

[ ] 1 from sklearn.naive_bayes import MultinomialNB
    2 from sklearn import metrics
    3 clf = MultinomialNB().fit(X_train_counts, Y_train)
    4 predicted = clf.predict(X_test_counts)

[ ] 1 # print(pd.DataFrame({"Actual":y_test,"Predicted":predicted}))
    2 print(metrics.classification_report(Y_test, predicted))

      precision  recall  f1-score  support
negative      1.00    0.03    0.05       74
neutral       0.00    0.00    0.00        11
positive      0.79    1.00    0.88      315

accuracy              0.79    400
macro avg    0.60    0.34    0.31    400
weighted avg    0.81    0.79    0.71    400
```

- **Decision Tree Classifier:**

Decision tree algorithms fall under the category of supervised learning. They can be used to solve both regression and classification problems. The intuition behind Decision Trees is that you use the dataset features to create *yes/no* questions and continually split the dataset until you isolate all data points belonging to each class and with this process we're organizing the data in a tree structure.

In our program we have used a maximum depth of 120 and `min_samples_split` as 3 to receive an accuracy of 87%.

- The minimum number of samples required to split an internal node
- The maximum depth of the tree. If None, then nodes are expanded until all leaves are pure or until all leaves contain less than `min_samples_split` samples

```
1 from sklearn.tree import DecisionTreeClassifier
2 clf_gini = DecisionTreeClassifier(criterion = "gini",
3     random_state = 100, max_depth=3, min_samples_leaf=5)
4 clf_gini.fit(X_train_counts, Y_train)
5 pred = clf_gini.predict(X_test_counts)
6 print(metrics.classification_report(Y_test, pred))
```

	precision	recall	f1-score	support
negative	0.00	0.00	0.00	2
neutral	0.00	0.00	0.00	3
positive	0.91	0.98	0.94	49
accuracy			0.89	54
macro avg	0.30	0.33	0.31	54
weighted avg	0.82	0.89	0.85	54

```
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted s
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted s
_warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0 in labels with no predicted s
_warn_prf(average, modifier, msg_start, len(result))
```

## 8. CONCLUSION AND COMPARISON:

In this project, we have implemented Multinomial Naive Bayes, Decision Tree Classifier and Logistic Regression. Comparing these three algorithms, we have got the best accuracy with Decision Tree Classifier and the least accuracy with Logistic Regression. This is because the decision tree classifier

has helped us in removing the outliers. In addition to that, since our classification is categorical, the decision tree is bound to give better results than Logistic Regression. The decision tree also performed feature selection better than multinomial naive bayes, hence, superseded the accuracy.

## **9. LIMITATIONS AND FUTURE WORK:**

The list of companies and reviews are only restricted to a role that was scrapped, if a user tends to look for a review for a company with other designation, he/she might not end up getting results intended.

The dataset scrapped was precisely for the designation, Pharmacist and Data scientist. In the future, this can further be scrapped for designations or roles of different fields and a huge dataset can be worked upon based on different industry as well.

## **10. REFERENCES:**

[1] Riloff, E., Qadir, A., Surve, P., De Silva, L., Gilbert, N., and Huang, R. "*Sarcasm as contrast between a positive sentiment and negative situation*", In EMNLP 2013, pp. 704–714

[2] B. Agarwal, N. Mittal, "*Prominent Feature Extraction for Review Analysis: An Empirical Study*", In Journal of Experimental and theoretical Artificial Intelligence, 2014, DOI: 10.1080/0952813X.2014.977830

[3] <https://scikit-learn.org/stable/>

[4] <https://www.geeksforgeeks.org/python-programming-language/>

[5] <https://medium.com/@nagam808surya/sentiment-analysis-using-naive-bayes-classifier-aab9980a8ebf>

[6] Munir Ahmad, Shabib Aftab, Syed Shah Muhammad and Sarfraz Ahmad,



“Machine Learning Techniques for Sentiment Analysis: A Review”,  
INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY SCIENCES AND  
ENGINEERING, VOL. 8, NO. 3, APRIL 2017

[7] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, M. Stede, “Lexicon-based  
methods for sentiment analysis”, Computational Linguistics, v.37 n.2, p.267-  
307, 2011