Restaurant Review Analysis and Classification Using SVM

Veda Waikul¹, Onkar Ravgan², Aruna Pavate³

¹²³(Computer Engineering, Atharva College of Engineering/ Mumbai University, India)

Abstract: The paper here presents a classification machine learning model to classify restaurant reviews. The reviews can be anything which are related to the food of the restaurant, staff of the restaurant and also overall review of the restaurant. The model uses Support Vector Machine (SVM) algorithm for classifying the reviews. The classified reviews are helpful for the restaurant to analyze their shortcomings in different areas and improve the quality of food and service in the restaurant. The reviews are stored on the cloud and can be accessed by the admin.

I. Introduction

In today’s world technology and automation in every sector is rapidly increasing. People rely more on mobile devices for almost every task in their day to day lives. Restaurant Business is a sector which has a very large scope in automation and use of technology. At such times waiting for the waiter to take orders, delivering food, lengthy queues, etc. can be very displeasing for the customers of the restaurant.

To overcome these problems a concept of automated restaurants using a system which uses LCD displays, mobile/tablet devices and a system for the chef to interact with customers is proposed. Using Machine Learning and Data Science predictions are made based on the reviews and other data of the customer can help make the dining experience better as well as it will help the restaurant to manage and make the restaurant grow.

In a restaurant while placing order, the customer has to ask the waiter whether a particular food item is available or not and after that he/she has to give the order. As well as several times it happens that customers have to wait for the waiter to come to their table which is sometimes frustrating. Storing the statistical data of the restaurant is a very tedious task. There is need of managing the data of inventory, customer orders and reviews, staff, payroll.

II. Material and Methods

Dataset Description

The Dataset contains 1000 reviews in text format in a “.tsv” file which is taken from www.kaggle.com. This dataset is used for training the SVM classifier. It is split into 70% training data and 30% test data. The dataset contains two columns. First column contains the text reviews given by different users which are related to the food of the restaurant as well as the overall review of the restaurant. The second column contains the sentiment i.e. if the review is positive or negative. 1 indicates that the review is positive and 0 indicates the review is negative.

The dataset is imported and converted into a Pandas Dataframe. The model should predict if the review is positive or negative.

The dataset is cleaned from 1000 reviews and the reviews which are not proper are discarded from the dataset and then the Dataframe is then served as input to the Count Vectorizer for further processing.

Support Vector Machine (SVM) Algorithm

The research has proposed a Machine Learning model which will help in classification of reviews of the restaurant as well as classification of reviews of food served by the restaurant. This model is trained using SVM (Support Vector Machine) Algorithm to classify the reviews.

Consider a set of training feature vectors \( \mathbf{x}_i \in \mathbb{R}^D \), \( i = 1, \ldots, t \), and the corresponding class labels \( y_i \in \{+1, -1\} \) (for the binary classification). Vectors with the class label +1 (Positive Review) are the positive ones (class \( C^+ \)), whereas the others (Negative Reviews) belong to the negative class \( C^- \).

Linear SVMs separate data in the D-dimensional input space with the use of the decision hyperplane defined as:

\[
\mathbf{f}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = 0,
\]

(1)

Where \( \mathbf{w} \) is the hyperplane normal vector, \( \mathbf{x}_i \in \mathbb{R}^D \), and \( b/||\mathbf{w}|| \) is the perpendicular distance between the hyperplane and the origin (\( ||\cdot|| \) is the 2-norm), \( b \in \mathbb{R} \). This hyperplane is positioned such that the distance between the closest vectors of the opposite classes to the hyperplane is maximal.
For two linearly separable classes (as already mentioned, with the class labels $y_i \in \{+1, -1\}$, the training data must satisfy the following conditions:

$$w^T x_i + b \geq 1 y_i = +1$$  \hspace{1cm} (2)

$$w^T x_i + b \leq -1 y_i = -1$$  \hspace{1cm} (3)

which can be re-written as:

$$y_i (w^T x_i + b) - 1 \geq 0 y_i \in \{+1, -1\}.$$  \hspace{1cm} (4)

The equalities from Eq. (4) hold for the vectors positioned on two parallel hyperplanes, with the distance to the origin given as $|1-b|/||w||$ and $|-1-b|/||w||$, respectively. There are no vectors between these two planes, and the distance between the separating hyperplane and each of these planes is $1/||w||$. Hence, the maximal theoretical margin possible to generate by the decision hyperplane is

$$\phi(w) = \frac{2}{||w||},$$  \hspace{1cm} (5)

Since we intend to maximize the separating margin, the value of $||w|| = \sqrt{w^T w}$ should be minimized:

$$\min_{w,b} ||w||.$$  \hspace{1cm} (6)

To simplify the calculations, it can be given as the quadratic term:

$$\min_{w,b} \frac{||w||^2}{2}.$$  \hspace{1cm} (7)

The optimization is performed with respect to the constraints in Eq. (4)—it becomes a quadratic programming (QP) problem. This formulation of the problem is called the primal form. The resulting hyperplane is exploited to classify the incoming data based on the decision function

$$f(a) = \text{sgn}(w^T a + b),$$  \hspace{1cm} (8)

where $a$ is a feature vector to be classified.

If we re-write Eqs. (4) and (7) to get the Lagrangian in its primal form, we have

$$L(w, b, \alpha) = \frac{||w||^2}{2} - \sum_{i=1}^{l} \alpha_i y_i (w^T x_i + b) + \sum_{i=1}^{l} \alpha_i,$$ \hspace{1cm} (9)

where $\alpha_i$ are the Lagrange multipliers. This transformation allows for representing the constraints given in Eq. (4) as the constraints on the Lagrange multipliers. In this formulation, the data in both training and test sets will appear in the form of the dot product between the vectors.

Since retrieving the SVM hyperplane is a convex optimization problem, determining the hyperplane is equivalent to finding a solution to the Karush–Kuhn–Tucker (KKT) conditions. The KKT conditions for Eq. (9) are:

$$\frac{\partial}{\partial w} L(w, b, \alpha) = w - \sum_{i=1}^{l} \alpha_i y_i x_i = 0$$

$$\frac{\partial}{\partial b} L(w, b, \alpha) = -\sum_{i=1}^{l} \alpha_i y_i = 0$$

such that
\begin{equation}
y_i (w^T x_i + b) - 1 \geq 0 \quad i = 1, 2, ..., t
\end{equation}
\begin{equation}
\alpha_i \geq 0 \quad \forall i
\end{equation}
\begin{equation}
\alpha_i (y_i (w^T x_i + b) - 1) = 0 \quad \forall i
\end{equation}

Incorporating the equation for \( w \) from Eq. (10) into Eq. (9)

\begin{equation}
w = \sum_{i=1}^{t} \alpha_i y_i x_i
\end{equation}

and knowing that

\begin{equation}
\sum_{i=1}^{t} \alpha_i y_i = 0
\end{equation}

we have

\begin{equation}
L_0 (\alpha) = \sum_{i=1}^{t} \alpha_i - \frac{1}{2} \sum_{i=1}^{t} \sum_{j=1}^{t} \alpha_i \alpha_j y_i y_j x_i^T x_j
\end{equation}

where \( LD \) denotes the dual form of the Lagrangian. The dual problem may be solved by maximizing \( LD \) with respect to \( \alpha \), subject to the constraints given in Eqs. (11)–(13) (this is the Wolfe dual of the problem). Only a small subset (containing \( s \) vectors) of the entire T (i.e., SVs) contributes to the position of the hyperplane. The Lagrange multipliers \( \alpha_i \) corresponding to the SVs are greater than zero. Finally, the decision function becomes:

\begin{equation}
f (a) = \text{sgn} \left( \sum_{i=1}^{s} \alpha_i y_i x_i^T a + b \right)
\end{equation}

In order to apply the above reasoning for non-separable cases, it is necessary to relax the constraints given in Eqs. (2) and (3), and to introduce an additional cost of this operation:

\begin{equation}
w^T x_i + b \geq 1 - \xi_i y_i = +1
\end{equation}

\begin{equation}
w^T x_i + b \leq -1 + \xi_i y_i = -1
\end{equation}

\begin{equation}
\xi_i \geq 0 \quad \forall i
\end{equation}

where \( \xi_i \) denotes a positive slack variable. The objective function should be modified to take into account the classification errors:

\begin{equation}
\min_{w, b, \xi} \frac{||w||^2}{2} + C \sum_{i=1}^{t} \xi_i
\end{equation}

such that

\begin{equation}
y_i (w^T x_i + b) \geq 1 - \xi_i \quad i = 1, ..., t
\end{equation}

\begin{equation}
\xi_i \geq 0 \quad i = 1, ..., t
\end{equation}

where \( C \) is the parameter that controls the trade-off between the margin and the slack penalty (the larger the value of \( C \), the higher penalty to the errors). Considering this trade-off allows for introducing the soft-margin SVMs. As in the separable case, Eq. (21) can be easily transformed into its Wolfe’s dual form:

\begin{equation}
L_0 (\alpha) = \sum_{i=1}^{t} \alpha_i - \frac{1}{2} \sum_{i=1}^{t} \sum_{j=1}^{t} \alpha_i \alpha_j y_i y_j x_i^T x_j
\end{equation}

It is to be maximized, subject to

\begin{equation}
0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_{i=1}^{t} \alpha_i y_i = 0.
\end{equation}

Finally, we have

\begin{equation}
w = \sum_{i=1}^{t} \alpha_i y_i x_i
\end{equation}

As in the separable case, we can retrieve the Lagrangian in its primal form:

\begin{equation}
L (w, b, \alpha) = \frac{||w||^2}{2} + C \sum_{i=1}^{t} \xi_i - \sum_{i=1}^{t} \alpha_i [y_i (w^T x_i + b) - 1 + \xi_i] - \sum_{i=1}^{t} \mu_i \xi_i
\end{equation}

where \( \mu_i \) enforces the positivity of \( \xi_i \). The KKT conditions can be retrieved for the non-separable case following the reasoning presented for the separable one.

### III. Result

The SVM model is trained on the dataset and the model is deployed using flask on the server and the positive and negative results of the review is stored in the database.

Table no 1 Shows a snapshot of the restaurant reviews dataset. Upon careful inspection and analysis of the dataset it was found that the few of the records in the dataset was not related to the restaurant reviews. Therefore the data was cleaned and relevant reviews were kept.

| Table no 1: Snapshot of the dataset. |
|---|---|
| Text Review | Sentiment |
| Wow... Loved this place. | 1 |
| Crust is not good. | 0 |
| Not tasty and the texture was just nasty. | 0 |
| Stopped by during the late May bank holiday off Rick Steve recommendation and loved it. | 1 |
| The selection on the menu was great and so were the prices. | 1 |
Follow up after 1 week
Table no 2: shows the results of accuracy, precision and recall of the SVM classifier in classifying the results. The SVM model shows the highest accuracy amongst the other model like naïve bayes, etc. Hence this model is suitable for the review classification.

<table>
<thead>
<tr>
<th>Table no2: Analysis of the model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
</tr>
<tr>
<td>Precision</td>
</tr>
<tr>
<td>Recall</td>
</tr>
</tbody>
</table>

Table no.3: shows the confusion matrix generated based on the training and test data after training the SVM classifier on the data. In the confusion matrix the number of correct and incorrect predictions are summarized with count values and broken down by each class.

- There are 115 True Positive (TP) values. TP means that the observation is positive, and is predicted to be positive.
- There are 37 False Negative (FN) values. FN means that the observation is positive, but is predicted negative.
- There are 116 True Negative (TN) values. TN means that the observation is negative, and is predicted to be negative.
- There are 32 False Positive (FP) values. FP means that the observation is negative, but is predicted positive.

<table>
<thead>
<tr>
<th>Table no3: Confusion Matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 1 Actual (Positive)</td>
</tr>
<tr>
<td>Class 1 Predicted (Positive)</td>
</tr>
<tr>
<td>Class 2 Actual (Negative)</td>
</tr>
</tbody>
</table>

IV. Conclusion
Thus, in this project an efficient and user-friendly method is proposed which will provide automated systems in the restaurant and solve problems faced by the restaurants using technologies like Android, Web Development and Machine Learning. Interactive User Interfaces for the customers and restaurant staff will be provided and customers can order food directly through the module without interacting with waiter. Using Machine Learning Models prediction of the food preferred by the customers and also information necessary for the restaurant to grow in business through customer reviews and other data. The system saves a lot of time of the customer as well as the restaurant staff and helps the restaurant in many ways.

References