Sentiment Analysis

# Part-1

## Goal

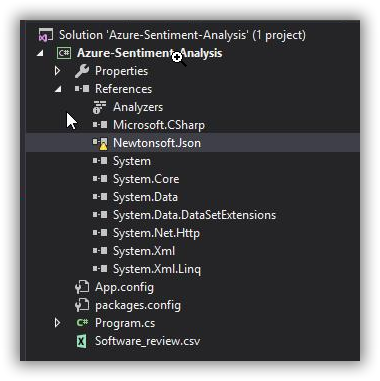
Find the sentiment score of existing feedbacks quickly and filter out good and bad comments.

## Steps

1. For a quick solution, we are going to use Azure Cognitive Service.
2. Download this sample C# console app - <https://tinyurl.com/qmhchc4>
3. Replace with this access key in **Program.cs**

f3481f9fce694a8381a3002a33207a50

1. Build the app to get the packages downloaded.



1. Run the app. You will find the sentiment score of the comment present in “**Review\_Text**” column of Feedback file – “Software\_review.csv”
2. Imagine that, like this we have analysed the sentiment score of all the comments we have received. And it is saved as additional column “**IsNegative**” in each row.

So now we have a dataset of all the feedbacks along with the sentiment analysed for the comment mentioned by user.

# Part-2

## Goal

With the analysed dataset, deploy a dashboard that helps to take decisions for stake holders.

## Steps

1. **CONNECT –> 2. VISUALIZE –> 3. SHARE**

This format of working with a business intelligence tool will be almost common for any BI tool - Bold BI, *Power BI, Tableau, Qlik, Sisense*.

### Connect

1. I hope all of you would have created a tenant or got added into <https://webinar.boldbi.com>
2. Set password (ignore if already did).
3. Login to your private tenant or my tenant.
4. **Connect** to Microsoft SQL Server as **Live** mode.  
   <https://help.syncfusion.com/bold-bi/data-connectivity/sql-data-source>
5. Credentials of SQL Server:

|  |  |
| --- | --- |
| **Server name** | webinar-boldbi.database.windows.net |
| **Username** | BoldBIReadOnlyUser |
| **Password** | BoldBIDemo@123 |
| **Mode** | Live |
| **Database** | Webinar-boldbi |
| **Table** | Software\_review |

1. You have completed the first part - **enabling connection to data source**.

### Visualize

1. Next to visualize the data, create a new dashboard.  
   <https://help.syncfusion.com/bold-bi/bold-bi-walk-through>
2. You will be having a blank dashboard. Now let us populate some widgets and configure data to the same.
3. Add a filter widget ‘Combo Box’ and bind ‘Rating’ column into it.
4. Similarly add another ‘Combo Box’ and bind ‘Sentiment’ column into it.
5. Create a column chart to list down *number of feedbacks ordered by rating.* Move the column ‘’ into Values and column ‘’ into Measures.

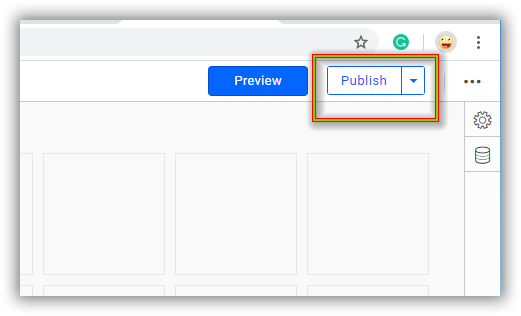
<https://help.syncfusion.com/bold-bi/visualize-data/configure-widgets/column-chart#how-to-configure-the-table-data-to-column-chart>

1. Add a Grid widget and populate Customer name, Order ID, Comments, Rating and Sentiment  
   <https://help.syncfusion.com/bold-bi/visualize-data/configure-widgets/grid>

You have completed the second part – **Visualizing the data**.

### Share

1. Save the dashboard by publishing it.



1. Now the dashboard is available, and you can apply filters in Rating, Sentiment combo boxes and workout all the features like sharing dashboards, exporting the charts, etc.

With that completed the third part – **Sharing the dashboard and insights**.

# Part-3

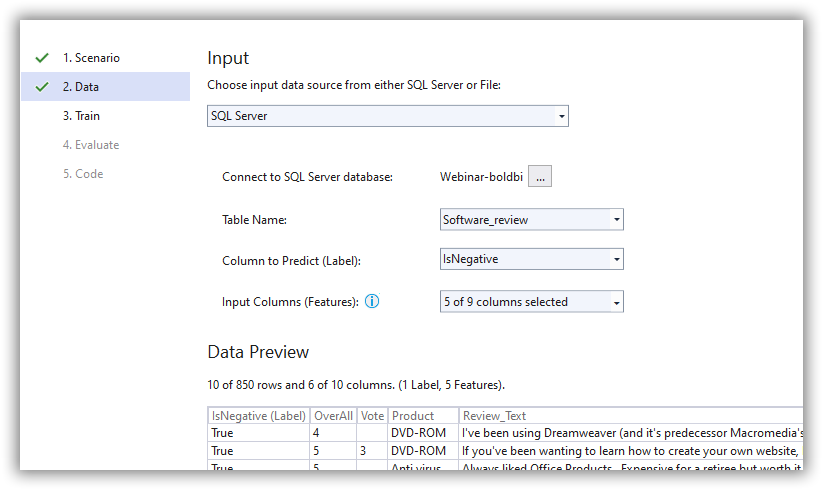
## Goal

Stackholders are satisfied with the sentiment analysis result and dashboard. And further requests *the Dev team* to deploy this as a pipeline, by creating own machine learning model to analyse the sentiment score.

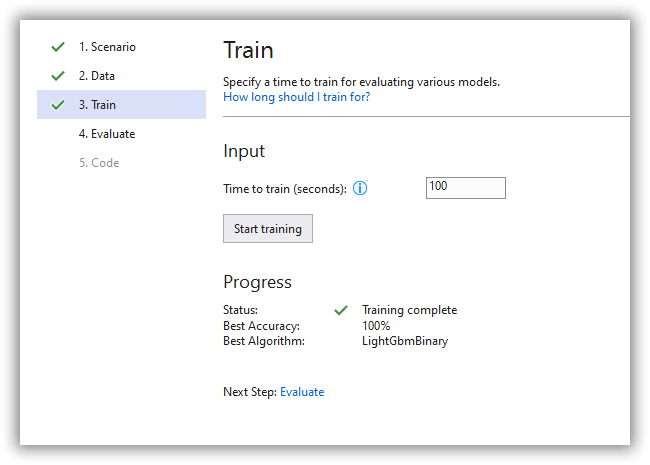
### Steps

Having the data and its defined sentiment result, dev team decides to deploy a new machine learning model. Not due to knowledge gap, but to implement it quicker, they choose ML.Net

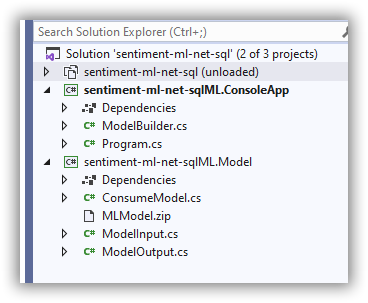
1. If you are having Visual Studio 2017 15.9.12 or later, follow further to deploy your own new Machine Learning model for sentiment detection which can be used instead of Cognitive Service.
2. Follow the instructions and install ML.Net Model Builder extension  
   <https://dotnet.microsoft.com/learn/ml-dotnet/get-started-tutorial/intro>
3. In “Download and add data” – instead of file, you can choose the SQL Server and connect to the same ‘Software\_review’ table which you connected while creating dashboard.
4. Under **Column to predict (Label)**, select "IsNegative".
5. **Input Columns** – Uncheck columns ‘Verified’, ‘Review\_Time’, ‘Reviewer\_Id’, ‘Reviewer\_Name’.



1. Set the Time as 50 seconds.



1. Proceed Evaluate and Code.
2. Once the new test app and ML model is deployed, “Unload” the actual console app to overcome the assembly mismatch error.



1. Run the new console app and see that it has evaluated the first row from data with the new Binary-Classification model created.
2. Now you can publish the Model as package (Nuget in case of Windows) and deploy it in any application and run in Windows / Linux / Mac OS. Because this deployment is in .Net Standard