Heart Disease Prediction
And
Awesomeness Analysis

Bishal Sainju, Supratik Chanda, Victor Lee

Heart Disease Prediction

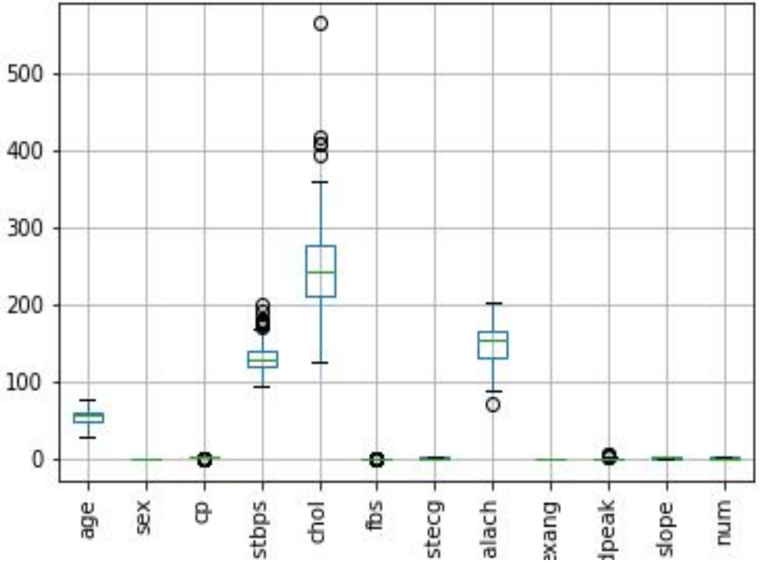
- Cleveland Dataset: 303 entries, 14 attributes.
- ‘num’ attribute contains info about whether a person has a heart disease or not.
- Values in ‘num’ attribute: 0 → No HD, 1, 2, 3 → Yes HD

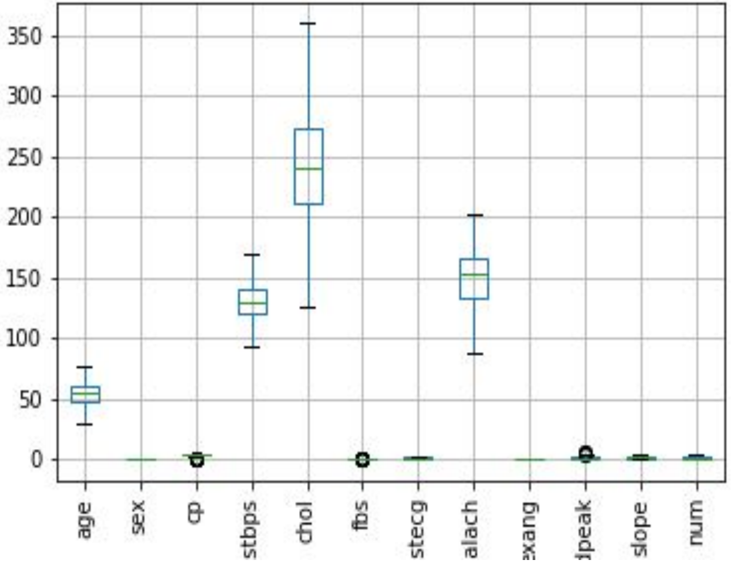
Data Preprocessing

- Removed NaN, ‘?’
- Removed outliers
- Standardized data
- Converted 1, 2, 3 in num to 1(person has heart disease)
- Encoded dummies to categorical variables:

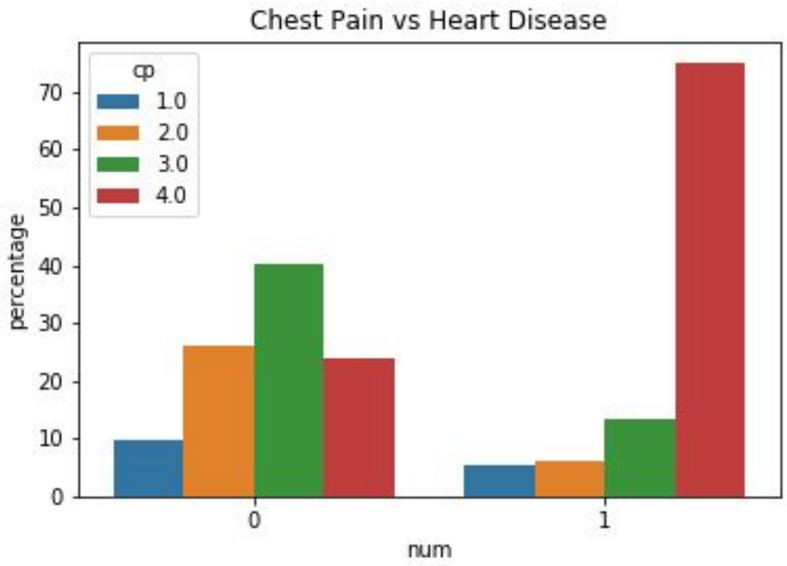
['cp', 'restecg', 'slope', 'ca', 'thal']

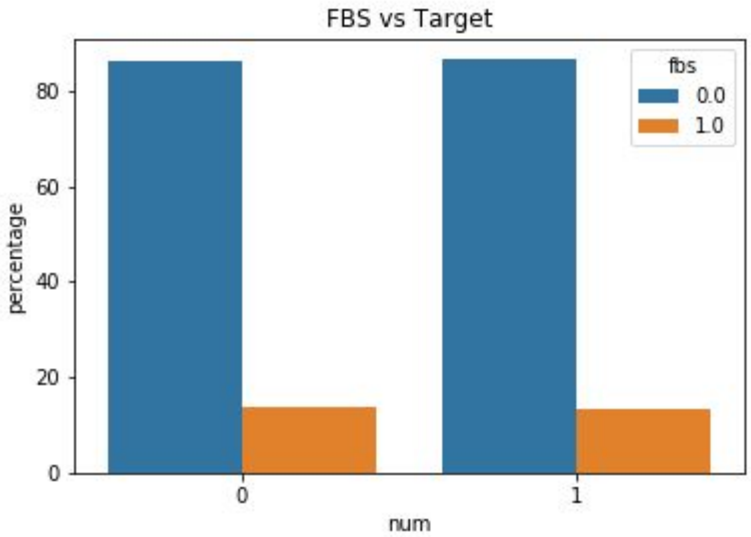
Outliers Removal



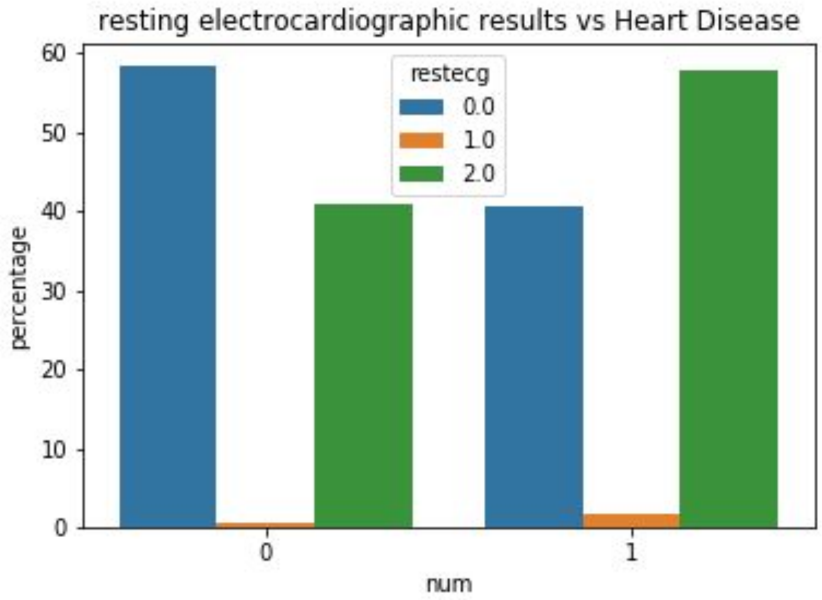


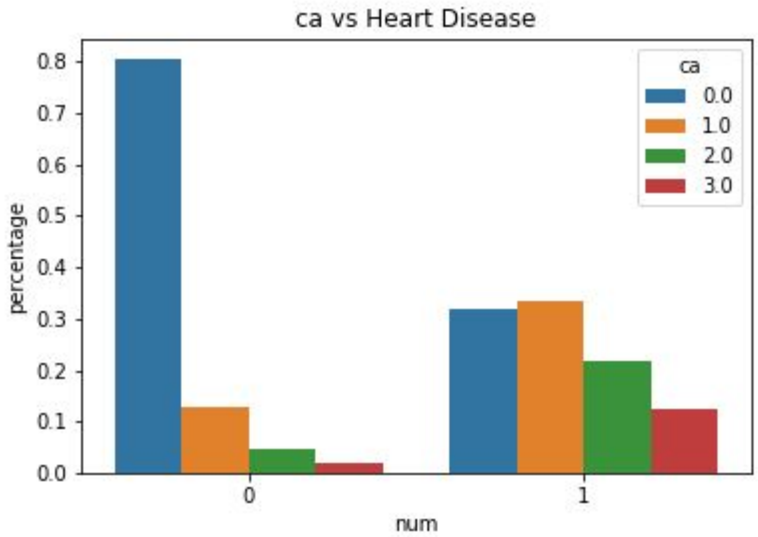
Data Analysis



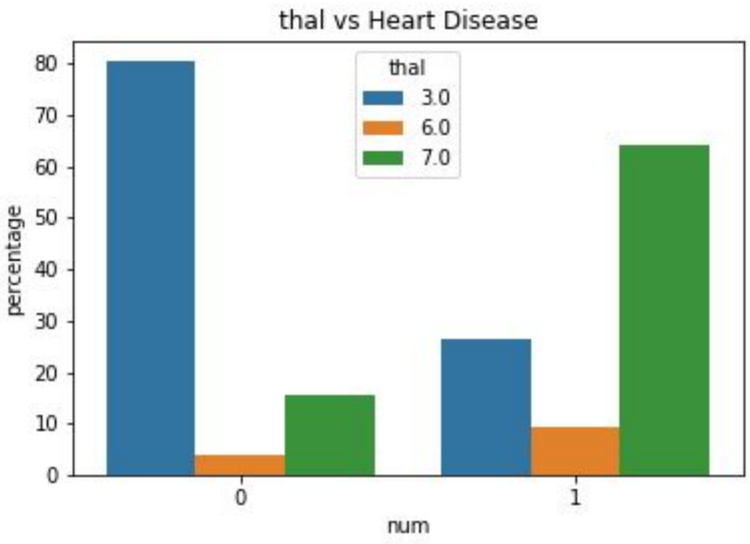


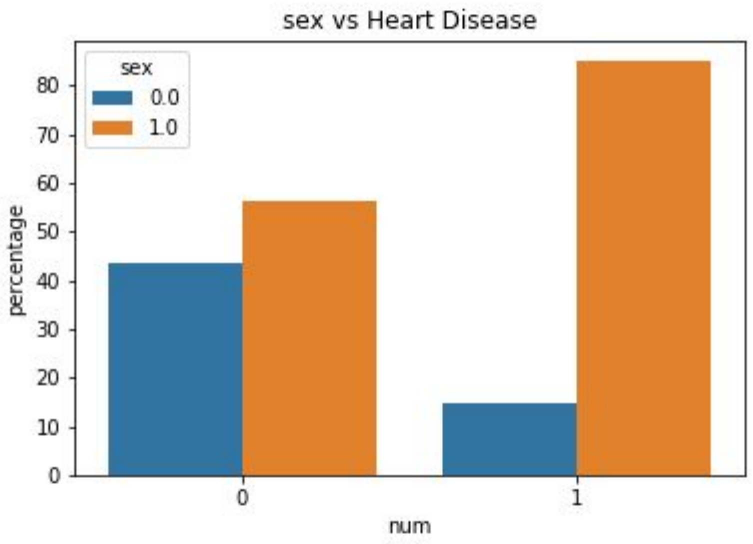
Data Analysis





Data Analysis





Correlation of various attribute with num

```
age             0.230561
sex             0.309960
trestbps        0.131340
chol            0.105463
fbs            -0.005178
thalach        -0.433597
exang           0.414825
oldpeak         0.438209
num             1.000000
cp_1.0         -0.079296
cp_2.0         -0.261296
cp_3.0         -0.299103
cp_4.0          0.508388
restecg_0.0    -0.177411
restecg_1.0     0.044314
restecg_2.0     0.168382
slope_1.0      -0.374651
slope_2.0       0.341924
slope_3.0       0.067508
ca_0.0         -0.489967
ca_1.0          0.246310
ca_2.0          0.261680
ca_3.0          0.209577
thal_3.0       -0.541220
thal_6.0        0.111589
thal_7.0        0.498312
Name: num, dtype: float64
```

['thal_3.0', 'cp_4.0', 'thal_7.0', 'ca_0.0', 'oldpeak', 'thalach'] are more important.

Important Features(Feature Selection)

['thal_3.0', 'cp_4.0', 'thal_7.0', 'ca_0.0', 'oldpeak', 'thalach', 'exang', 'slope_1.0', 'slope_2.0', 'sex', 'cp_3.0', 'ca_2.0', 'cp_2.0', 'ca_1.0', 'age', 'ca_3.0', 'restecg_0.0', 'restecg_2.0', 'trestbps', 'thal_6.0', 'chol', 'cp_1.0', 'slope_3.0', 'restecg_1.0', 'fbs']

Used SelectKBest from sklearn.feature_selection, using f_regression as scoring_function.

Tried among f_regression, f_classif, mutual_info_classif etc. but f_regression gave better results

Model Creation

For KNN:

N_Neighbor: range(5, 15)

Distance: [‘Euclidean’]

Weights: [‘uniform’, ‘distance’]

For Feature Selection:

N_Componenet: range(15, 21)

Feature Selection Method: [SelectKBest, PCA]

Best model using custom KNN model

- For 19 features and 6 nearest neighbor we found the best f1-score of 86.32%
- Weights: ‘Uniform’, Metric: ‘Euclidean’, K:6, N_Component: 19, SelectKBest

Best model using custom KNN model

```
Cross-Val: 4
              precision    recall  f1-score   support

         0.0       0.87      0.93      0.90        29
         1.0       0.92      0.86      0.89        28

   micro avg       0.89      0.89      0.89        57
   macro avg       0.90      0.89      0.89        57
weighted avg       0.90      0.89      0.89        57

[[27  2]
 [ 4 24]]
f1: 0.888888888888889, acc: 0.8947368421052632

{
    "19": {
        "6": {
            "acc": 0.883,
            "f1": 0.8632,
            "prec": 0.872,
            "rec": 0.8568
        }
    }
}
```

We found the maximum f1-score of 86.32%

Using built-in model

{'classify__algorithm': 'auto', 'classify__n_neighbors': 9, 'classify__weights': 'uniform', 'reduce_dim': SelectKBest(k=19, score_func=<function f_regression at 0x1a15a0f048>), 'reduce_dim__k': 19, 'reduce_dim__score_func': <function f_regression at 0x1a15a0f048>}

Obtained f1 score: 85.17%

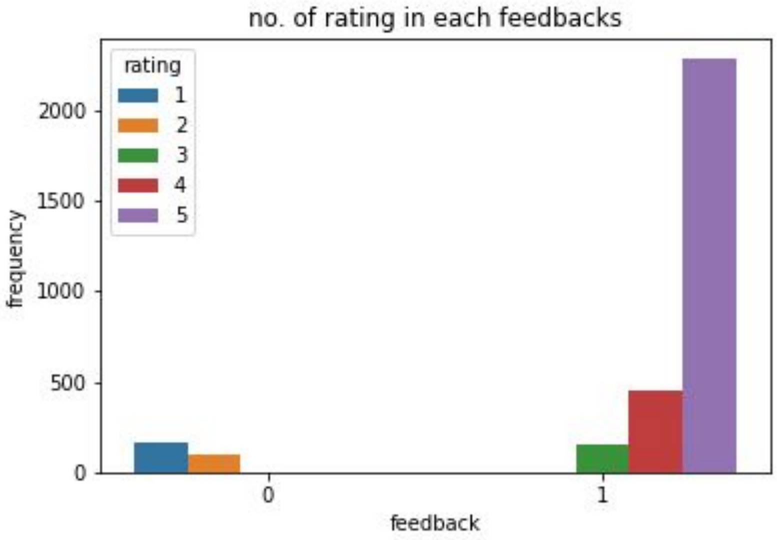
Using built-in model

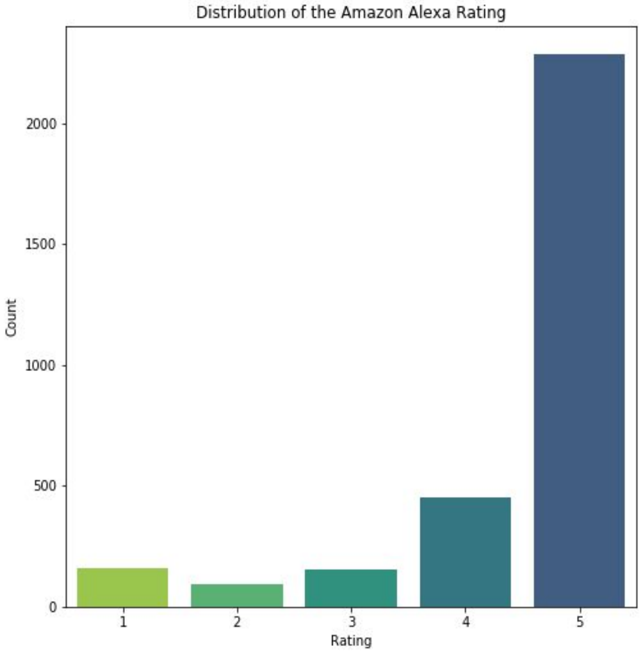
```
              precision    recall  f1-score   support

         0.0       0.90      0.93      0.92        30
         1.0       0.92      0.88      0.90        25

   micro avg       0.91      0.91      0.91        55
   macro avg       0.91      0.91      0.91        55
weighted avg       0.91      0.91      0.91        55

[[28  2]
 [ 3 22]]
acc:0.9090909090909091, prec:0.9166666666666666, recall:0.88, f1:0.8979591836734694


0.8517
```

Awesomeness Analysis

- 3150 Amazon customer reviews(input text), ratings(1-5), date of review, variant and feedback of various amazon Alexa products
- Feedback: 1→ 2741, 0→ 409
- Rating:5 = ‘Awesome’ , Otherwise: ‘Now Awesome’

Data Analysis





Data Preprocessing

- Data Cleaning(checked for any missing value)
- NLP using NLTK package
- Lowercased
- Stopword removal
- Tokenization
- Stemming

Feature Extraction

- Used TF-IDF vectoriozer to get the TF-IDF matrix.
- Created bag of words of 500 different features(words)
- X.shape: 3150 X 500

Bag of Words

```
['abil', 'abl', 'absolut', 'access', 'account', 'activ', 'actual', 'ad', 'adapt', 'add', 'addit', 'advertis', 'alar
m', 'alexa', 'allow', 'almost', 'alon', 'along', 'alreadi', 'also', 'although', 'alway', 'amaz', 'amazon', 'annoy',
'anoth', 'answer', 'anyth', 'app', 'appl', 'around', 'ask', 'assist', 'audibl', 'audio', 'avail', 'away', 'awesom',
'back', 'bad', 'base', 'basic', 'bass', 'batteri', 'bed', 'bedroom', 'bedsid', 'begin', 'best', 'better', 'big', 'bir
thday', 'bit', 'blue', 'bluetooth', 'book', 'bose', 'bought', 'brand', 'brief', 'built', 'bulb', 'button', 'buy', 'ca
bl', 'call', 'came', 'camera', 'cannot', 'capabl', 'cell', 'chang', 'channel', 'chat', 'check', 'clear', 'clock', 'co
lor', 'come', 'command', 'commun', 'complaint', 'complet', 'comput', 'connect', 'consid', 'constantli', 'contact', 'c
ontinu', 'control', 'conveni', 'cook', 'cool', 'cord', 'cost', 'could', 'coupl', 'creat', 'current', 'custom', 'dail
i', 'daughter', 'day', 'deal', 'decid', 'definit', 'deliv', 'design', 'devic', 'differ', 'direct', 'disappoint', 'dis
cov', 'dislik', 'display', 'done', 'door', 'dot', 'download', 'drop', 'eas', 'easi', 'easier', 'easili', 'echo', 'el
s', 'enabl', 'end', 'enjoy', 'enough', 'entertain', 'especi', 'etc', 'even', 'ever', 'everi', 'everyon', 'everyth',
'exactli', 'excel', 'except', 'excit', 'expect', 'experi', 'explor', 'extern', 'extra', 'extrem', 'face', 'fact', 'fa
mili', 'fan', 'fantast', 'far', 'fast', 'favorit', 'featur', 'feel', 'figur', 'final', 'find', 'fine', 'fire', 'fires
tick', 'first', 'fix', 'flash', 'forward', 'found', 'free', 'friend', 'friendli', 'frustrat', 'full', 'fun', 'functio
n', 'futur', 'game', 'gave', 'gen', 'gener', 'get', 'gift', 'give', 'glad', 'go', 'goe', 'good', 'googl', 'got', 'gre
at', 'group', 'half', 'handi', 'happen', 'happi', 'hard', 'hear', 'heard', 'help', 'high', 'highli', 'home', 'hook',
'hope', 'hour', 'hous', 'household', 'howev', 'hub', 'hue', 'huge', 'husband', 'immedi', 'impress', 'improv', 'inclu
d', 'inform', 'instal', 'instead', 'instruct', 'integr', 'interact', 'intercom', 'internet', 'issu', 'item', 'job',
'joke', 'keep', 'kid', 'kind', 'kitchen', 'know', 'lamp', 'later', 'learn', 'least', 'less', 'let', 'life', 'light',
'like', 'limit', 'link', 'list', 'listen', 'littl', 'live', 'lock', 'lol', 'long', 'longer', 'look', 'lot', 'loud',
'louder', 'love', 'low', 'lyric', 'made', 'mainli', 'make', 'mani', 'may', 'mini', 'minut', 'miss', 'mom', 'money',
'month', 'morn', 'mostli', 'mother', 'move', 'movi', 'much', 'multipl', 'music', 'must', 'name', 'nd', 'need', 'netfl
ix', 'never', 'new', 'news', 'next', 'nice', 'night', 'nightstand', 'noth', 'number', 'offer', 'offic', 'often', 'o
k', 'old', 'one', 'open', 'oper', 'option', 'order', 'origin', 'outlet', 'overal', 'packag', 'paid', 'pandora', 'par
t', 'pay', 'peopl', 'perfect', 'perfectli', 'perform', 'person', 'philip', 'phone', 'pick', 'pictur', 'piec', 'plan',
'play', 'playlist', 'pleas', 'plu', 'plug', 'power', 'pretti', 'price', 'prime', 'probabl', 'problem', 'product', 'pr
ogram', 'provid', 'purchas', 'put', 'qualiti', 'question', 'quick', 'quit', 'radio', 'rang', 'rd', 'read', 'readi',
'real', 'realiz', 'realli', 'reason', 'receiv', 'recip', 'recommend', 'refurbish', 'regret', 'regular', 'remind', 're
mot', 'repeat', 'replac', 'request', 'respond', 'respons', 'return', 'review', 'right', 'ring', 'room', 'run', 'sai
d', 'sale', 'satisfi', 'save', 'say', 'schedul', 'screen', 'search', 'second', 'secur', 'see', 'seem', 'servic', 'se
t', 'setup', 'sever', 'shop', 'show', 'simpl', 'sinc', 'sit', 'size', 'skill', 'sleep', 'small', 'smaller', 'smart',
'someon', 'someth', 'sometim', 'son', 'song', 'soon', 'sound', 'space', 'speak', 'speaker', 'specif', 'spot', 'spotif
i', 'st', 'stand', 'star', 'start', 'station', 'step', 'stick', 'still', 'stop', 'stream', 'stuff', 'suggest', 'supe
r', 'support', 'suppos', 'sure', 'surpris', 'switch', 'sync', 'system', 'take', 'talk', 'tech', 'technolog', 'tell',
'terribl', 'thank', 'thermostat', 'thing', 'think', 'third', 'though', 'thought', 'three', 'throughout', 'time', 'tim
er', 'told', 'took', 'tooth', 'top', 'total', 'touch', 'tri', 'troubl', 'turn', 'tv', 'two', 'type', 'understand', 'u
nit', 'unless', 'unplug', 'updat', 'upgrad', 'us', 'use', 'user', 'valu', 'via', 'video', 'view', 'voic', 'volum', 'w
ait', 'wake', 'walk', 'want', 'watch', 'way', 'weather', 'week', 'well', 'went', 'white', 'whole', 'wife', 'wifi', 'w
ireless', 'wish', 'without', 'wonder', 'word', 'work', 'worth', 'would', 'wrong', 'year', 'yet', 'youtub']
(3150, 500)
```

Model Creation

- Created KNN Model
- Used GridSearchCV to find the best estimate and best parameter.

Best Model

{'algorithm': 'auto', 'leaf_size': 30, 'metric': 'jaccard', 'n_neighbors': 10, 'weights': 'distance'}

Obtained f1 score: 0.9031

```
              precision    recall  f1-score   support

           0       0.94      0.84      0.88        87
           1       0.94      0.98      0.96       229

   micro avg       0.94      0.94      0.94       316
   macro avg       0.94      0.91      0.92       316
weighted avg       0.94      0.94      0.94       316

[[ 73  14]
 [  5 224]]
acc:0.939873417721519, prec:0.9411764705882353, recall:0.9781659388646288, f1:0.9593147751605996
```

Future Improvements

- For heart disease, more dataset and more attributes could better the performance
- For Awesomeness analysis, using other NLP methods could help (like dictionary comparison, spelling correction, POS tagging)
- We haven’t tried on various variation of feature extraction, (only tf-idf used), we could check to see if other vectorization such as count vectorization etc. could help.
- Could use PCA to reduce the dimension.