Support Vector Machines (SVM)

Another machine learning algorithm is the support vector machine (SVM). It provides resourceful predictions for complex real-word problems, reduces redundant information, and is ideal for smaller datasets. Some real-world examples of when SVMs can be used are bioinformatics, face recognition, speech recognition, or cancer detection.

To perform a classification using the support vector machine algorithm, complete the following:

1. Access the "UCI Machine Learning Repository," located in the topic Resources. Note: There are about 120 data sets that are suitable for use in a classification task. For this part of the exercise, you must choose one of these datasets, provided it includes at least 10 attributes and 10,000 instances.
2. Ensure that the datasets are suitable for classification using this method.
3. You may search for data in other repositories, such as Data.gov, Kaggle or Scikit Learn.
4. Examine the repository through which you accessed the dataset and discuss data management measures set in place, such as protecting the privacy of those accessing the site and protecting the intellectual property rights of the data owners/contributors.

For your selected dataset, build a classification model as follows:

1. Explain the dataset and the type of information you wish to gain by applying a classification method.
2. Explain what makes SVM algorithm very special and very different from most other machine learning algorithms. Explain how you will be using it in your analysis (list the steps, the intuition behind the mathematical representation, and address its assumptions).
3. Explain the concepts: kernel, hyperplane, and decision boundary, and their role in SVM.
4. Explain the concepts: maximum margin, support vectors, and maximum margin hyperplane, and their role in SVM.
5. Import the necessary libraries, then read the dataset into a data frame and perform initial statistical exploration.
6. Clean the data and address unusual phenomena (e.g., normalization, feature scaling, outliers); use illustrative diagrams and plots and explain them.
7. Formulate two questions that can be answered by applying a classification method using the SVM
8. Split the data into 80% training and 20% testing sets using the train test split class.
9. Use a linear kernel to train the SVM classifier on the training set (e.g., fit the support vector regressor to the dataset). Explain the intuition behind each of the key mathematical steps.
10. Explain the choice of the optimal hyperplane.
11. Make classification predictions.
12. Interpret the results in the context of the questions you asked.
13. Validate your model using a confusion matrix, accuracy score, ROC-AUC curves, and k-fold cross validation. Then, explain the results.
14. Include all mathematical formulas used and graphs representing the final outcomes.

Prepare a comprehensive technical report as a markdown document or Jupyter notebook, including all code, code comments, all outputs, plots, and analysis. Make sure the project documentation contains a) problem statement, b) algorithm of the solution, c) analysis of the findings, and d) references.

Solution

Support Vector Machines (or SVMs) are a set of machine learning algorithms which are known for great results and it's efficiency for classification. In this project, an SVC (Support Vector Classifier) will be used to perform sentimental analysis.

Problem Statement

Often times, going to a doctor to discuss which over the counter medicine may work best for a specific condition may be a daunting tasks. Not only would one require to spend an entire day dedicated to the doctor's visit, but the odds of having to ask for some time off to visit the doctor can be dreadful as that may mean in some cases that the usual paycheck will be lighter later on. Current review systems, may be incredibly difficult to ascertain in terms of how useful a drug may be, hence one may desire to use an app or web extension that rates the usefulness of the review based on it's description. This may lead to better confidence in patients buying their drugs from online stores (such as Amazon's new pharmaceutical section) while taking advantage of deals and discounts.

In [ ]:
```
import nltk
import pandas as pd
import datetime as dt
from sklearn import svm
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
from nltk.tokenize import word_tokenize
from sklearn.metrics import classification_report
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split, cross_val_score, KFold
from sklearn.metrics import accuracy_score, precision_score, recall_score, roc_auc_score, roc_curv

df__drug_review__train = pd.read_csv('drugsComTrain_raw.tsv', sep='\t')
df__drug_review__test = pd.read_csv('drugsComTest_raw.tsv', sep='\t')
df__drug_review = pd.concat([df__drug_review__test,df__drug_review__train])
df__drug_review.head(10)
```

Out[ ]:

	Unnamed: 0	drugName	condition	review	rating	date	usefulCount
0	163740	Mirtazapine	Depression	"I've tried a few antidepressants over th...	10.0	February 28, 2012	22
1	206473	Mesalamine	Crohn's Disease, Maintenance	"My son has Crohn's disease and has done ...	8.0	May 17, 2009	17
2	159672	Bactrim	Urinary Tract Infection	"Quick reduction of symptoms"	9.0	September 29, 2017	3
3	39293	Contrave	Weight Loss	"Contrave combines drugs that were used for al...	9.0	March 5, 2017	35
4	97768	Cyclafem 1 / 35	Birth Control	"I have been on this birth control for one cyc...	9.0	October 22, 2015	4
5	208087	Zyclara	Keratosis	"4 days in on first 2 weeks. Using on arms an...	4.0	July 3, 2014	13
6	215892	Copper	Birth Control	"I've had the copper coil for about 3 mon...	6.0	June 6, 2016	1
7	169852	Amitriptyline	Migraine Prevention	"This has been great for me. I've been on...	9.0	April 21, 2009	32
8	23295	Methadone	Opiate Withdrawal	"Ive been on Methadone for over ten years and ...	7.0	October 18, 2016	21

	Unnamed: 0	drugName	condition	review	rating	date	usefulCount
9	71428	Levora	Birth Control	"I was on this pill for almost two years. It d...	2.0	April 16, 2011	3

Now that we have the data, one can begin cleaning the data set through Exploratory Analysis and Data Processing.

Exploratory Analysis

First, there are weights associated with the review which determines how useful the aforementioned comment was in comparison to other comments. there is also the date which can be utilized to determine the relevance of the review, the review and rating provided by the patient, the patient's condition and the name of the drug.

Before one engineers new features from the review itself, let's start by cleaning the data and formatting the features into more readable forms. This can be done by first changing the format of the date to a more readable format for the model.

In [ ]:
```
df__drug_review.dropna(subset= ['review'])
df__drug_review['date'] = pd.to_datetime(df__drug_review['date'])
df__drug_review.head(10)
```

Out[ ]:

	Unnamed: 0	drugName	condition	review	rating	date	usefulCount
0	163740	Mirtazapine	Depression	"I've tried a few antidepressants over th...	10.0	2012-02-28	22
1	206473	Mesalamine	Crohn's Disease, Maintenance	"My son has Crohn's disease and has done ...	8.0	2009-05-17	17
2	159672	Bactrim	Urinary Tract Infection	"Quick reduction of symptoms"	9.0	2017-09-29	3
3	39293	Contrave	Weight Loss	"Contrave combines drugs that were used for al...	9.0	2017-03-05	35
4	97768	Cyclafem 1 / 35	Birth Control	"I have been on this birth control for one cyc...	9.0	2015-10-22	4
5	208087	Zyclara	Keratosis	"4 days in on first 2 weeks. Using on arms an...	4.0	2014-07-03	13
6	215892	Copper	Birth Control	"I've had the copper coil for about 3 mon...	6.0	2016-06-06	1
7	169852	Amitriptyline	Migraine Prevention	"This has been great for me. I've been on...	9.0	2009-04-21	32
8	23295	Methadone	Opiate Withdrawal	"Ive been on Methadone for over ten years and ...	7.0	2016-10-18	21
9	71428	Levora	Birth Control	"I was on this pill for almost two years. It d...	2.0	2011-04-16	3

Now that the date is in a more readable format, it can be separated into year, month and day.

In [ ]:
```
df__drug_review['Year'] = df__drug_review['date'].dt.year
df__drug_review['Month'] = df__drug_review['date'].dt.month
```

```
df__drug_review['Day'] = df__drug_review['date'].dt.day
df__drug_review.head(10)
```

Out[ ]:

	Unnamed: 0	drugName	condition	review	rating	date	usefulCount	Year	Month	Day
0	163740	Mirtazapine	Depression	"I've tried a few antidepressants over th...	10.0	2012-02-28	22	2012	2	28
1	206473	Mesalamine	Crohn's Disease, Maintenance	"My son has Crohn's disease and has done ...	8.0	2009-05-17	17	2009	5	17
2	159672	Bactrim	Urinary Tract Infection	"Quick reduction of symptoms"	9.0	2017-09-29	3	2017	9	29
3	39293	Contrave	Weight Loss	"Contrave combines drugs that were used for al...	9.0	2017-03-05	35	2017	3	5
4	97768	Cyclafem 1 / 35	Birth Control	"I have been on this birth control for one cyc...	9.0	2015-10-22	4	2015	10	22
5	208087	Zyclara	Keratosis	"4 days in on first 2 weeks. Using on arms an...	4.0	2014-07-03	13	2014	7	3
6	215892	Copper	Birth Control	"I've had the copper coil for about 3 mon...	6.0	2016-06-06	1	2016	6	6
7	169852	Amitriptyline	Migraine Prevention	"This has been great for me. I've been on...	9.0	2009-04-21	32	2009	4	21
8	23295	Methadone	Opiate Withdrawal	"Ive been on Methadone for over ten years and ...	7.0	2016-10-18	21	2016	10	18
9	71428	Levora	Birth Control	"I was on this pill for almost two years. It d...	2.0	2011-04-16	3	2011	4	16

Next up is to change the *condition* feature and the *drugName* feature into categorical values.

In [ ]:

```
df__drug_review['drugName'] = pd.Categorical(df__drug_review['drugName'])
df__drug_review['condition'] = pd.Categorical(df__drug_review['condition'])

dict__cat = dict()
for (col, data) in df__drug_review.iteritems():
    ctype = df__drug_review[str(col)].dtype
    if str(ctype) == 'category':
        dict__cat[str(col)] = dict(enumerate(df__drug_review[str(col)].cat.categories))
        df__drug_review[str(col)] = df__drug_review[str(col)].cat.codes
```

Now that the dataset is completely assembled and well positioned. Let's view how the features correlate with each other (with the exception being the review as that will require some feature engineering).

In [ ]:

```
corr_matrix = df__drug_review.corr()
corr_matrix\
    .style\
        .background_gradient(cmap='coolwarm', axis=None, vmin=-1, vmax=1)\
            .format(precision=2)
```

Out[ ]:

	Unnamed: 0	drugName	condition	rating	usefulCount	Year	Month	Day
Unnamed: 0	1.00	-0.02	-0.02	0.02	0.02	-0.01	0.00	-0.00
drugName	-0.02	1.00	0.08	0.01	0.03	0.01	0.00	-0.00
condition	-0.02	0.08	1.00	0.05	0.10	-0.04	-0.00	-0.00
rating	0.02	0.01	0.05	1.00	0.23	-0.19	-0.02	0.00
usefulCount	0.02	0.03	0.10	0.23	1.00	-0.26	-0.03	0.00
Year	-0.01	0.01	-0.04	-0.19	-0.26	1.00	-0.08	-0.01
Month	0.00	0.00	-0.00	-0.02	-0.03	-0.08	1.00	-0.01
Day	-0.00	-0.00	-0.00	0.00	0.00	-0.01	-0.01	1.00

As can be observed above, there seems to be little to no correlation between any of the features and the response variable (usefulCount). In this case, let's transform the response variable *usefulCount* to see if the correlation can be improved.

In [ ]:

```
df__drug_review['usefulCount_percent'] = df__drug_review['usefulCount'] / df__drug_review['usefulC
df__drug_review['usefulCount_percent'] = df__drug_review['usefulCount'].round()

corr_matrix = df__drug_review.corr()
corr_matrix\
    .style\
        .background_gradient(cmap='coolwarm', axis=None, vmin=-1, vmax=1)\
            .format(precision=2)
```

Out[ ]:

	Unnamed: 0	drugName	condition	rating	usefulCount	Year	Month	Day	usefulCount_percer
Unnamed: 0	1.00	-0.02	-0.02	0.02	0.02	-0.01	0.00	-0.00	0.0
drugName	-0.02	1.00	0.08	0.01	0.03	0.01	0.00	-0.00	0.0
condition	-0.02	0.08	1.00	0.05	0.10	-0.04	-0.00	-0.00	0.1
rating	0.02	0.01	0.05	1.00	0.23	-0.19	-0.02	0.00	0.2
usefulCount	0.02	0.03	0.10	0.23	1.00	-0.26	-0.03	0.00	1.0
Year	-0.01	0.01	-0.04	-0.19	-0.26	1.00	-0.08	-0.01	-0.2
Month	0.00	0.00	-0.00	-0.02	-0.03	-0.08	1.00	-0.01	-0.0
Day	-0.00	-0.00	-0.00	0.00	0.00	-0.01	-0.01	1.00	0.0
usefulCount_percent	0.02	0.03	0.10	0.23	1.00	-0.26	-0.03	0.00	1.0

From the new correlation matrix above, one is able to notice that by normalizing the *usefulCount* feature causes other numerical values such as rating and the year to remain the same in correlation with the response variable. For the sake of not allowing bias/misrepresentation of the data to be introduced into the model, only the review will be taken into consideration.

Now that the exploratory analysis of the data is finished, one can begin the data processing portion.

Data Processing

First the data must be split into it's X and y components (explanatory and response variable(s)) and the explanatory variables must then be cleaned and vectorized.

In [ ]:
```
X = df__drug_review['review']
y = df__drug_review['usefulCount']
nltk.download('stopwords')
nltk.download('punkt')

def remove_special(text:str) -> str:
    rem = ''
    for i in text:
        if i.isalnum() == False and i != " ":
            text = text.replace(i, '')
    return text

X = X\
    .apply(remove_special)

X = X\
    .apply(lambda text: text.lower())
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\wpegu\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\wpegu\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [ ]:
```
def remove_stopwords(text:str) -> list:
    stop_words = set(stopwords.words('english'))
    words = word_tokenize(text)
    return [w for w in words if w not in stop_words]

X = X.apply(remove_stopwords)
```

In [ ]:
```
def extract_stem(text:list) -> str:
    ss = SnowballStemmer('english')
    return " ".join([ss.stem(w) for w in text])

X = X.apply(extract_stem)
```

Now that the explanatory variable has been cleaned, one can now vectorize the reviews and start the creation of the data model.

Data Modeling

In [ ]:
```
tfidf = TfidfVectorizer(
    min_df=5,
    max_df=0.8,
    sublinear_tf=True,
    use_idf=True
)

text_count_matrix = tfidf.fit_transform(X)
x_train, x_test, y_train, y_test = train_test_split(
    text_count_matrix,
    y,
    test_size=0.30,
```

```
    random_state=2
)
```

Now to create and train the model

In [ ]:

```
classifier_linear = svm.LinearSVC()
classifier_linear.fit(x_train, y_train)
```

Out[ ]:

```
LinearSVC()
```

Now that the model has been created, predictions can be made and said predictions can then be utilized to draw conclusions about the efficiency of the model at classifying the reviews.

Analysis of Results

In [ ]:

```
y_pred = classifier_linear.predict(x_test)
report = classification_report(y_test, y_pred, output_dict=True)
```

```
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classific
ation.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classific
ation.py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in
labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classific
ation.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classific
ation.py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in
labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classific
ation.py:1318: UndefinedMetricWarning: Precision and F-score are ill-defined and being set to 0.0
in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classific
ation.py:1318: UndefinedMetricWarning: Recall and F-score are ill-defined and being set to 0.0 in
labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

From the above, it seems that the of the system to predict the usefulness of the review to be very low. Looking at the statistical analysis done below can paint a better image of how the model performs in a more general sense.

In [ ]:

```
print(f'Accuracy                          :{accuracy_score(y_test, y_pred)}')
print(f'error rate                        :{1-accuracy_score(y_test,y_pred)}')
print(f'precision                         :{precision_score(y_test,y_pred, average=None).mean()}')
print(f'recall                            :{recall_score(y_test, y_pred, average=None).mean()}')
```

```
Accuracy                 :0.287868689843302
error rate               :0.712131310156698
precision                :0.6586362045609665
recall                   :0.6098811670101787
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classific
ation.py:1318: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels wit
h no predicted samples. Use `zero_division` parameter to control this behavior.
```

```
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\wpegu\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Recall is ill-defined and being set to 0.0 in labels with no true samples. Use `zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

From the statistical analysis above, it can be shown that the model is not very good at predicting the usefulness of the reviews (it's accuracy is lower than flipping a coin). In the end, SVMs may not be the best for sentimental analysis

Source(s):

1. Urooj, Wajiha. “Support Vector Machine in Python.” Edureka, 13 May 2020, medium.com/edureka/support-vector-machine-in-python-539dca55c26a. Accessed 17 Feb. 2022.
2. Vasista Reddy. “Sentiment Analysis Using SVM.” Medium, Medium, 12 Nov. 2018, medium.com/@vasista/sentiment-analysis-using-svm-338d418e3ff1.