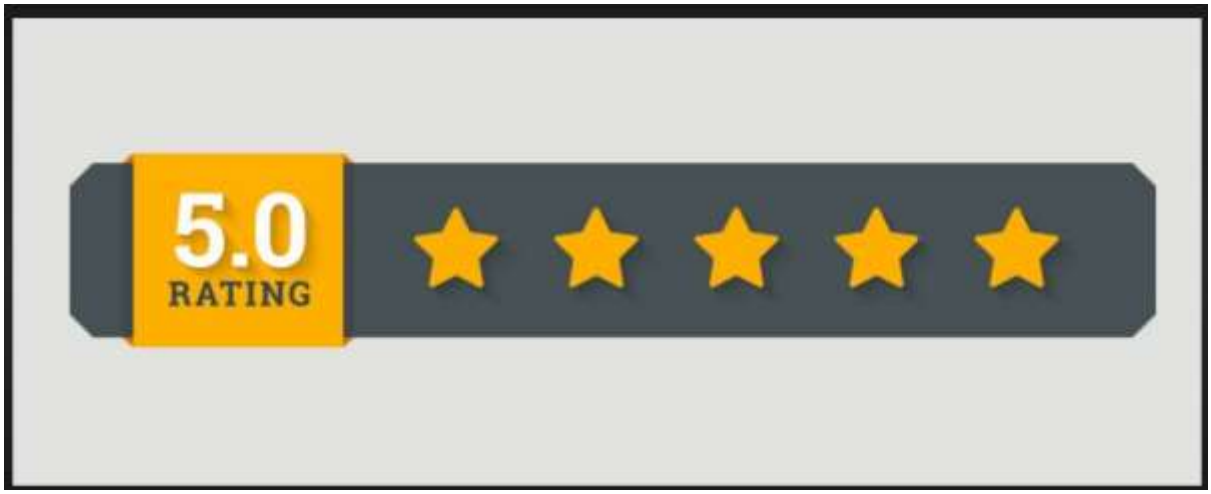
FLIP ROBO

RATINGS REVIEW PROJECT



Submitted by:

YASHNA SHAH

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my SME (Subject Matter Expert) Sapna Verma as well as Flip Robo Technologies who gave me the opportunity to do this project on Ratings Review Prediction Project, which also helped me in doing lots of research wherein I came to know about so many new things especially the data collection part.

Also, I have utilized a few external resources that helped me to complete the project. I ensured that I learn from the samples and modify things according to my project requirement. All the external resources that were used in creating this project are listed below:

1) https://www.google.com/

2) https://www.youtube.com/

3) https://scikit-learn.org/stable/user_guide.html

4) https://github.com/

5) https://www.kaggle.com/

6) https://medium.com/

7) https://towardsdatascience.com/

8) https://www.analyticsvidhya.com/

INTRODUCTION

- ## Business Problem Framing

We have a client who has a website where people write different reviews for technical products. Now they are adding a new feature to their website i.e. The reviewer will have to add stars(rating) as well with the review. The rating is out 5 stars and it only has 5 options available 1 star, 2 stars, 3 stars, 4 stars, 5 stars. Now they want to predict ratings for the reviews which were written in the past and they don't have a rating. So, we have to build an application which can predict the rating by seeing the review

- ## Conceptual Background of the Domain Problem

Internet has revolutionized the way of shopping. Now we can do shopping seating at our home just by few clicks. The ecommerce industry is growing rapidly by extending its reach to almost every corner of the world. So there are plenty of online marketing websites available. This leads to confusion in our mind from where we can get the best products. Rating prediction is a well-known recommendation task aiming to predict user's rating for those items which were not rated yet by customers. Predictions are computed from users' explicit feedback.

- ## Review of Literature

Rating is a classification or ranking of someone or something based on a comparative assessment of their quality, standard or overall performance. Reviews and ratings play a very vital role in deciding the correct product. Nowadays, buyers can get the real idea of the products by reading the reviews given by other buyer on the websites. Everyday we come across various products in our lives, on the digital medium we swipe across hundreds of product choices under one category. It gets tedious for the customers to make selections.

- ## Motivation for the Problem Undertaken

As we know, ratings can be easily sorted and judged whether the product is good or bad. But when it comes to reviews, we need to read through every line to make sure what the review conveys. Many product reviews are not accompanied by scale rating system, and are consisted only of textual evaluation. In this case , it becomes daunting and time consuming to compare different products in order to eventually make a choice between them. Therefore it is critically important to predict the user rating

from the text review. Getting an overall sense of a text review could in turn improve consumer experience.

The goal of this project is to build an application which can predict the rating by seeing the review. In the long run, this would allow people to better explain and review their purchase with each other in this increasingly digital world.

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

As per the client requirement, I scrapped ratings and reviews from Flipkart. This is then saved into CSV format file. Then loaded this data in the data frame and did some important NLP processing steps and gone through several EDA steps to analyse the data. After completing all the necessary steps, I have built an NLP model to predict the ratings.

- ## Data Sources and their formats

Data Collection Phase

We have to scrape at least 20000 rows of data. You can scrape more data as well, it's up to you. more the data better the model In this section you need to scrape the reviews of different laptops, Phones, Headphones, smart watches, Professional Cameras, Printers, Monitors, Home theater, Router from different e commerce websites.

Basically, we need these columns:

1) reviews of the product.

2) rating of the product.

Fetch an equal number of reviews for each rating, for example if you are fetching 10000 reviews then all ratings 1,2,3,4,5 should be 2000. It will balance our data set.

Convert all the ratings to their round number, as there are only 5 options for rating i.e., 1,2,3,4,5. If a rating is 4.5 convert it 5.

Model Building Phase

After collecting the data, you need to build a machine learning model. Before model building do all data preprocessing steps involving NLP. Try different models with different hyper parameters and select the best model.

Follow the complete life cycle of data science. Include all the steps like:

1. Data Cleaning

2. Exploratory Data Analysis

3. Data Preprocessing
4. Model Building
5. Model Evaluation
6. Selecting the best mode

- ## Data Pre - processing Done

The following pre-processing pipeline is required to be performed before building the classification model prediction:

1. Load dataset
2. There were no null values in our dataset.
3. Drop Unnamed:0 column as it was not useful.
4. Convert review text to lower case and replace '\n' with single space.
5. Keep only text data ie. a-z' and remove other data from comment text.
6. Remove stop words and punctuations
7. Apply Stemming
8. Convert text to vectors using TfidfVectorizer
9. Saving the mode

- ## Data Inputs- Logic- Output Relationships

I have analysed the input output logic with word cloud and I have word clouded the reviews as per their ratings classification . A tag/word cloud is a novelty visual representation of text data, typically used to depict keyword metadata on websites, or to visualize free form text. It's an image composed of words used in a particular text or subject, in which the size of each word indicates its frequency or importance. These are the reviews that belongs to different ratings so with the help of word cloud we can see all the frequently used words in each and every ratings. It is observed that 5 – star rating reviews have mostly positive words while the 1 – star reviews are loaded with negative words.

- ## State the set of assumptions (if any) related to the problem under consideration

The 5-star rating system allows respondents to rank their feedback on a 5 point scale from 1-5. The more stars that are selected, the more positively your customer is responding to the purchased products. People tend to overlook businesses with a less than 4-star rating to lower. Usually, when people are researching a company, the goal is to find one with the highest overall score and best reviews. Having a 5-star rating means a lot for your reputation score and acquiring new customers.

- Hardware and Software Requirements and Tools Used

Hardware technology being used.
RAM : 16 GB
CPU : 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
GPU : intel Iris Graphics Software technology being used.
Programming language :
Python Distribution : Anaconda Navigator Browser based language shell : Jupyter Notebook
Libraries/Packages specifically being used.
Pandas, NumPy, matplotlib, seaborn, scikit-learn, NLTK

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

1. I checked through the entire training dataset for any kind of missing values information and all these pre - processing steps were repeated on the testing dataset as well.

2. Then we went ahead and took a look at the dataset information. Using the info method, we are able to confirm the non-null count details as well as the datatype information.

3. Since there was no use of the "Unnamed:0" column I have dropped it.

4. Stemming is the process of reducing a word to its word stem that affixes to suffixes and prefixes or to the roots of words known as a lemma. Stemming is important in natural language understanding (NLU) and natural language processing (NLP)

- Testing of Identified Approaches (Algorithms)

The complete list of all the algorithms used:

1) Logistic Regression

2) Random Forest Classifier

3) K Nearest Neighbors Classifier

4) Decision Tree Classifier

- Run and Evaluate selected model

1. Logistic Regression

```
LR=LogisticRegression()
Model.append('LogisticRegression')
LR.fit(x_train,y_train)
print(LR)
pre=LR.predict(x_test)
print('\n')
AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)
score.append(AS*100)
sc=cross_val_score(LR,X,Y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc,'\n')
cv_score.append(sc*100)
```

```
LogisticRegression()


Accuracy_score=  0.9357826489657302
Cross_val_score= 0.9014326443372862
```

2. KNN Classifier

```
KNN=KNeighborsClassifier(n_neighbors=6)

Model.append('KNeighborsClassifier')
KNN.fit(x_train,y_train)
print(KNN)

pre=KNN.predict(x_test)
AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)

score.append(AS*100)
sc=cross_val_score(KNN,X,Y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc)
cv_score.append(sc*100)
```

```
KNeighborsClassifier(n_neighbors=6)
Accuracy_score=  0.8817536276628589
Cross_val_score= 0.8150212824293535
```

3. Decision Tree Classifier

```
DT=DecisionTreeClassifier()
Model.append('DecisionTreeClassifier')
DT.fit(x_train,y_train)
print(DT)
pre=DT.predict(x_test)
print('\n')
AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)
score.append(AS*100)
sc=cross_val_score(DT,X,Y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc,'\n')
cv_score.append(sc*100)
```

```
DecisionTreeClassifier()


Accuracy_score=  0.9501389317690645
Cross_val_score= 0.8708286888979521
```

4. RandomForest Classifier

```
RF = RandomForestClassifier()
Model.append('RandomForestClassifier')
RF.fit(x_train,y_train)
print(RF)
pre=RF.predict(x_test)
print('\n')

AS=accuracy_score(y_test,pre)
print('Accuracy_score= ',AS)
score.append(AS*100)
sc=cross_val_score(RF,X,Y,cv=5,scoring='accuracy').mean()
print('Cross_val_score=',sc,'\n')

cv_score.append(sc*100)
```

```
RandomForestClassifier()


Accuracy_score=  0.9646495832046929
Cross_val_score= 0.9164841222674281
```

- Key Metrics for success in solving problem under consideration

The key metrics used here were accuracy score, cross validation score, classification report and confusion matrix. We tried to find out the best parameters and also to increase our scores by using Hyperparameter Tuning and we will be using GridSearchCV Method.

```
#Calculating the difference between accuracy score and cross validation score
difference = list(np.array(score) - np.array(cv_score))

# Creating dataframe
result = pd.DataFrame({'Model':Model,'Accuracy_score':score,'Cross_val_score':cv_score,
                       'Difference':difference})
result
```

	Model	Accuracy_score	Cross_val_score	Difference
0	LogisticRegression	93.578265	90.143264	3.435000
1	KNeighborsClassifier	88.175363	81.502128	6.673235
2	DecisionTreeClassifier	95.013893	87.082869	7.931024
3	RandomForestClassifier	96.464958	91.648412	4.816546

```
# Lets select the different parameters for tuning our best model (RandomForestClassifier)
grid_params = {'n_estimators':[100,200],
               'criterion':['gini','entropy'],
               'max_depth': [500,800],
               'bootstrap':[True,False]}

# Train the model with given parameters using GridSearchCV
GSCV =  GridSearchCV(RF, grid_params, cv=3, verbose=3)
GSCV.fit(x_train, y_train)
```

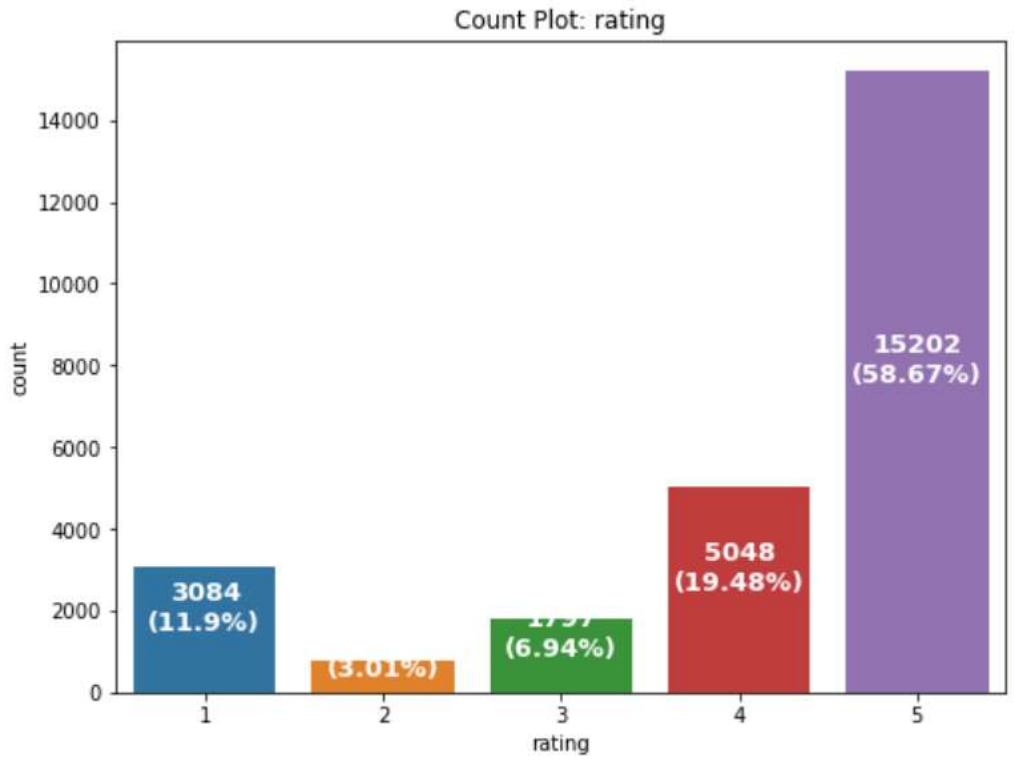
```
GSCV.best_params_
```

```
{'bootstrap': False,
 'criterion': 'gini',
 'max_depth': 800,
 'n_estimators': 200}
```

```
ACCURACY SCORE: 96.40321086755172

CLASSIFICATION REPORT:
              precision    recall  f1-score   support

           1       0.98      1.00      0.99       778
           2       0.99      0.85      0.92       212
           3       0.97      0.84      0.90       447
           4       0.95      0.92      0.93      1295
           5       0.96      0.99      0.98      3746

    accuracy                           0.96      6478
   macro avg       0.97      0.92      0.94      6478
weighted avg       0.96      0.96      0.96      6478


CONFUSION MATRIX:
 [[ 775    1    0    0    2]
 [  12  181    0   13    6]
 [   7    0  376   26   38]
 [   0    0   11 1191   93]
 [   0    0    0   24 3722]]
```

- Visualizations
  1. Checking feature rating using count plot

```
#checking feature rating using count plot
x = 'rating'
fig, ax = plt.subplots(1,1,figsize=(8,6))
sns.countplot(x=x,data=df,ax=ax)
p=0
for i in ax.patches:
    q = i.get_height()/2
    val = i.get_height()
    ratio = round(val*100/len(df),2)
    prn = f"{val}\n({ratio}%)"
    ax.text(p,q,prn,ha="center",color="white",rotation=0,fontweight="bold",fontsize="13")
    p += 1

plt.title("Count Plot: rating")
plt.show()
```

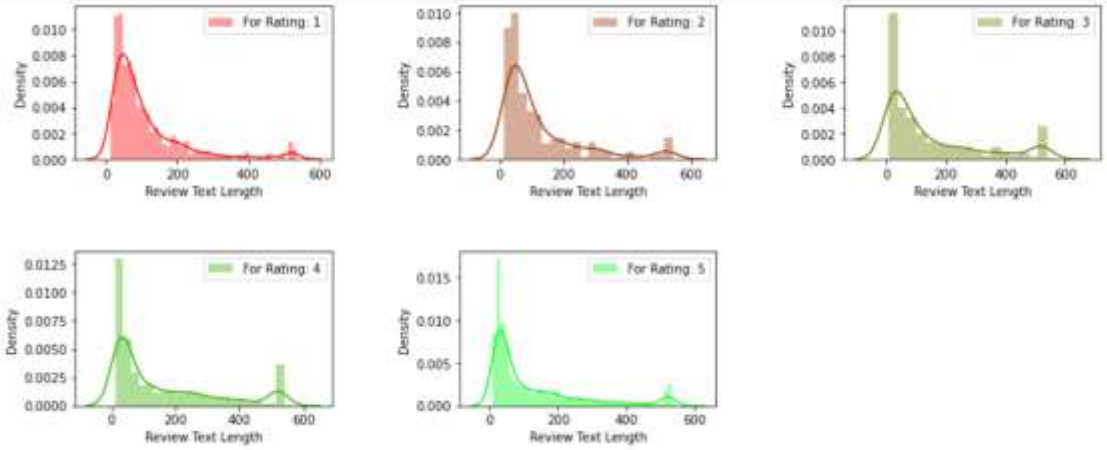


Observation:

We can see that the highest number of customer rating received are for 5 star rating.

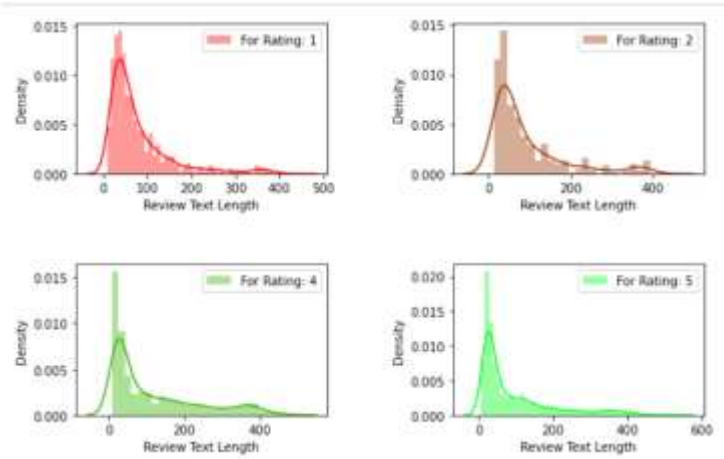
2. Checking review text length distribution for each rating

```
#checking review text length distribution for each rating
ratings = np.sort(df.rating.unique())
cols = 3
rows = len(ratings)//cols
if rows % cols != 0:
    rows += 1

fig = plt.figure(figsize=(15,6))
plt.subplots_adjust(hspace=0.6, wspace=0.5)
p = 1
colors = [(1,0,0,1),(0.6,0.2,0,1),(0.4,0.5,0,1),(0.2,0.7,0,1),(0,1,0.1,1)]
for i in ratings:
    axis = fig.add_subplot(rows,cols,p)
    sns.distplot(df.length[df.rating==i],ax=axis,label=f"For Rating: {i}",color=colors[i-1])
    axis.set_xlabel(f"Review Text Length")
    axis.legend()
    p += 1

plt.show()
```

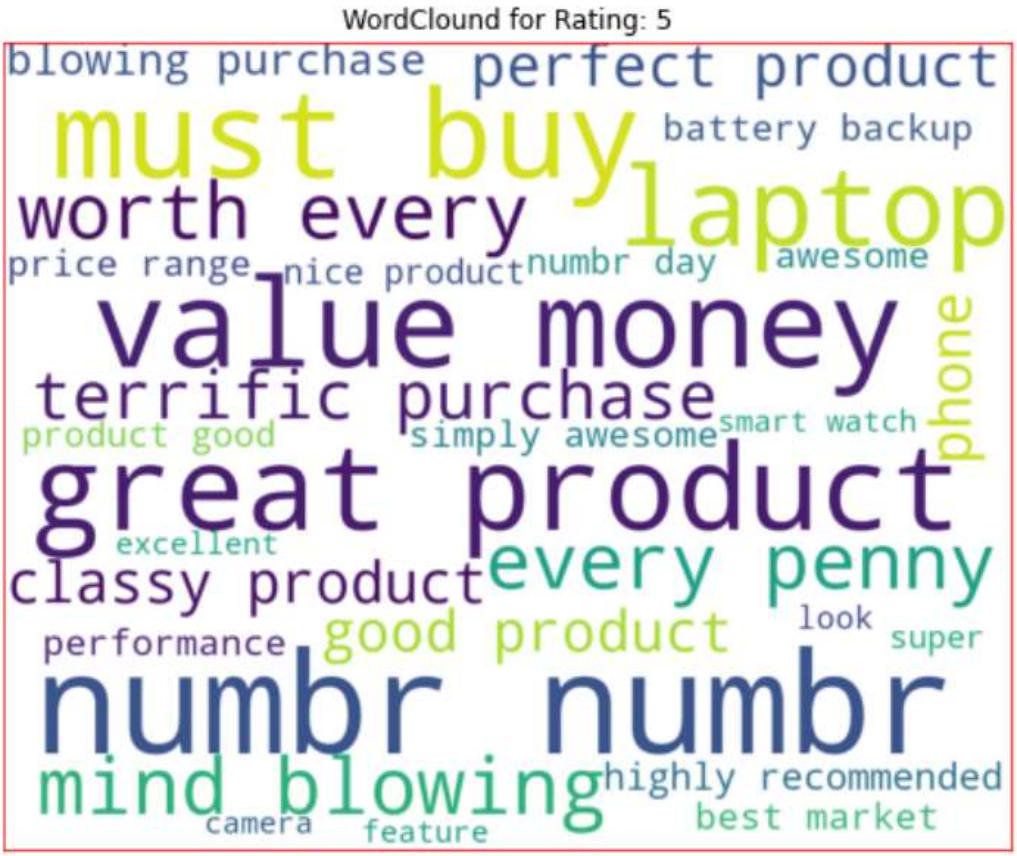


3. Checking review text length distribution for each rating after cleaning

```
#checking review text length distribution for each rating after cleaning
ratings = np.sort(df.rating.unique())
cols = 3
rows = len(ratings)//cols
if len(ratings) % cols != 0:
    rows += 1

fig = plt.figure(figsize=(15,6))
plt.subplots_adjust(hspace=0.6, wspace=0.5)
p = 1
colors = [(1,0,0,1),(0.6,0.2,0,1),(0.4,0.5,0,1),(0.2,0.7,0,1),(0,1,0.1,1)]
for i in ratings:
    axis = fig.add_subplot(rows,cols,p)
    sns.distplot(df.clean_length[df.rating==i],ax=axis,label=f"For Rating: {i}",color=colors[i-1])
    axis.set_xlabel(f"Review Text Length")
    axis.legend()
    p += 1

plt.show()
```



4. Word Cloud
   getting sense of loud word in each rating



- Interpretation of the Results

1. With word cloud it was found that rating 1 consists of words like waste money, worst experience,horrible, absolute rubbish etc; rating 2 consists of words like problem, issue, bad, poor, slow etc; rating 3 consists of words like average nice etc; rating 4 consists of words like really nice, good choice, good value etc; rating 5 consists of words like mind blowing, best market, good product, excellent etc.
2. It was evident that the length of review text decreases by a large amount after removal and replacement of certain items like punctuations, extra spaces, money symbols with the help of distribution plots.

CONCLUSION

- ## Key Findings and Conclusions of the Study

This research evaluated that the rating of a product classification using machine learning and NLP techniques. We compared the various machine learning algorithms accuracy by performing detailed experimental analysis while classifying the text into 5 categories.

RandomForest Classifier have shown a better performance with our real life data than others and the most performing models are all ensemble classifiers.

We found that the minimum difference between the accuracy score and the cross validation score is for RandomForest apart from Logical Regression. So the best fit model for our project is Random Forest Classifier.

- ## Learning Outcomes of the Study in respect of Data Science

Through this project we were able to learn various Natural language processing techniques like lemmatization, stemming, removal of stopwords. We were also able to learn to convert strings into vectors through hash vectorizer. In this project we applied different evaluation metrics like log loss, hamming loss besides accuracy.