

**Classification of Movie Reviews**

**Using Sentiment Analysis**

**Application of Data Mining Tools & Techniques**

**DATA MINING - B9DA103**

**TERRI HOARE**

Brijith Elizabeth Joseph

STUDIENT ID: 10393634

DATE: 22-04-2019

Contents

Introduction ……..……………………………………………………………………… 3

Business Understanding ……...…………………………………………………… 4

Data Understanding …………………………………………………………………. 4

Data Preparation ………………………………..……………………………………. 5

Modeling ………………………………………………….……………………………… 9

Evaluation ……………………………………………………………………………….. 14

Deployment …………………………………………………………………………….. 15

Conclusion ………………………………………………………………………………. 21

References ………………………………………………………………………………. 21

**Introduction**

Data mining is the process of discovering patterns in large data sets involving methods at the intersection of machine learning, statistics, and database systems. Data mining is an interdisciplinary subfield of computer science and statistics with an overall goal to extract information from a data set and transform the information into a comprehensible structure for analytics. Data mining is the analysis step of the "knowledge discovery in databases" process, Aside from the raw analysis step, it also involves database and data management aspects, data pre-processing, model and inference considerations, interestingness metrics, complexity considerations, post-processing of discovered structures, visualization, and online updating.

Data Mining is supposed to be the process of exploration through data in order to understand and predict new algorithms to understand and predict the future. It helps in taking a step ahead for analyzing the future. The process of data mining contains three important major steps, which is considered to be foundation. They are Statistics which is the numeric study of data relationships, Artificial Intelligence which is the human-like intelligence displayed by software and/or machines, and Machine Learning which is the algorithms for understanding the data for the purpose of prediction.

There has been a large amount of prior research in sentiment analysis, especially in the domain of product reviews, movie reviews, and blogs. Sentiment analysis works are mainly focusing on designing platform or tools to do automatic sentiment analysis using models from machine learning area such as latent semantic analysis (LSA), Naive Bayes, support vector machines (SVM) etc. The Rotten Tomatoes movie review dataset is a corpus of movie reviews used for sentiment analysis, originally collected by Pang and Lee In their work on sentiment treebanks, Socher et al.used Amazon's Mechanical Turk to create fine-grained labels for all parsed phrases in the corpus. We classified the reviews as negative, neutral, positive based on the words used.

The whole idea of this analytical process was carried out with the CRISP-DM data modeling steps. We started to build ahead by splitting the whole data mining process into six steps, and these are as follows,

* Business Understanding,
* Data Understanding,
* Data Preparation,
* Modeling,
* Evaluation,
* Deployment.

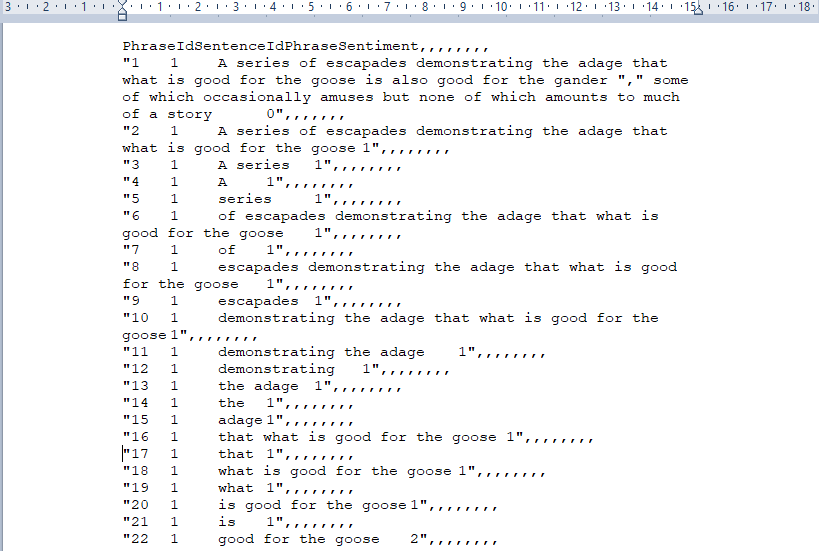
**Business Understanding**

Movie review classification is the model which has been created to identify the sentiment from the given text. It is defined as a process in which the text mining techniques are extracted and further it is being studied ahead to understand the polarity of the given document or text. In this implemented principle, we are mainly focusing in understanding how accurate it is when we compare it with the human judgments and other conclusions to understand the similarity and focus. These understanding and comparisons are mainly calculated by different measures which is based on precision and recall oven three variables which are, Negative, Positive and Neutral.

While we consider from a business point of view, the final output from the prepared data can be helpful in many ways. it could be helpful in making a decision, it could be helpful in understanding end result of a political or a social media campaign etc. Here in this assignment, we are mainly focusing on implementing the sentimental analysis on movie sentiment classification

**Data Understanding**

We got the dataset from <https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews> . it is extracted from Rotten Tomatoes . The dataset contains the SentenceId , phraseid ,phrase and their associated sentiment. The training dataset selected contains 156060 records. And 66291 records were chosen for testing. The objective is to assign a sentiment label for each phrase. From this, we took only the phrase for analyzing. SentenceId and phraseid were not used because they were not having any noticeable relation with the sentiment of the phrase. In sentient there were 5 different values as 0,1, 2,3 and 4, in which 0 shows negative, 1 represents somewhat negative, 2 represents neutral , 3 represents somewhat positive and 4 represents the phrase is positive. To make the prediction more accurate we selected only the positive negative and neutral phases for classification



**Data Preparation**

The data which we used need filtering and should get prepared for processing. The data was not a clean version and filtering was needed. First the special characters like ‘, . - were removed. All the words were converted to lower case as the next step. Phrases that are repeated (such as short/common words) are only included once in the data.

![A picture containing screenshot

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDuRXhpZgAATU0AKgAAAAgABAE7AAIAAAAMAAAISodpAAQAAAABAAAIVpydAAEAAAAYAAAQzuocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEJoYXJhdGggTSBSAAAFkAMAAgAAABQAABCkkAQAAgAAABQAABC4kpEAAgAAAAM4NwAAkpIAAgAAAAM4NwAA6hwABwAACAwAAAiYAAAAABzqAAAACAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAMjAxODoxMjoxNCAwMToyOTo1NwAyMDE4OjEyOjE0IDAxOjI5OjU3AAAAQgBoAGEAcgBhAHQAaAAgAE0AIABSAAAA/+ELHmh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMTgtMTItMTRUMDE6Mjk6NTcuODY1PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPkJoYXJhdGggTSBSPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAKUEHQMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APpGiiqGs3UtnpjzW7BXBABIz3oAv0VkveXNpe20cs3npOpJG0Arx7VNbavHPZvdGJ44VBO5sc+1AGhRVGHVYpGZZUaEqm8B+6+tEGprOy5glRHUsshGQR+HSgC9RVCLV4pZUXY6pIxVJD0Yimx6xHLJiOGRk37C64OD9OtAGjRWc2sRieSOOKSQRttcrjg/TqaXWbyW003zrdtrllAJGeD9aANCisz7XNaajDBcy+akykq20Agj6U+HV4ZZkXY6pIxVJD0YigDQoqguqo8h8uGR4w+wyKMjP0601tYiW3upTG+LZ9jDjk5xxQBo0VmHXIw0yiCQ+SoZiMdCM0+LWIZJokKOizKWRz0OKANCiqEWrxSyoux1SRiqSHoxFRLrsTQ+b5MgTzPLJOODQBqUVkW2sOReSXUZSOB8DGDjtj61btNQF1Js8l0yu4McEEfUUAXKKyJdQujrL2ygpFFHvbAUkj15qeHV45bFrt4njiAyC2OfagDQorO/tmJY5WkidHjTfsOMkeopF1qNniXyJd0ybkHHNAGlRVazvUvbdpIwV2sVIPYisqC/vW0k37zhgjkNHsAGM4oA3qKoTaosMQk8lmXYGJyBjIz36/hVeHV5LjVkhjiPkNFvzxkj1oA16Kz21eOOaNJYnQSNtViRz+HUVoUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVn23/AB/33/XRf/QBWhWfbf8AH/ff9dF/9AFAGhVHWLSW9014IMbyQRk471eooAqRafBDHmOICTZjJJPb3qnb6VL/AMI+bGYhZCScg5HXIrXooAxxpc9zKz3W2PEHkqFOc+9TWlvexwJayCNYUjKFgclv8K0qKAMS00iSCSJHhiZYn3eazEk85GBng0kmk3DXW+JEhbfkyxtjI/3a3KKAMW80qea5d4kjRmYFZUbaR9R3q1qtlNd6WIIsNICpJJxnFaFFAGY1jPeXqT3KrEsSFVUNu5PeoINKuFNtDLtENtIXVgeW5z0raooAy7S0vbHdBAIzG0hfzGPY9sVWn0q7MN9FFsK3Em8En3zit2igDETSrkfb8hf38SonzdwMUv8AZVwzadnaBBGUk56ZGOK2qKAMS00iSCSJHhiZYn3eazEk85GBng03+yLr+yPs+E8z7T5n3uMVu0UAZUOnTRfblKxstw+9S3I+hFJp2nT216ZSBFFtwYw+4E+vtWtRQBmvYzNq9xcDb5clv5a885qJNKlbw6LGUhZBk8HIznIrXooAxJtLurnz5ZAiytAIkUNwffNTJp063mnSELtt4dj898YrVooAz9Ks5bO3nSbGXlZxg54NUoNNvV0s2Dxoqs+Wk354znpW7RQBi3OmXLTS+WFljaERx7mxs4xRZaXc211BKdp22/lNhvun1962qKAOe/si8KwK6ozRz+Y0hflhmuhoooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArPtv8Aj/vv+ui/+gCtCs+2/wCP++/66L/6AKANCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKhurf7VbmLzpYckfPC21h+NcfoOqXUHhi+1q8urm8ktpJEEUkmFIBGO3X3rlq4hUpqMlo03f03OujhnWg5ReqaVvXY7aisFfEFz/AGJFfzWUcfnhTGpm+XBGck44/I1Qh8cG4hsjb2IeS5uWtivm4CsMcg4561MsbQja738mVHAYiV+WO2m6/wA/I62iuWfxk0VnePNYEXFpcCB41fcuTnnOOnHpVDWPF1+/h25nsoI4HjdVMyS7hg45XgEnn2rOeY4eMW73sr7Pp/wxpDLcTOSja13bddf+HO4orl01y5sPs+nJC15ftB9olEs/Cr/vbefypE8ZG7udNh02zExvkZhvk27CvUdD6Vf16itG9fR+X+aI+oV3rFXXe689fwZ1NFcrdeLruG91OCLTUkXTUDysZ8ZBGePlqaz8Vvcahp8E1l5UWooXgcSZPAzyMcULHUHLlv5bPvbt30B4DEKPNbS1912v37anSUVzmt3t1J4l0/SLadrdJ0MskiHDEDPAP4VadrrQ7GWSe6N8pcCPzflKf7zc5H4Vf1hOUlbSO7+V/Uj6s1GLvrLZfO3obNFckvjkNpktx9izJFceQyq5KA/3t2M4/Cp18YomkXN7dW2zyWCr5cm5ZCemDgH9KhY/DPaXS/U0eX4lbx626bnTUVzFr4xE0F20tm3mW8XmAREsrjIGMkDnn0qfQ/Ez6vdiF7VEDJuDxy7gPZsgYP51UcbQnJRjLV+pMsDiIRlKUdFvsdBRRRXWcQUUUUAFFFFABRRRQAUUV5p8SZNR03xD4dOna3qVqmrakltPFFMAipwPlGODQB6XRXFat4ul0DVrfw3pVrLq+oR2puZWurjaRED94sFOT+ArkNW+KyDxJ4Y1iC6uIdGuLOeS5tR0d1DAA/RsDNAHslFcHqHxL/szwtpmqXlhGlxqshW1g88lcerMFyOvYGs+f4umLQEvY9HMlx/aEdi8XmlUO8MQysVyw+X0FAHplFeU+J/ibcWum69Z3unT2d7pTQMws73aWWRlxhynHXkYrZ1bxzrWk6tp+nxaJa3hviotwt+fNK4GWZfLwOvrQB3tFeb6p8Xrex1y8s7XT/tMFjOIJ2EjeYWzg7UCkED3I6V6JbzLcW0UyAhZEDgEYOCM0ASUUUUAFFFFABRRRQAUVxHxd1a/0T4c3l9pN1JaXSSxBZYzhgC4B/SsDVPEE+g+EI76QeJS811bRk31ykRO4NnYQG49RjnigD1aivArzx94us9a1G3tLkTwHxILOMzS4KruOIgMcKcdefpXca/8V00bW7nTIdPW4nso1a6XzWBDEZ2phDuP1xQB6LRXCWnxEutV8YRaJpGjrKj2Ud6Zp5zGVRsZBXaeRn1q18R7nXrTSrOTQRP5IuR9ua1/1qw8ZKe/WgDsaK8W1vxlq+l+Ezd+HddOpxS30cLm7BS4tM5+Q9d2cHnjGOhzXZat4+vLDU5dMtNIjur2ysheagguSqwrgE7Tt+br6CgDt6K88PxQmvta06w0DR1vft+nG+jeS48vGGZSpG09160tr8WLS4sdJvXsvKtb/wA1JJGl/wBTIgzt6c5weeKAPQqK830X4uwa39hjt9O23N1f/ZTC0vKJwfM6f3TnH61N/wALUXf9u/sz/iR/bfsP23zvn8z/AHMdPxoA9CoriNF8fXWranrqHSY4bDRJ5Yp7j7QSxCBjkLt746Zrmb3x9rWr6x4NurWB9O03UdQZB5c277TGCBhhgY78c9aAPXaK830z4ryalrMdqujhYnuDCVM+J0H94xlQMf8AAq2PiHr91pFppdjp8hhutVvo7VJQeUBYZI98GgDsKK8h0zVtVvviH4ksby78Q3NvY3MaQJprpsjBBzvz2OP51pXXxYu7eHXbpdCR7LRL77JPL9qwxG7buA29fbP40AemUVDaXC3dlBcoMLNGsgB7AjNTUAFFFcH8X9X1DRfBkFzpN3NazPfxRs8BwxU7sgUAd5RXl2sa/LoFrpO5fEgN5q0Vv/xMLlI2wR/shty+o4+tcrpPj3xh/aFjbRzJdx3PiCa2Vp5yGcLs/dfdO1eevPXpxQB73RXnOs/FqHTdevbC208XKWEgjuD5jB92cEIoUhse5FaGm+PLzWPG91oWnaQjQ2sUM8lzLcFWCSKrfd2nkbumaAO2orkPiZ9qtfBN/qun6leWNzYwNJH9mkCq5yPvDHNZdn4mPhX4f6Pqd9c3mq32sLAIo7mYEGV1HAOPlXn3oA9Dorx/xt8R9QbwhqkNsJNH1rTLyCO4WGXeAr88NgZz9K6fTfiXbX2jaxrAtsaXpYA8/wAz55W6fdxwM980AdzRXm2nfFsXkN20ulgNFYyXkPlSsysEUttZig2nA7ZqTTfijc3M2g/b9EFtb65G7W7x3G9gV6gjaPXrmgD0WivONE+K0msaxDajR/LimmMWPP8A30eM4ZkKgAfRjWv4912606TRdL0+VoLnV71bcSqeUXjOPzoA7CivIrPx7rejax4yuLuF9S07TNQVD5k+37NGSR8owc9uOK2z8VYodP1ie807yZbGOOWCLzc/aEkUFD04zuA74oA9Corzv4jeIdXsvh7pmpWjyaZeXF7AsixPkqrZyucc9q7DxJr1t4Z8PXWr3oYw2y5IXqSTgD8yKANSiuH8I/EYeJdZ/s6aw8h3h86KSF2kQj0JKrhvbmu4oAKKKKACiiigAooooAKK8o8Q6vrNn4w1KPXNWv8ARtPJUabdW6boAOc+YO/arOjeMNd/4WRqVhqL20+l2GmC4eSKQjKbQwlAxyW/u54z1OKAPTqK840P4tRazq1rbjTttveOyQyxyM7KQcDeNoCg+xNUbf4w38uh2msS+HY0sbq9+xIy3hL+ZzjjZ049aAPVaK5Xw14xn1nxLqmhajpy2V9pyo7hJvMVlYAjnA556Vyfifxbqs9x40Om3klrFoNmsaCM4PmsT834AfrQB6tRXj3hTxTqX9s6RNFqt9qenNpZudWa5ywgkEe47Tgd66PS/ia15eaYbzS/s1hrDumn3Am3NIynHzLgbfzNAHfUV5NefEzxDqvw01HxFpOj21nFCSnnPdlmQhgpIXYM9fUVpXPxJn0PRdBg1C0hk1bULNZ2Vp2EezH3iwQnJxnGOCevegD0eivOG+Kss66CmnaK0s+rzyweXNMY/LdNv+ycg7uuBWmfEN9p/wATLDSb8/u9WsjIkIfcsMiD5sHAyODQB2lFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAVn23/H/AH3/AF0X/wBAFaFZ9t/x/wB9/wBdF/8AQBQBoUUUUAFFAIYZByPUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFYcHheCDw7d6QtxIY7l2ZpCBld2P8K3K5bU9c1VNfmsdO+zLHFGrEyoWJJz6EelcmJ9krOor3uvv3/I68N7V3VOVrWb+W35mheeGoLzSLGxeaRRZFDHIBySoxyOlU7TwXb2slq4u5WNvdtdDKj5mOOPpxVb+1fEf/AD0sP+/Lf/FUf2r4j/56WH/flv8A4quSSw8pczpO/wDl8zsjPERjyqqrf579DR/4ReENqDLdzo19MJiUONhGeOOo56Gqkvga0ms7uKS5k8y7Kl5ERVxtIIwo47VD/aviP/npYf8Aflv/AIqj+1fEf/PSw/78t/8AFUpRw8lZ0n1/G9+vmxxqYiLuqy6fha3TyRq33hyK6uorqG5ktrhIvJMiAEsnoc1HbeFLOzvtOubeR0FgjIqY4fdnJP51nf2r4j/56WH/AH5b/wCKo/tXxH/z0sP+/Lf/ABVU/YOXN7J3/rz8kQpV1Hl9qrbf1p5s0ZPC8Elxq8puJAdUjCOMD5MDHFEfheCO40iUXEhOlxlEGB8+Rjms7+1fEf8Az0sP+/Lf/FUf2r4j/wCelh/35b/4qj9xe/s3/Tv376j5q9re1X9Ll7dtDc1PRYtRuILlZXgubc/u5UGSPbB61TuPCyXmny215f3M7SSrL5jN90jphegHNZ/9q+I/+elh/wB+W/8AiqP7V8R/89LD/vy3/wAVTk6M23Km9d/6uTD20ElGqtNv6sXYPCMUFvdxpezhrmfzy6/Lg+mBwR7Gmx+C7EWt3FM7SNdkF3ChMEdMAcCqn9q+I/8AnpYf9+W/+Ko/tXxH/wA9LD/vy3/xVRy4b/n0/wCvn5mntMR/z+X/AA3y8jXg0JkjnW51G6naVPLDbyuwewHGfeotO8Lw2GqJfvcNNNGhRSUC8e+Op9zWb/aviP8A56WH/flv/iqt6HrWo3Osy2OpCBtsQkVoVK9/cmtIug5xTg1rpfv95m/b8k+Wonprbtt2Ojooor0zywooooAKKKKACiua1DV9V/tq6tbCS1jjt9n+tiZySy57MKg/tLxD/wA/On/+Azf/ABdS5xWjHys6ysDxL4Tt/E17o9xcXMkB0q7W6jCAHeRjg57cVS/tLxD/AM/On/8AgM3/AMXR/aXiH/n50/8A8Bm/+Lpc8R8rJNf8EQa1rSavbXs2n36wG3eaFQxeM/wkGs1fhRoiXWkPGziDTLeWBbdlDLL5gILNnvzmr39peIf+fnT/APwGb/4uj+0vEP8Az86f/wCAzf8AxdHPEOVlf/hW1p/YFhpw1G4Emm3BnsrnaN8BJzgDoR9aTUPhxHqumQ22o61fXEsd+l8ZpCDllzhQvRRz2qz/AGl4h/5+dP8A/AZv/i6P7S8Q/wDPzp//AIDN/wDF0c8Q5WZ/iD4VWHiDUNbup9RuIW1hYVkVEUiPyyCMZ9dtWY/AE9v4ok1u08QXEc7osQV7aOQIg/hUsCQPpU/9peIf+fnT/wDwGb/4uj+0vEP/AD86f/4DN/8AF0c8Q5WRTfD2Iape3Wm6ve6dDqEwmuoLY7fMcHOd3Vc+1dfGgjiVASQoABY5J/GuV/tLxD/z86f/AOAzf/F0f2l4h/5+dP8A/AZv/i6OeIcrOsork/7S8Q/8/On/APgM3/xdH9peIf8An50//wABm/8Ai6OeIcrOsorE0HVL28u7u11DyWkgVHDxKVBDZ7En+7W3Vp3JCiiigDC8Y+FoPGXhqbRrq4kto5XRjJGAWG1ge/0qLxR4Pt/FHh+10m4upYI7aaKYSRgEkoMAc+ua6KigDgLj4TafO17J/aNys1xqg1SOQKp8mUE8AdCOe9Xbv4eibVLnUbPWryxur6JY72S3VVM+Oh/2T7iuyrlrnWdYk1S9hspLSKK2l8sCWFmJ+UHOQw9aTaW47XJNO8FW2neMn8Qx3lxJM1itl5cp3fKCDuLHknjvV/X9EfW7aFIdRutPlhkEiS2zkcjsw6MPY8Vk/wBpeIf+fnT/APwGb/4uj+0vEP8Az86f/wCAzf8AxdTzxHysz5vhTp11pt/Dd39xLd39zHdS3e0Bt6Z2naOO5q1qXw+XUb5746xcw3txbfZbyeONR9pjxjBHQdO1Tf2l4h/5+dP/APAZv/i6P7S8Q/8APzp//gM3/wAXRzxDlZzlz8Or6L4gaZLoryWGk2ekm1F1GylkfcxxtJyc7vStO/8AhNpF74PsfD63E0MNpcef5qgFpGwQc59c1of2l4h/5+dP/wDAZv8A4uj+0vEP/Pzp/wD4DN/8XRzxDlZXtfhjpNl4ztfENtLIkltCIhbhRsbCbAx98AVGPhdpwuPL+2z/ANmfa/tn9nYHl+b656/hVz+0vEP/AD86f/4DN/8AF0f2l4h/5+dP/wDAZv8A4ujniHKyxofgqz0U68BPJcJrdw88yOANm4EFRjtzWNa/Cy3trrR2/ti7e30a5M9rblF2rk52564960f7S8Q/8/On/wDgM3/xdH9peIf+fnT/APwGb/4ujniHKyhP8LLK71OG4u9SubiKG5+0pHKoZw3p5p+fb7ZxWj488N3OvWNhcacqtfaZdpdQKxADFWBIyfYU3+0vEP8Az86f/wCAzf8AxdH9peIf+fnT/wDwGb/4ujniHKyGy8DXWn+INS1nTdeuLOXVZEluYfs8ci5UdAWB9TUM/wAL7GfRfEWnHULgJr12LqVwq5iO7dhfUfWrn9peIf8An50//wABm/8Ai61NA1K7vmu4b/yjLbOo3xKVDAjPQk01JPYVmjDn+Hs8sd5HD4p1a3S4EARYpcCARjGE543d63dG0OTSb/U7mTU7u9F9MJVinfK245+VPQc/pWvRVCCsHxh4Ug8YaPFp91cyW6RXKXAeMAklc8c/Wt6igDnvFfhG38Wf2Z9puZLf+zrxLtPLAO9l7HPasGD4T2VrDD9m1S5jubfUn1GCfYpMbvtyNp4I+Uda7+igDjpvh8p1O8vLHWr2x/tBle8S3wvnMDnOeqn6etX9I8HW2keLtS16K6mllv4YoWjk5ChAADnqScd6rSazrM17draS2ccUM7xKJIWYnaeuQwpP7S8Q/wDPzp//AIDN/wDF1PPFD5WbPiPRI/Efhy90ieV4Y7yPy2kQAlR6jNZeoeBbDUfCmm6LPNIP7MEX2a5AG9HjAAbHTPFRf2l4h/5+dP8A/AZv/i6P7S8Q/wDPzp//AIDN/wDF0ueI+Vmbe/CfT9R0e+tbvUbiS71CeOe5vSq732fdGOmK0U+HemI+sxiR/sWroFmtFUBFYdGGO9L/AGl4h/5+dP8A/AZv/i6P7S8Q/wDPzp//AIDN/wDF0c8Q5WRQfD9k0+eyu9ev7mB7J7KKNiFWJGUrnaOGIB6mmx/DayjTwwov7jHh0OITtA83dj73p07VP/aXiH/n50//AMBm/wDi6P7S8Q/8/On/APgM3/xdHPEOVlA/Cyxl1a2vLvUrm5S1uftMSSqC4bOQDL98gehOK0fG/h661hdLv9MVXvtKu1uIkYgBumRk/Sm/2l4h/wCfnT//AAGb/wCLo/tLxD/z86f/AOAzf/F0c8Q5WZtz8LILq51iRtYu44NZuRPd24RdrYOdufTPes7UPAlz4h8caZcy6W+m6ZpqJG5eZG+0iM5ThSeOB1ro/wC0vEP/AD86f/4DN/8AF0f2l4h/5+dP/wDAZv8A4ujniHKy54x8JQeL9Di02a6ktFinSdZIlBIK5wMH61FceFLnVdLvNO8RazLqlpdR7PLa3ji2HIIYFQDnioP7S8Q/8/On/wDgM3/xdH9peIf+fnT/APwGb/4ujniHKxlv4FuIbO4hk8S6m7SWv2WB1fZ9nXnBAXgsM9TzxUbeAbowyp/wlesAyWCWYYSnKlSp80c/fO3BPua3fD+pXGo29wLwR+dbzGJmjBCt8oOcEnHWtarJIbK3NpYW9s0rzGGJYzLIcs+Bjcfc9amoooAKKKKACiiuQj1vXLsyyW8tlHGJXRVeBmOFYrydw9KTaW40ri674DOuT3ofXL6GzvwBcWmQ6ED+7u+5+GKSH4eWNprZv7G7lgSWxWwnt9oZZYlUKBk8jgDkU/8AtLxD/wA/On/+Azf/ABdH9peIf+fnT/8AwGb/AOLqeeI+VjNG8AjRpbZIdavmsLRmMFkG2Iu45wSOWHsc1Rj+FFhH4TsdBGo3Jis9R+3rLsXczc/KfbmtH+0vEP8Az86f/wCAzf8AxdH9peIf+fnT/wDwGb/4ujniHKy7p3hO307xpqviOO5kebUo443hYDagQAAg9e1c/e+AprvxD4ji3GPTPENsonmQjfHIpPY+oP6Vp/2l4h/5+dP/APAZv/i6P7S8Q/8APzp//gM3/wAXRzxDlZt2GjQWXhq30Vmaa3htRaktwXULt5+orm9M+Gllp95ZPJf3FzbaazPY2sigLbljkkEcn8as/wBpeIf+fnT/APwGb/4uj+0vEP8Az86f/wCAzf8AxdHPEOVkenfDrT7DwBdeE3upp7S5Z2aVgAw3MG7ehFNk+H29dLmTWbmPUdLi8i3vViQMIsY2leh4qb+0vEP/AD86f/4DN/8AF0f2l4h/5+dP/wDAZv8A4ujniHKxh+HsD3+g3k+qXs82jTSTK0z7zMz4zknoOOgqSbw5c3/xNh127ULa6famK1ORlmb7xx26mk/tLxD/AM/On/8AgM3/AMXR/aXiH/n50/8A8Bm/+Lo54hys6yiuT/tLxD/z86f/AOAzf/F1s6DqE+o6e8l0EEsczxMUGASp6gU1JPYTTRp0UUVQgooooAKKKKACiiigAooooAKKKCQoyTgepoAKKKKACs+2/wCP++/66L/6AK0Kz7b/AI/77/rov/oAoAy7DxBOsqJfBWRjjeoxipPE+q3VjLp0NpcxWi3UjB7mVNyoFXIH49Kw2H7vHtXTR2dxc6fDHdR2syBVIWWPdj0696APP9F8Talp/h6xitriJEtrSOVYmjy14XkcFVPbGM8etdRpOv3974jaKe9t0i+1TW4sTGfMCpuw+QO+0HnAwa2v7L/1X+iWH7n/AFX7gfu/930/CnrYypdNcpDZrOww0oiwxH160AaNFVMX/wDeg/75P+NGL/8AvQf98n/GgC3RVTF//eg/75P+NGL/APvQf98n/GgC3RVTF/8A3oP++T/jRi//AL0H/fJ/xoAt0VUxf/3oP++T/jRi/wD70H/fJ/xoAt0VUxf/AN6D/vk/40Yv/wC9B/3yf8aALdFVMX/96D/vk/40Yv8A+9B/3yf8aAMnxF4jk06dLHTolmvJBuO4/LGPU1y2l3N5P4nvG1CRJJWiXJRcDvU8wlXxZqf2vBmOwgjpt2jpUGm/8jVd/wDXFf5msqsU43fQ0pyadl1N6eaO2t5J5m2RxqXdj2AGSaWKRJoUliO5HUMp9Qelcn4zt7m4kKiG8mt2s5FRbUMf3xBxu29sZ61StLbVE1W2DRXizK8OxsN5SwBF3Ke27O73rLl0Kud3RXKeFYbqO8zdx36z+SRdNMT5TyZHK5/H7vFdXUtWAByRVl4LeNtslyiH0YgVXX7w+tR6rDvvCfn6DokZ/wDQlJroowjK9zGtKUbWLWy0/wCfyL/vsf402RYFXMU6yH0Ug1kSxiKMuxkwP+mcP/xNGlFmErMWOWH3lQY/75Arp9jDlbsc3t5qSVzSooorzT0AqG1u4L22W4tZBJExIDAdcEg/qDWP4pjuJIrPy0uJLYT5uVts7yu04xjn72K5SKx1aHSrWNLe/RxA32RUDDy5jK5zJ7YK/eqkroVz0muenu9QtvFjtpssaOsAyJFyGGelU7OC7HihnuY7/wA77Qx8xSRB5ODtBz8vp05q5L/yN0v/AFwFOMU5K4+ZqLsdf4e8Qf2uskNzH5F5DjzI85BHqPatuuA0oTHxpF9jKhxbt5m4cYyMZ/Wu0xf/AN6D/vk/411GBboqpi//AL0H/fJ/xoxf/wB6D/vk/wCNAFuua1/xLNaXo07So1kutu53f7sQ9/etrF//AHoP++T/AI1wsQkXXtVFzj7R5+Wx6Y4/rQA7Rrm7udV1Fr+RZZiybmVcDhcCtS7u4LG0luruQRwxKWdyCdo9eKydF/5DGo/7y/yrE8cWl5dSXSLb3k8T2W23W3DECXJ3bgPbb1rnkryNVsduDkAjoaK4c2+rf245hjvBcedJmQ7vJMO07AP4c7tvvWl4UhljkkMkV/G3lKJzdE7Wl/iK7ucfTiosO501Ko3MB6nFJTo/9an+8KS3B7E7w28bbXukVvRmANN2Wn/P5F/32P8AGqWow775z8/4JGf5qTVKZBDGWbzPYeXDz/47Xoexh2OH2s11NeRYVA8qZZD/ALJBxUdZ+k7jC5csSWz8wUf+ggCtCubEQUJJI3w9Rzi2+4UE4BJ6Ciue8UxTySWWI7qS1Bfzltd27dgbPu84zmsFqdBt2l3BfWkV1aSCSGVQyOARuHrzU1ecWNlqsFjp0TW98k6W8C220MEiI/1gcdOnrW3okN0uvs13Hfi48yXzZGJ8hkydmM8dMdO/WqcRXLv9oanaeJ7s6ZJGv7qPeki5DgFsc9uprsfD+urrVu/mR+RcwnbLET0PqPauMH/I03n/AFyT+taGgC4Pi6c2ZUYtv3u4cdRj8a6I7Iye53VFVMX/APeg/wC+T/jRi/8A70H/AHyf8aoRboqpi/8A70H/AHyf8aCNQwcNBn/dP+NAGFrvia5i1BtO0dEaZBmWZ/ux+3uaxPD9xc3E2oNeOskxuMuyjAJ2iorAOt7qIn/1/wBqfzPzOKf4e/4+NR/67/8AsorOp8JUdzWuruCytmuLqQRxKQCxHTJAH6kVNXEeK7S7udRuAbe8nVhB9l8gMUGHBfdjj06+lRpb619tuvs6Xa3hW48ySTd5TAhvJ254yPl6Vjy6Glzu6K57wtE0YuMRX0UZCcXZPL87iobnHT29K6Gpegx0a75FXpk4qZobZGKvdRqw6gsBj9ajg/4+E/3qp30G69kPz9eyRH+a5rpowjJO5z1Zyi9C/stP+fyL/vsf40yRYlx5Uokz3Ugisa4UQRFj5mew8uH/AOJqxpefsmWLElj94KP/AEEAVtKjDkcrGCrS51G5dooorzzvGyyJDC8sp2oilmPoB1pIJo7m3jnhbfHIodGHcEZBrmPGEFzPNEBFdzWpt5V2WobPmnG3O3tjd14rLtLXVY7yxUw3iyoLUREBhEkQRBIrds53deavl0Fc76sCHU9Vs9e1A6c8RQMheKQffwvr2ql4bhuo9TzdR363Gx/tLyk+SzZ+Xbn2z938at23/Ie1H6r/ACq6atImWx2+ha1HrVkZVUxTRtsliPVWrTrhvDC3B8S6ibMoE8tBJuHG7n9elddi/wD70H/fJ/xrczLdFVMX/wDeg/75P+NGL/8AvQf98n/GgC3XJax4oum1CSw0RE3Q8S3EnKqfQDvXQsNQ2na0GccfKf8AGuA0gEQTh/8AWid/M+uf8MUAW/Ds889vdPdOHlNy5ZgMZJPNaN1eQWUaPdSCNXkWNSQeWY4A/Emszw3/AMe93/18N/Ouf12zvZ/EeXt7yXF9bPA0YYxLEGQtnHGchjzXM1eTNVsdzRXn8NvreLj7PHerc+RILlpN215Nw27M8dN3Suk8MRiO3n8qK+igLjy1vCd3TnG7nGfX8Klqw7m5T4k8yVUzjNMqW2/4+U+tEVeSQpaJjzFaqxDXcYI6gsP8aTZaf8/kX/fY/wAazbmDN1If3n3j0SL+q5qpcjyIs/vNx6Dy4f8A4mvQ9jB9Didaa6mzIIlI8qQSe45FMqnpgIs03FieclgoPX/ZAFXK5K8FCdkdFCbnC7Corm5is7WW5uXEcMKF3cj7qgZJqWuM8cWt3deeiQXk8L2LpCtsGI845+8B2xjrWKV2bM7JHWSNXQ5VhkH1FLXDRW+qLrCbYrwXAlGHO7yRBs+7/d3bvxrQ8Jw3EczG5iv0m8kC5a4J8tpcjJXP49OKdguWNP1bVrLUtRawMUkQuNzQOMbjtHQ9uld1ourxazp63MSlGB2yRt1Ru4rg9N/5CGo/9d//AGUVseEluv7S1U2hQQ70zuBxu5zXUtjJ7naUVUxf/wB6D/vk/wCNGL/+9B/3yf8AGmIt0VUxf/3oP++T/jTJ11HyH2NBu2nHyn/GgDnNV8U3s97Na6GsapCSr3MnIJ9AO9UPDUskukkzNufzZCxA6kuSapaLxo4B++Mh89d3ernhj/kEn/rs/wD6EazqfCVHc0ri7gtTELiQIZpBFHkfeY5wP0NTVwmpWd9L4jVjb3rzLqAeOQBjCsWxse2c5/Ooba21gWNwIY75X8hBdeZuy8vmLvMee23djbWPKaXPQaKx/DUZj0+QLHdxxGUmJbsneFwPXnGc9ea2KljJIY/NlCE4zTzHag4N3GD/ALw/xos/+Plfoay5rfMzn951/uRf/E11UacZRu0c1WcoysjU2Wn/AD+Rf99j/Go5BGrARSeYPUdKxbv9xHgGTe3T93Dx/wCO1esARZx5JJx3AHf24rSpRiqbkkZQrSdRRuWaKKK4DuIbu7gsbV7m6kEcMYyzEdKmrh/GNpeXV5dL9nvJ42gjFqLcMVDbvn3AcdNvX0NH2fWP7ZuDCl2LkvcbpG3eSYyreUFz8uc7OnNXy6Cudwa5/Sta1ewku3tfKmtluXZoGGCeecGo/C0UkbTkxX8YMaiT7WTgyc7iobnHT2p2kdbz/r4f+daU1ZsiWx32k6nDq+nR3dv0bhlPVSOoNXa4zwYt1nUjamMQG4+XcD12jOPxrqMX/wDeg/75P+NbEFuiqmL/APvQf98n/GjF/wD3oP8Avk/40AW6KqYv/wC9B/3yf8aMX/8Aeg/75P8AjQBboqpi/wD70H/fJ/xpFGoEHcYBz6H/ABoAuUVUxf8A96D/AL5P+NGL/wDvQf8AfJ/xoAt1558QdWuyNS00XUVtbRW0ThHTLXBZ+Qp7Y2j867jF/wD3oP8Avk/41XuNOe7ZWurexnZeAZIdxH50Acq3i7VI9Vnw0cqiW6iFksfzxiJXKuT1IbYP++q2fCuq3eoPKLvUba9zEkoEMZVoS2coeMHGPXPtWkLCVbhrhYLITOMNIIvmYehPWlt7Ka0DC0hs4Axy3lxbcn1OKAK2ra21rN9ntFDS4+Zj0Wm6DPLcpdSzkNI0oyQMfwisS5Eg1C587/WeYd2K2PDf/Hvc/wDXX+goAdYeHykiy3zBivIjXpn61u0UUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGHr/h3+1ZEurSb7PeRDCvjIYehrktMtLy18UXiagIxKsS58s5BHNek1xd1/yOl7/1xj/m1cuJm48qXV2/Bs6sPTU1Nvor/il+pboooqCAooooAVfvD61Hqqt9sJCEjA58gP8AruFSL94fWszxHdILwxBcnA3EQO/6gEV24VXucmJdkmZl3LNNJhYm2L0/0Rf/AI5WhogZYZN6lfmHWIJ/JjWREiTTLGiAMxwN1rIB+ZXit7TrRrRGV/L+Y5+QV3SVoP0POi7zXqi9RRRXiHuBRRRQAVz81jqF54tddNEW8wAsZSQFGa6CotF/5HKX/r1H8zUSm4Sjbq7GtOCnGd+iua+haCmjxyPJJ591NjzJSMZ9h7Vr0UV6BxhRRRQAVz+veGjqFwL7T5hBeKu0kjKyD0NdBRQB57o9neWeraimoCNZgybhG2RyuRWzUdx/yM+qfWL/ANAFSVyz+I1jsFFFFQUFOj/1qf7wptOj/wBan+8Ka3E9ivqQYXkjbCV9fs4b9d4/lXPXUk08nETbR0H2Rf8A45Wh4guozfyRBc4+8fs7tn8QMVlwRJcTLFGihm6braRR+ZXFe1GPU8apK+iNnRgwtjuXad3Tywn6AmtKqWnWrWsRR9mSc/IOKu1wYv8AiL0O7B/w36hRRRXGdgUUUUAYQ0vUr7xPdjTvJUCKPe8hPyglsYHfoa7LQ9Ej0a2ZQ5mnlO6WVhgsf8KzvDv/ACMOp/8AXGD+b10tdcfhRi9woooqhBRRRQBzeueGZLu8N/pcyw3RGHRx8sn19KwdAtbm1n1GO9CCZbjDhDkZ2qf616FXHR/8hrV/+vv/ANppWdT4So7liiiiuY1CiiigCSD/AI+E/wB6qWoblupW2HaD1+zhv13irsH/AB8J/vVga5dRvqEsYXIDcn7NI2T9QuDXdhldM4sTLlaKVxJNPISYm29h9kX/AOOVs6QCLIbl2nJ48sJ+gJ/nWJbwLdTCKJFDEZ+e2kUfmVxXQWFu1tbiN9mck/IOK66q/ds4qbvVj6lqiiivGPaCiiigArCt9I1S+17UPsDQxx7kDyOTlcrngd63aseGv+Qjq3/XSL/0Ctae5EtjR0bR4dGs/JiJd3O6SRurt61oUUV0GYUUUUAFcvrPhaaW9e/0aVIp5P8AWRSfckPrnsa6iigDz/w9bz20F1HdBRKty6uFOQCDzWvVa2/4+9R/6/Zf51Zrll8TNlsFFFFQMKltv+PlPrUVS23/AB8J9auHxImXwszr3dHNK5jOAT/y7A5/HeKwZ2mmkLGJsdh9kXj/AMiVb1a6jlvZECEqrH/l1kOT9QtVba3F3L5cSIGxn57d0H5kV7EY2PGnK+huaWCLFAw2nnjZt/TJ/nV2q1jCbe3WNtuR/d6VZrzsV/FPRwv8IKKKK5TqCiiigDB03RdUv9S1D7JJDDb/AGja8rElgdqngfQ+td1pOlw6RYLbW+W5yzt1Y9yazPCnTVP+v0/+i0roK7I7GL3CiiimIKKKKAOT1jwrc/a5bzRJURpjmSCT7rH1B7VmeG4pIdJKzABxNIGAPcOQf1rv64nSv+PWX/r5m/8ARjVlU2LjuXaKKK5zQKKKKAJ7P/j5X6Gsm6LRNI5jOAeP9FU5/wDH61rQ4uAT6GuV1G5jmunAQ7VJx/oshz/47XoYZXicGJlyyK8pnmkLNE3/AICLx/5ErotPBFlHkYO3pt2/pk/zrAtbX7Y7LCkYKjJ8yB0/mK6K0jMNukbbcqMfL0rorr90zmoP96iaiiivHPYCiiigANYWkaDqmoS3YhligtDcurSZJfg84H/163au+FP+Qddf9fsv862pbkTNLTdOg0uxjtbYYRB1PVj3Jq1RRW5mFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAZOq6KL2Tz4H2TYwcjhqi0K3ktVuoZ8b1lGcHI+6K0ru/hs8K+55G+7GgyTVPTJjcT3kjRtGTKPlbqPlFAGrRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFcXdf8AI6Xv/XGP+bV2lcPqdxFaeK9RnuG2RxwIzMewy1cOL3p+v6M7cLpGp/h/VF+q97fQafbGe6fag/M1mWnirT7m0lnLGMRjIVurDtiuRvLy/wDFWsra22QP/HYV9T71MppLQ86riYxiuTVvY9FguIrqBZoHDxuMqwPWpKpaTpkOkadHaW+dqcknue5q7VnTG9lfcKDycnrRRVKUo7MHGMt0JgUYFLRVe0m+rF7OC6IKKKKzLCiiigAqLRf+Ryl/69R/M1LUWi/8jlL/ANeo/maxqfFD1R00Phqf4X+h11FFFemeeFFFFABRRRQByVx/yM+qfWL/ANAFSVU1S8hsNb1i5un2RRiIk/8AABWfD4qsJdPkuSWUpx5Z+83piuSo0pFe0hHSTNG+1C206ES3cmxScD1NTxSpNEskTBkYZVh3FecSSX/i7W/Jj4UffYfdhX0+teg6fYw6bYRWlsMRxjA9/U1nGXM/Iyo1pVZNpe6WaOlFFWdQYz1pMCloq/aT7sz9nDshMClooqXJy1Zaio6JBRRRUjCiiigCTw7/AMjDqf8A1xg/m9dLXNeHf+Rh1P8A64wfzeulrrj8KMXuFFFFUIKKKKACuOj/AOQ1q/8A19/+00rsa4S91CDS7zWrq7bbGt3+Z8tOKzqfCNNLVmlVS/1O101Fa7kCBjgVnjxTYnS2u8kODtEJ+8T/AIVyEaXvjDWmQsRCv+tkH3UX+6K45T7GNXEpWVPVs9JR1kQMhDKwyCO9LUNpaxWVnFbQDbHEu1RU1WdavbUAcHIpMCloqlKUdmJxjLdCYFLiiim6k2rNiVOCd0goooqCwooooAKseGv+Qjq3/XSL/wBAqvVjw1/yEdW/66Rf+gVrT+IiWx0NFFFdBmFFFFABRRRQBxtt/wAfeo/9fsv86s1lXWp2+krqVxdNhft0qqO7HPSoJfFVmuk/ak+aVvlWE9d3v7e9cc2lJlOrCOkmaF7qlpp7RrdShDIcL/jVsEEAg5B6GvN7K0vPF+sO8zN9mU/vpex/2Vr0WCFLa3jhhG2ONQqj0AqIyctTOhVlVvK2nQkoBx0ooqzpEwKMCloq/aT7sz9nDsgxiiiiobbd2WkkrIKKKKQwooooAs+FOmqf9fp/9FpXQVz/AIU6ap/1+n/0WldBXZHZGL3CiiimIKKKKACuJ0r/AI9Zf+vmb/0Y1dtXnR1e30fS3muTy91MqKOrHzGrKrpEako3bNyqd3qtnZXEUNzKEeU4UH+tZ154ptIdKW5gO+WQYWI9Qff0rldK0658Wao9xduxtEbMkh/5aH+6PauRz6IyqYjVQp6tno4ORkUU2NFjjVEGFUAAe1OqzrDOOlJgUtFWpyjsyXCMt0JgUuMUUUOc2rNiUIJ3SCiiioLCiiigAq74U/5B11/1+y/zqlV3wp/yDrr/AK/Zf51tS3ImblFFFbmYUUUUAFFFFABRRRQAUUUUAFcD491vUAupadaSw28NvbRSSM4PmSb3xhCCMY289etd9VC/0PS9UlSTUtPt7p0GFaWMMQPx+goA5VvGmoRalMzxW72nm3MMcCqfNUwq7bmOcYbZxx3FbHhnV9Q1CSRNSmspSYknjFtkMitn5WBJ6Y6/pWoujaamoSXyWMC3Uq7XmCDcw9CadYaTp+liQabZw2olbc/lIF3H1NAGXBqVrFqN1LdbjKH2L8udqirGlTJcTXksedrSjGR/siqWs6VMl091bIXSTl1HUH1qXw3kW9zkEHzehGOwoA36KKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArib+JJ/F2oRTIHR4EDKwyCPmrtqxdT8MWupX32wzTQTFQrGJsbgOlcmKpzmouCvZ3/AAZ14acI80Zuyat+KPLtS8JX9pqQt9JXfa3DfKzc+R3OT6V2GiaJbaJYiG3G6RuZZT95zWv/AMIZD/0Ebz/v5R/whkP/AEEbz/v5XMoVk7qn+KM6eEwlOTlGf4MioqX/AIQyH/oI3n/fyj/hDIf+gjef9/Kv9/8Ayfijo5KH/Pz8GRUVL/whkP8A0Ebz/v5R/wAIZD/0Ebz/AL+Ufv8A+T8UHJQ/5+fgyKipf+EMh/6CN5/38o/4QyH/AKCN5/38o/f/AMn4oOSh/wA/PwZFRUv/AAhkP/QRvP8Av5R/whkP/QRvP+/lH7/+T8UHJQ/5+fgyKipf+EMh/wCgjef9/KP+EMh/6CN5/wB/KP3/APJ+KDkof8/PwZFUWi/8jlL/ANeo/matf8IZD/0Ebz/v5WhpOgWukPJJC0ks0gw0krZJHpSVOtOceaNknfcpSo04S5ZXbVtjUooor0jzgooooAKKKKAOO1C3iu9e1eC4RZI5BEGVh1+QVwd34Q1KDUxZWPzWkxys7DPlDuDXq994dtb68e6M1zBI4AfyZNobAwM8VB/wilv/AM/+of8Af/8A+tXPUo87uZ1aVOqlzdDF0jSLbRrFba0X3dz95z6mr1XP+EUt/wDn/wBQ/wC//wD9aj/hFLf/AJ/9Q/7/AP8A9aj2TRvFxirJFOirn/CKW/8Az/6h/wB//wD61H/CKW//AD/6h/3/AP8A61P2THzIp0Vc/wCEUt/+f/UP+/8A/wDWo/4RS3/5/wDUP+//AP8AWo9kw5kU6Kuf8Ipb/wDP/qH/AH//APrUf8Ipb/8AP/qH/f8A/wDrUeyYcyKdFXP+EUt/+f8A1D/v/wD/AFqP+EUt/wDn/wBQ/wC//wD9aj2TDmRToq5/wilv/wA/+of9/wD/AOtR/wAIpb/8/wDqH/f/AP8ArUeyYcyK/h3/AJGHU/8ArjB/N66WqWm6Vb6WjiDe7SHLySNlm+pq7WyVlYh6sKKKKYgooooAK4e5s4L++1u2u4xJE91gqf8ArmldxWPdeGrS6vJbkT3UDynLiGXaGOMZ6e1TOPMrDVup5TJ4Q1NdU/s+E5s2O4XRHKp6fWu603TbbSbFLWzjCIo5Pdj6n3rX/wCEUt/+f/UP+/8A/wDWo/4RS3/5/wDUP+//AP8AWrnjQ5djKlRp0m3FblOirn/CKW//AD/6h/3/AP8A61H/AAilv/z/AOof9/8A/wCtV+yZ0cyKdFXP+EUt/wDn/wBQ/wC//wD9aj/hFLf/AJ/9Q/7/AP8A9aj2TDmRToq5/wAIpb/8/wDqH/f/AP8ArUf8Ipb/APP/AKh/3/8A/rUeyYcyKdFXP+EUt/8An/1D/v8A/wD1qP8AhFLf/n/1D/v/AP8A1qPZMOZFOirn/CKW/wDz/wCof9//AP61H/CKW/8Az/6h/wB//wD61HsmHMinVjwz/wAhHVv+ukX/AKBUn/CKW/8Az/6h/wB//wD61aWnaZb6ZC0dvuO45d3OWY+5qoQcXcTldFuiiitSAooooAKKKKAODn0+31NdVtbtA6Pey/VTnqPeuNTwfqcmpnT5G22ancboDkp6fWvVbjwxaT3Us63F3AZWLssMu1ST1OMVH/wilv8A8/8AqH/f/wD+tXNOhzu7M6lGnVs5LYy7Gxt9Os0trSMRxIMADv7mrFXP+EUt/wDn/wBQ/wC//wD9aj/hFLf/AJ/9Q/7/AP8A9an7Jm6kkrIp0Vc/4RS3/wCf/UP+/wD/APWo/wCEUt/+f/UP+/8A/wDWp+yY+ZFOirn/AAilv/z/AOof9/8A/wCtR/wilv8A8/8AqH/f/wD+tR7JhzIp0Vc/4RS3/wCf/UP+/wD/APWo/wCEUt/+f/UP+/8A/wDWo9kw5kU6Kuf8Ipb/APP/AKh/3/8A/rUf8Ipb/wDP/qH/AH//APrUeyYcyKdFXP8AhFLf/n/1D/v/AP8A1qP+EUt/+f8A1D/v/wD/AFqPZMOZDPCnTVP+v0/+i0roKrWGnwabbeTbA7c7mZjksfUmrNbLRGYUUUUwCiiigArzw6Vbaxo09tdrkG4n2sOqHzG5Feh1iP4Vs2ld47m8hDsWKRS4UEnJ4x61E48ysGjVmeW2vhDUrjUzZXvyWcJy06jBlHYCu/tbWGytkt7WNYokGFVe1aP/AAilv/z/AOof9/8A/wCtR/wilv8A8/8AqH/f/wD+tWEaHLsRRpU6N+Up0Vc/4RS3/wCf/UP+/wD/APWo/wCEUt/+f/UP+/8A/wDWq/ZM35kU6Kuf8Ipb/wDP/qH/AH//APrUf8Ipb/8AP/qH/f8A/wDrUeyYcyKdFXP+EUt/+f8A1D/v/wD/AFqP+EUt/wDn/wBQ/wC//wD9aj2TDmRToq5/wilv/wA/+of9/wD/AOtR/wAIpb/8/wDqH/f/AP8ArUeyYcyKdFXP+EUt/wDn/wBQ/wC//wD9aj/hFLf/AJ/9Q/7/AP8A9aj2TDmRTq74U/5B11/1+y/zpP8AhFLb/n/1D/v/AP8A1q1bKyh0+1W3tlwi889SfU1cIOLJlK5YooorQkKKKKACiiigAooooAKKKKACiiigAooooAKz7b/j/vv+uq/+gCtCs+2/4/77/rov/oAoA0KKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigArPtv+P++/66L/6AK0Kz7b/j/vv+ui/+gCgDQooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACs+2/4/77/rov/oArQrPtv+P++/66L/6AKANCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKz7b/j/vv+ui/+gCiigD//2Q==)

The operators are

* **Transform cases**

This operator is used to transform all the characters to lowercase.

* **Tokenize**

This operator will split the sentences in to tokens. There are several options how to specify the splitting points. We used non-letters to split in to tokens. For example, the operator will create token when there is a full stop, coma, space etc.

* **Filter Stopwords (English)**

This operator will remove English stopwords. For example, am, is, are etc.

* **Filter Tokens (by length)**

This operator filters the tokens according to their length. We have given the range 4 to 25. So, the operator will filter every words except this range.

* **Stem (Porter)**

This will change all the words to its base form. For instance, gone, going will be changed to go.

So, after getting the result from these operators we give this to set role operator.

**Set Role Operator**

It is used to change the role of one or more attributes and here we make sentiment as the target attribute so that the system can identify the sentiment attribute as the target attribute. Output from this operator is given to another operator called Select Attribute.

**Select Attribute.**

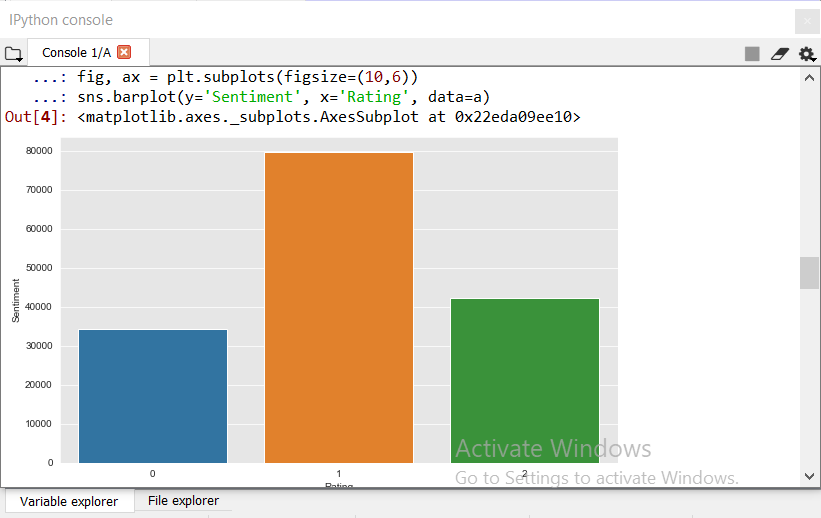
This is used to filter out the attributes which have any missing values. So, we have given attribute filter type as no\_missing\_values. The output of this operator is given to the Validation Operator. We used split validation here.

**Modeling**

In the modeling phase, various modeling techniques were applied to the dataset. The performance and quality of each model were determined after experimenting with different models. Finally, the best and the accurate technique was chosen to create a final model for the data mining problem. 8 different models were used in this process in rapid miner and python. The data was spitted and different algorithms were applied. i.e., by Split Validation operator. Here the entire data set was spitted into training and test data sets in a 70 30 ratio. Now in the validation operator, we used 8 different models namly SVM, Decision Tree, GLM, Logistic Regression, Fast Large Margin, Deep Learning, Random Forest and Naïve Bayes. The training data is given to the operators and they create models and ‘Apply Model’ operator is used to apply the model on the testing dataset. Then we used ‘Performance’ operator to find out the performance by each model.

We used Rapid Miner and Python keras package to compare the accuracy using different machine learning algorithms. Keras gave a better accuracy than rapid miner

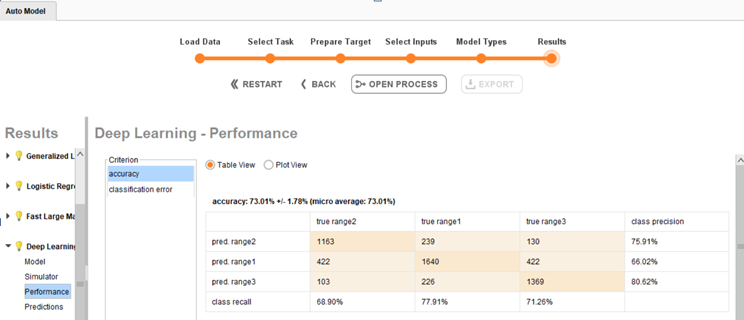
The below figure shows the count of records under three classifications in the test dataset



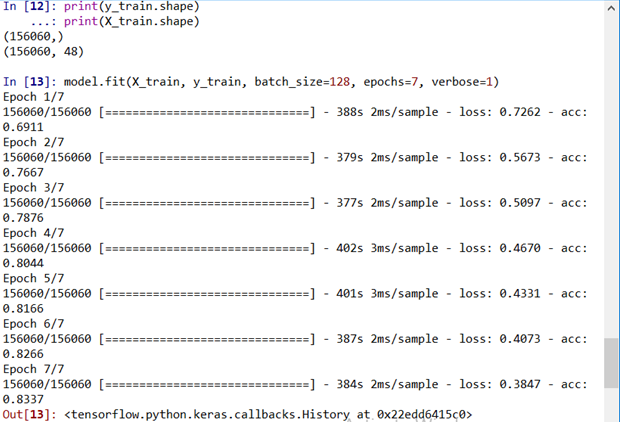
**Dataset Bar**

1. **Deep Learning**

Rapid Miner

****

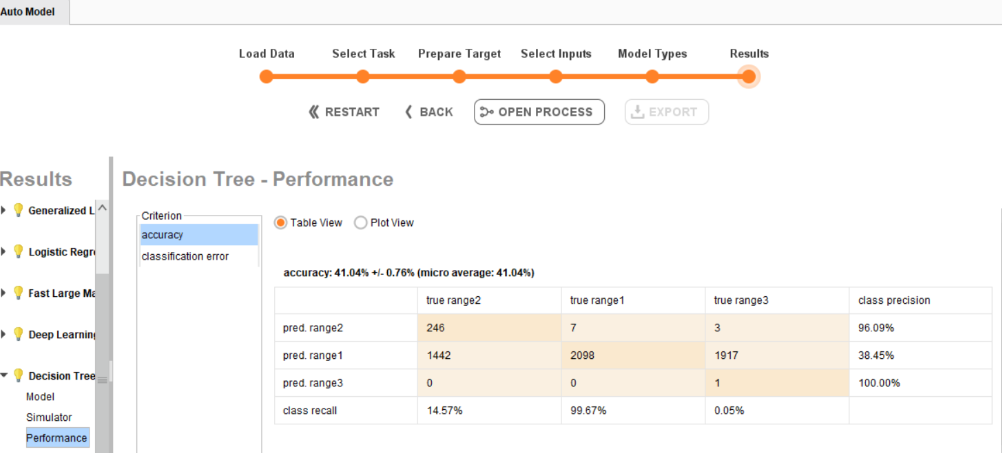
Python

****

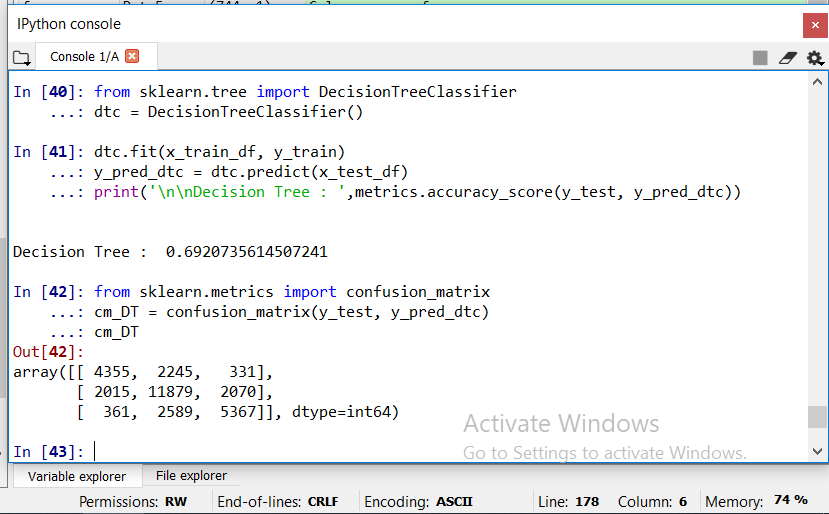
In rapid miner deep learning gave an accuracy of 73 where as keras gave 83

1. **Decision Tree**

Rapid Miner

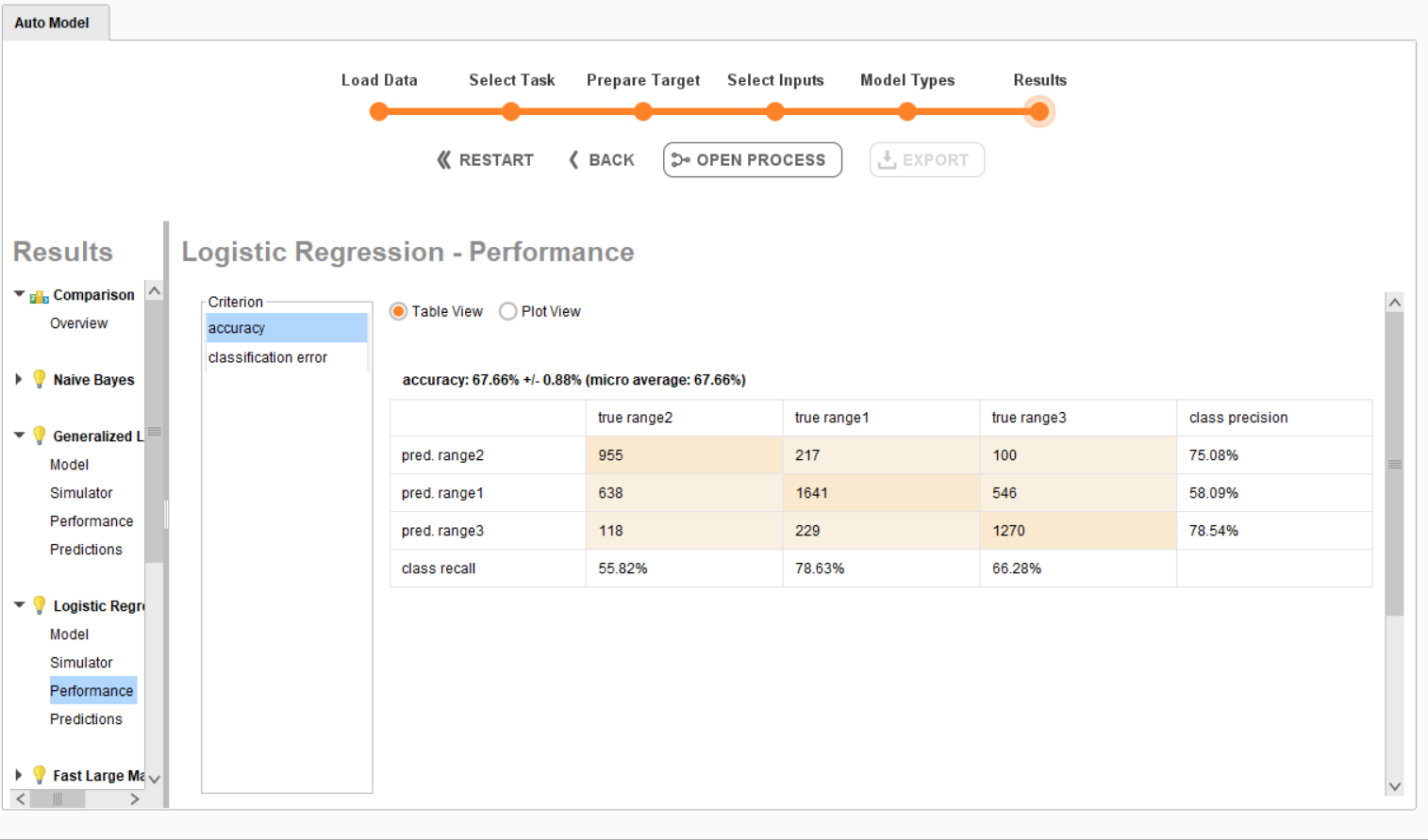


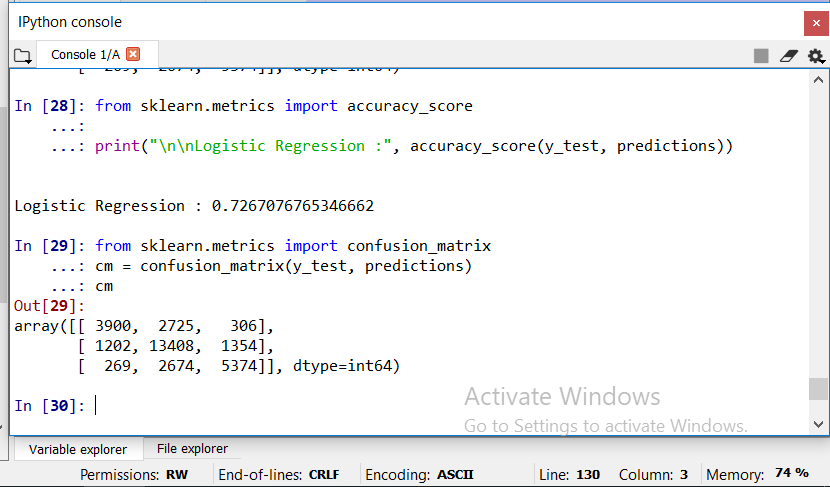
Python



In rapid miner Decision Tree gave an accuracy of 43 where as Python implementation gave 69. The confusion matrix is used for cross validation

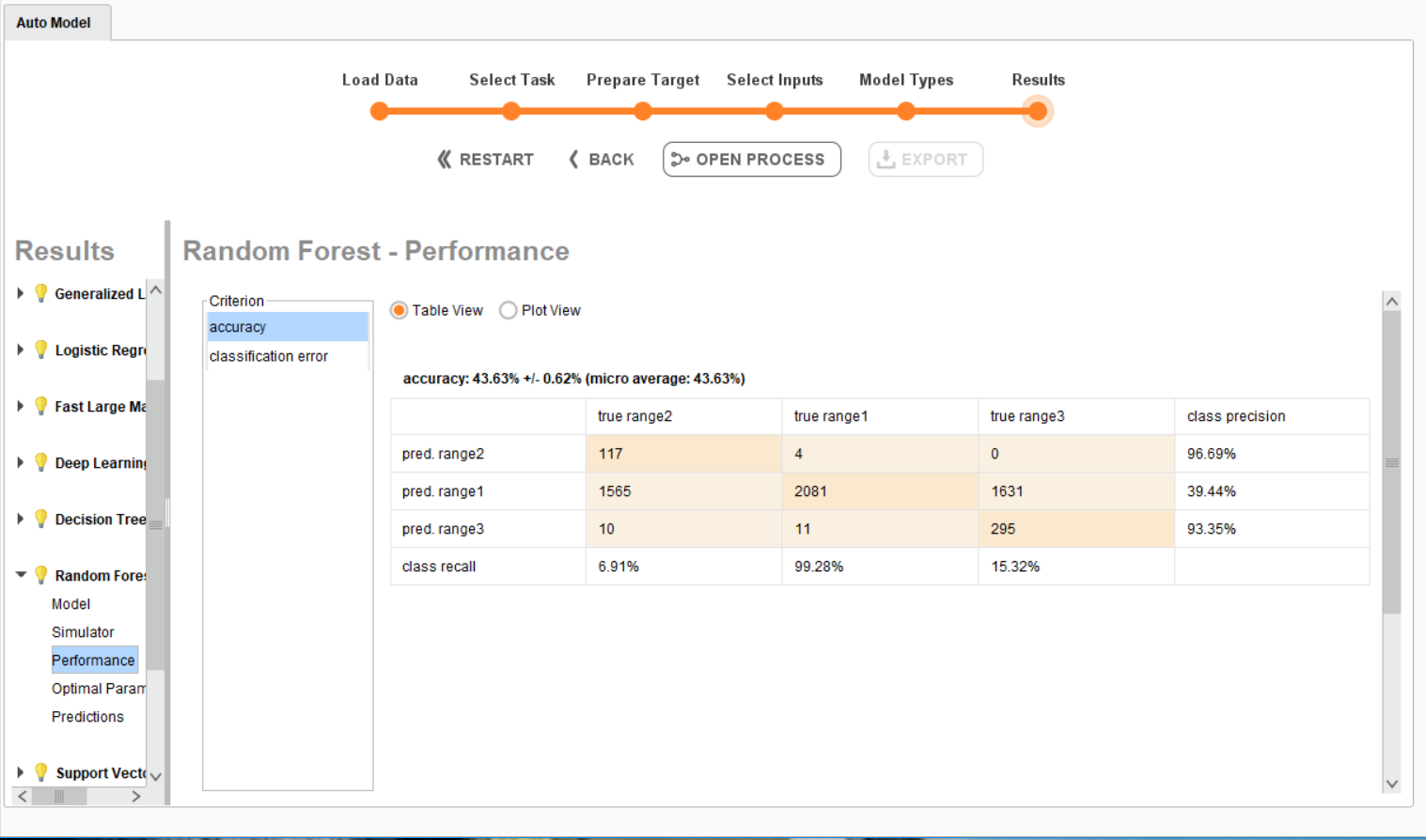
**3.Logistic Regression**

Rapid Miner****Python

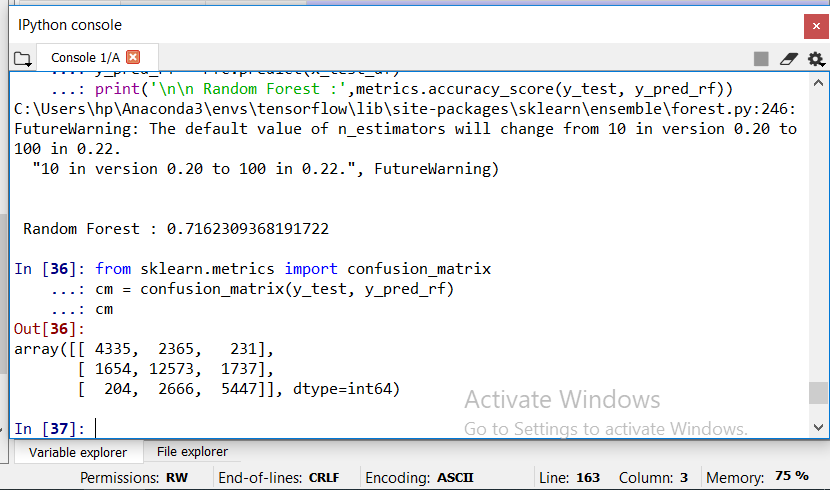
****In rapid miner Logistic regression gave an accuracy of 67.6 where as Python implementation gave 72. The confusion matrix is used for cross validation

**4.Random Forest**

Rapid Miner

****

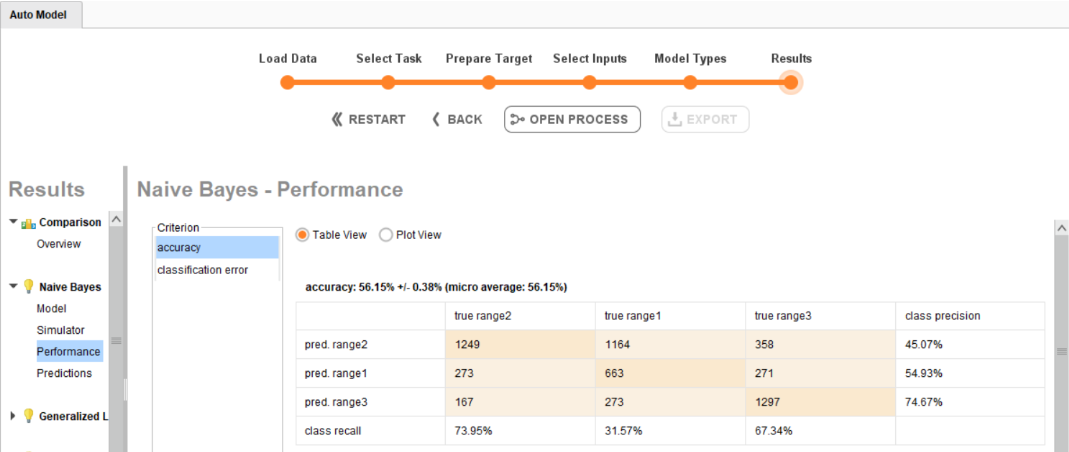
Python



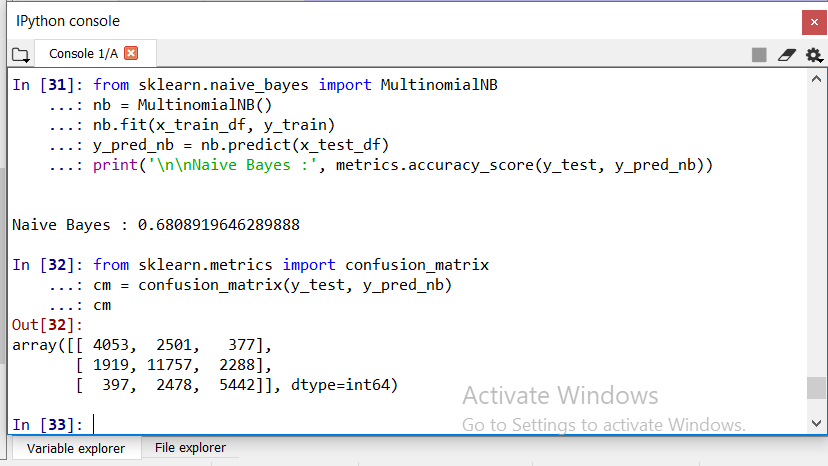
In rapid miner Random Forest gave an accuracy of 43.6 where as Python implementation gave 71. The confusion matrix is used for cross validation

**5.Naive Bayes**

Rapid Miner

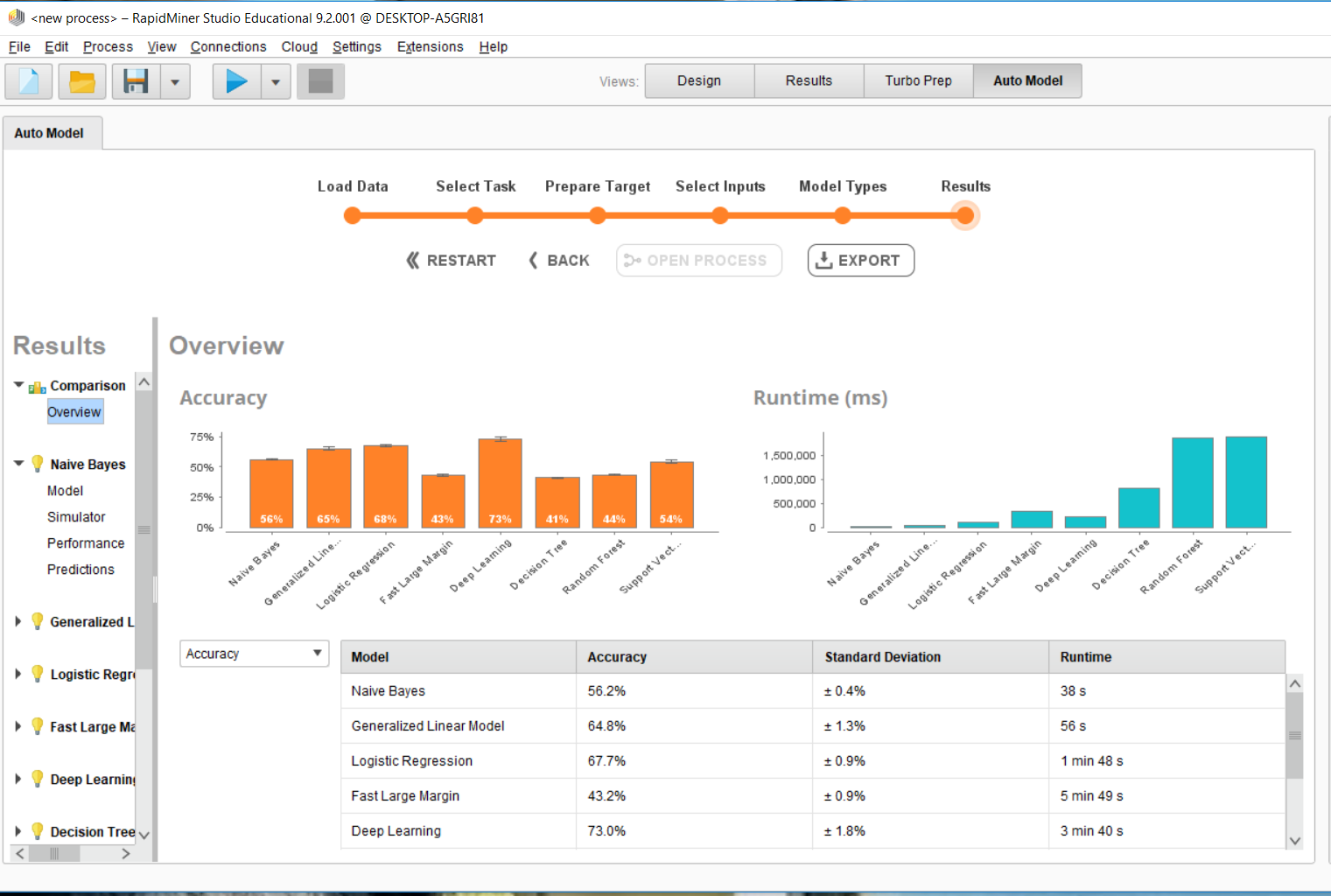


Python



In rapid miner Naive Bayes gave an accuracy of 43.6 where as Python implementation gave 71. The confusion matrix is used for cross validation

**Performance of Different Algorithms in Rapid Miner**



By analyzing different models and accuracy, we concluded that deep learning is the best model for this problem. Deep learning gave the greatest accuracy in both Rapid miner and Keras(LSTM) using Python

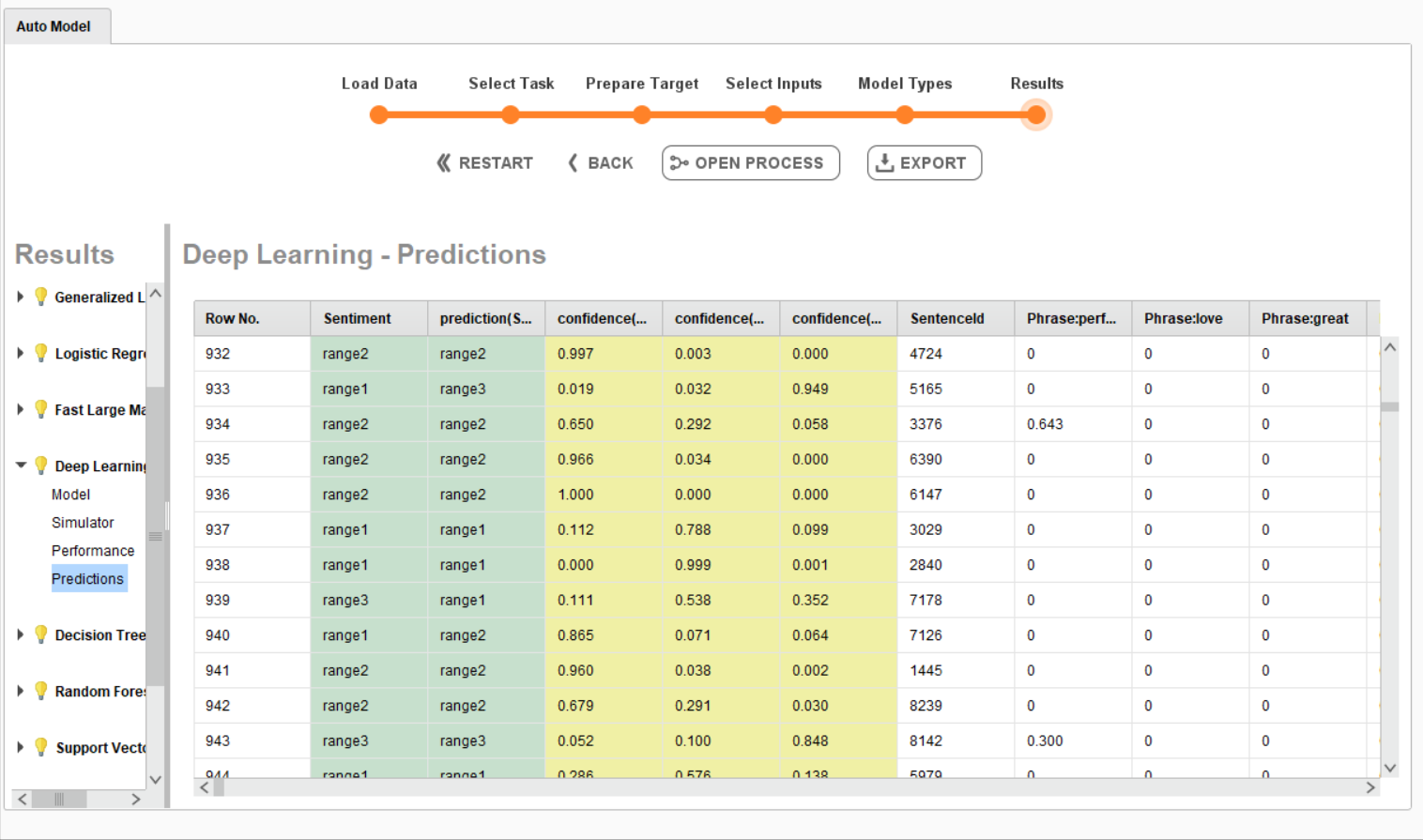
**Evaluation**

This step of Evaluation involves determining if some crucial business issue has not been adequately considered. At the end of this step, we had to determine exactly how to make use of this data mining results.

From the results of performance operators, deep learning had the best accuracy rate. So, this model is selected for further steps. With an accuracy of 83%, the model has correctly labelled the phrases from the test dataset as positive, negative or neutral . This is validated on a separate test dataset of 65000 phrases to verify the accuracy of prediction.

The steps for evaluation are

* Evaluate the Results.
* Process review.
* Determination of next steps

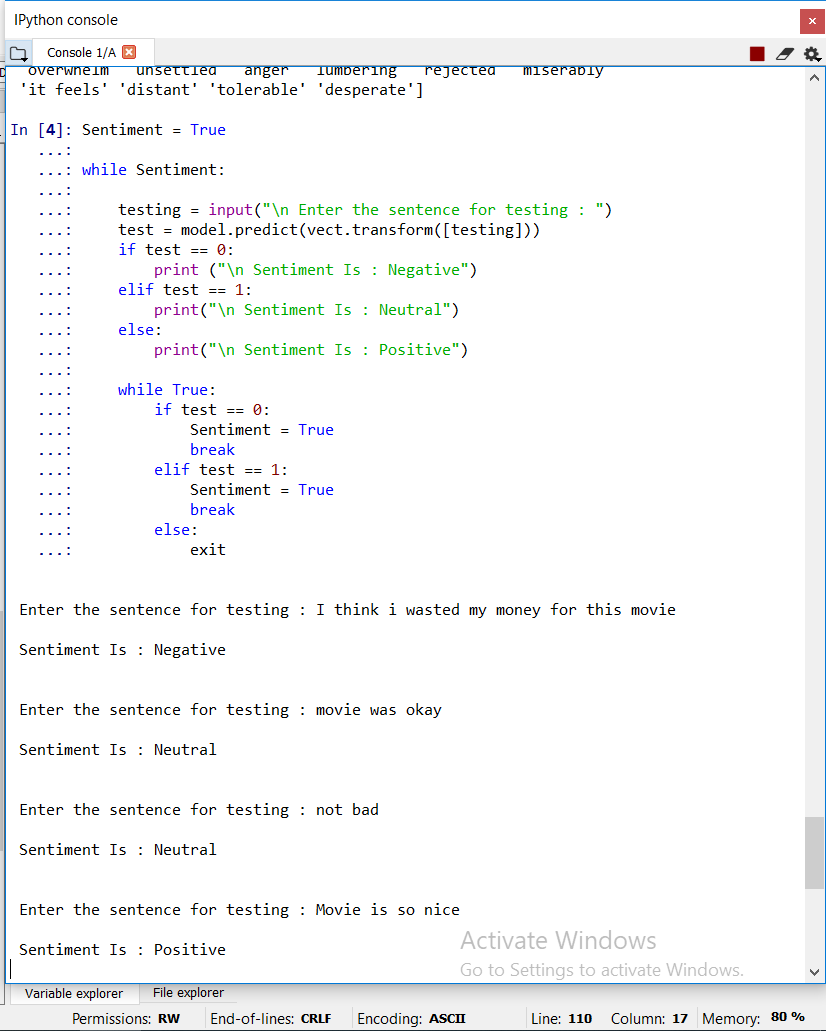


This accuracy level suggests that the model can be used effectively as a solution to the suggested business problem and this will be applied in the deployment phase.

The main challenges of the model were

* Contrasts with standard text-based categorization
* Domain dependent
* Sarcasm/Thwarted expressions
* the sentences/words that contradict the overall sentiment of the set are in majority

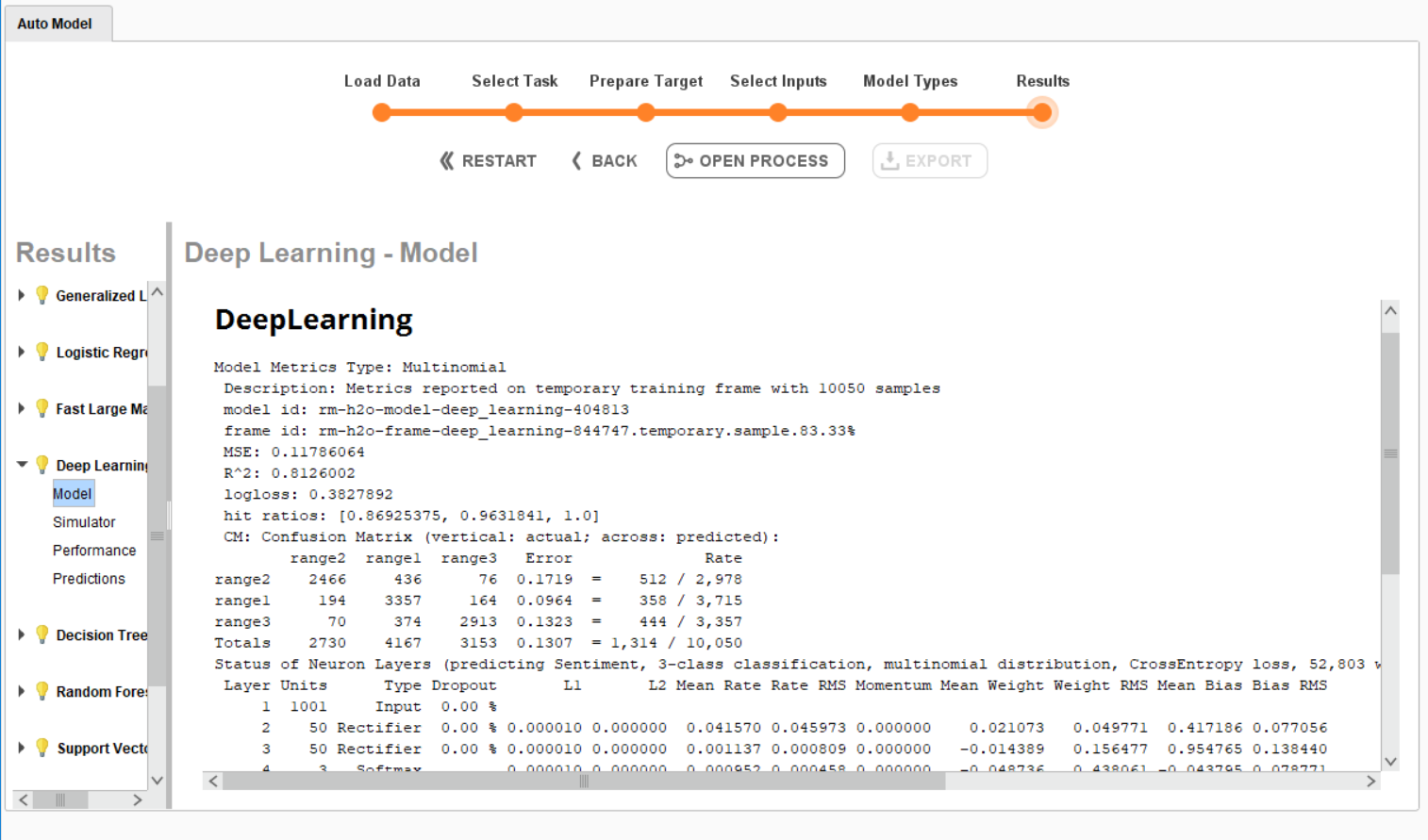
**Example**: The actors are good; the music is brilliant and appealing. Yet, the movie fails to strike a chord.



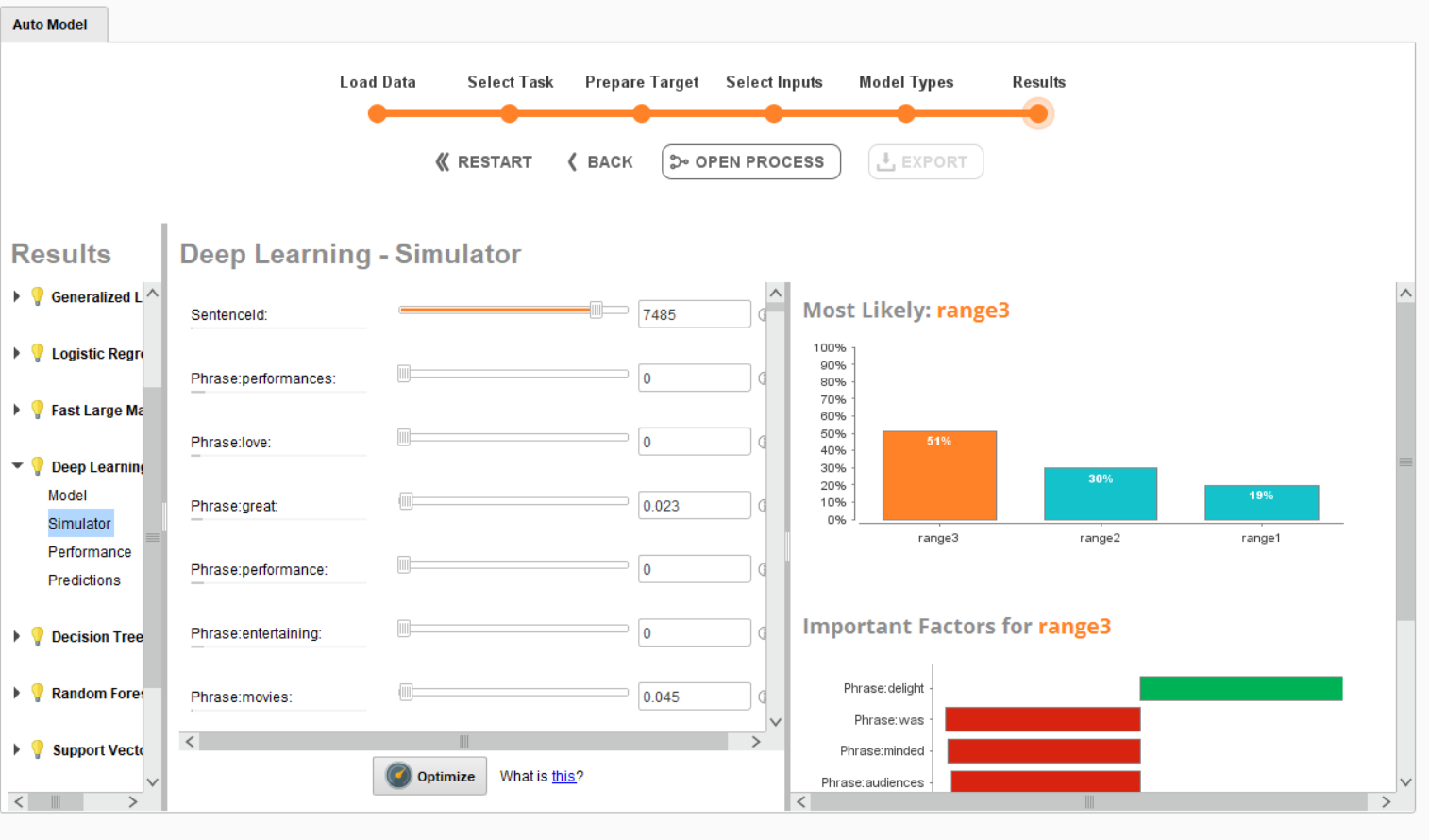
The model was tested by giving different phrases which belongs two different sentiment categories. We have experimented with 25 phrases and 23 of them were returned the correct sentiment

**Deployment**

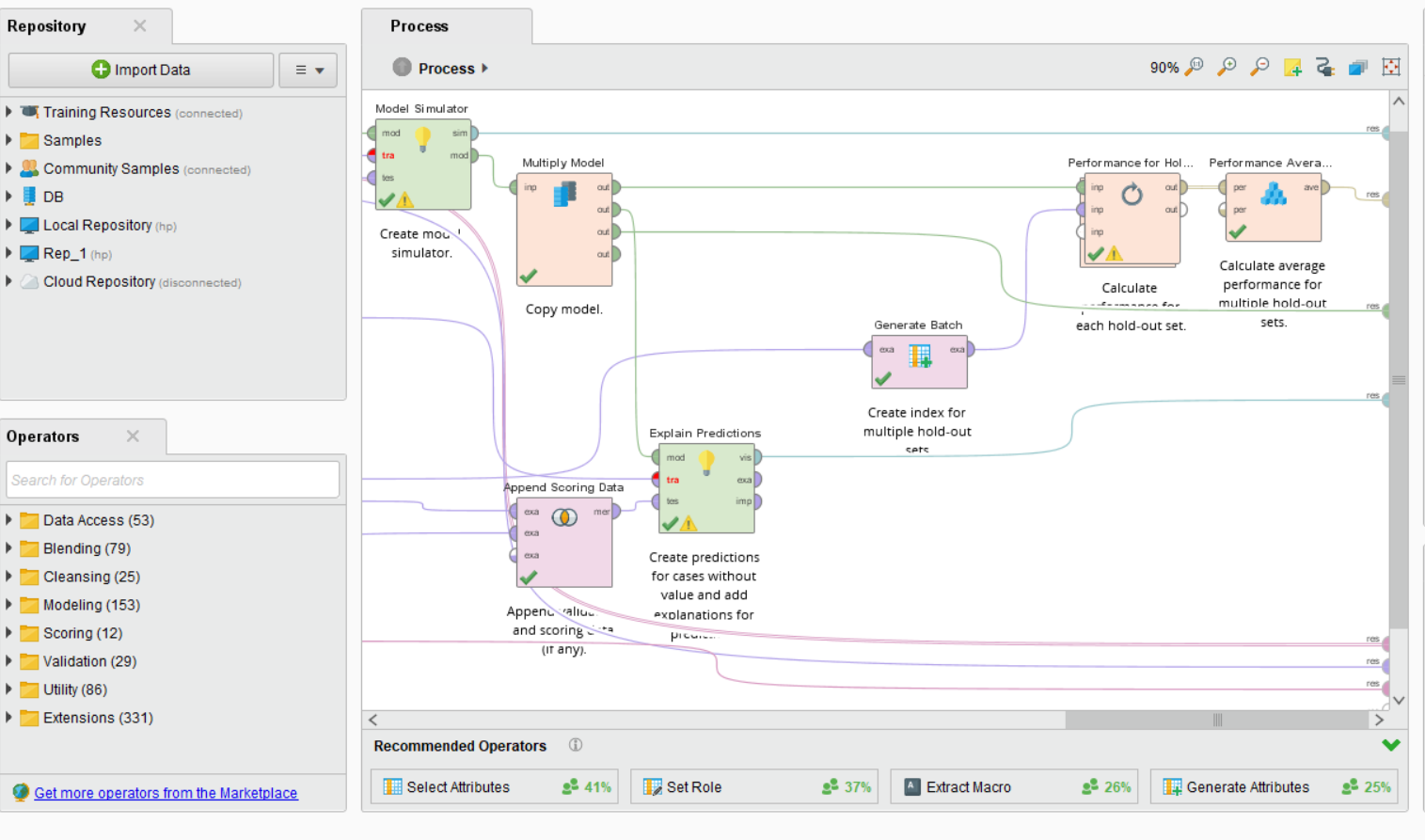
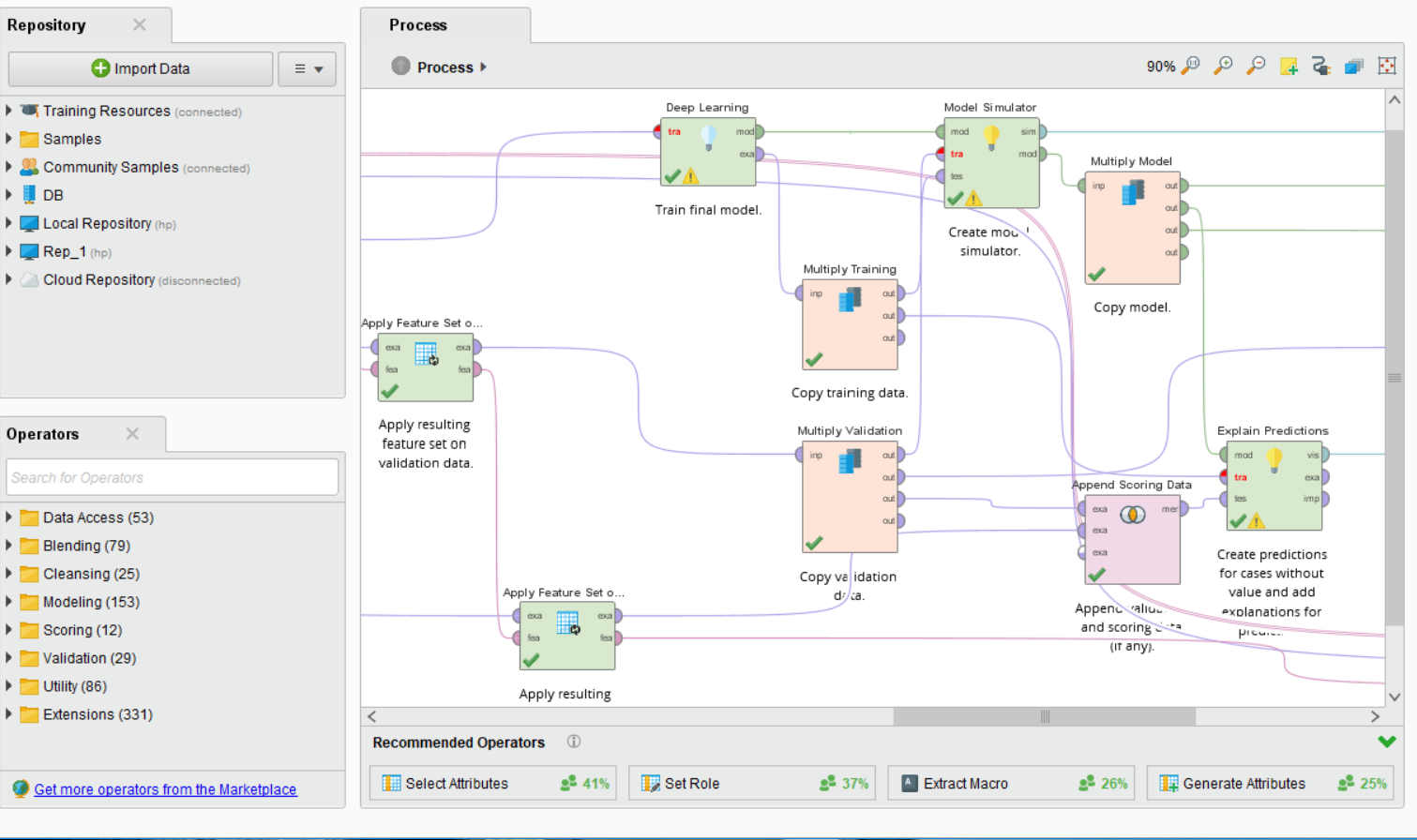
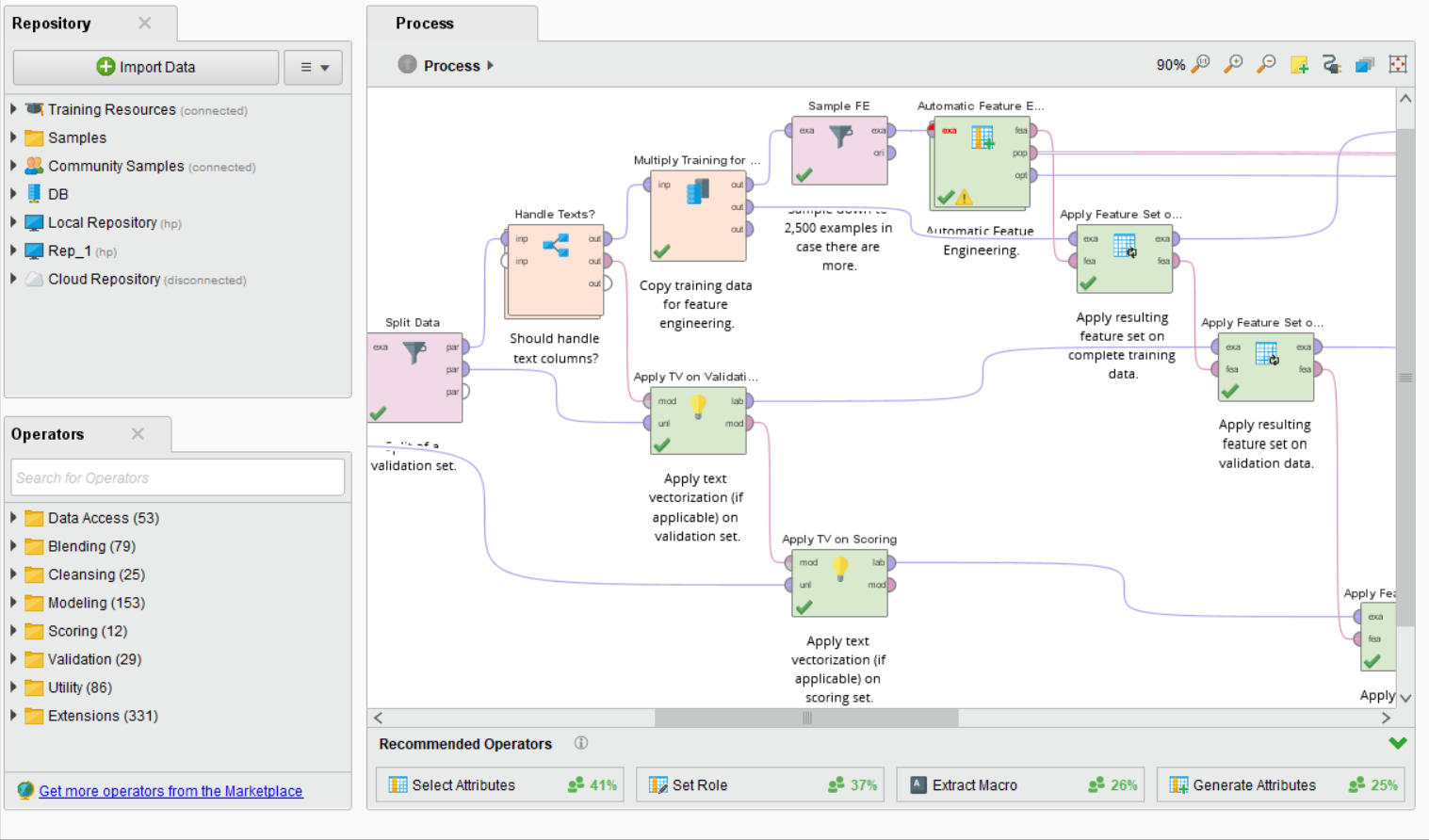
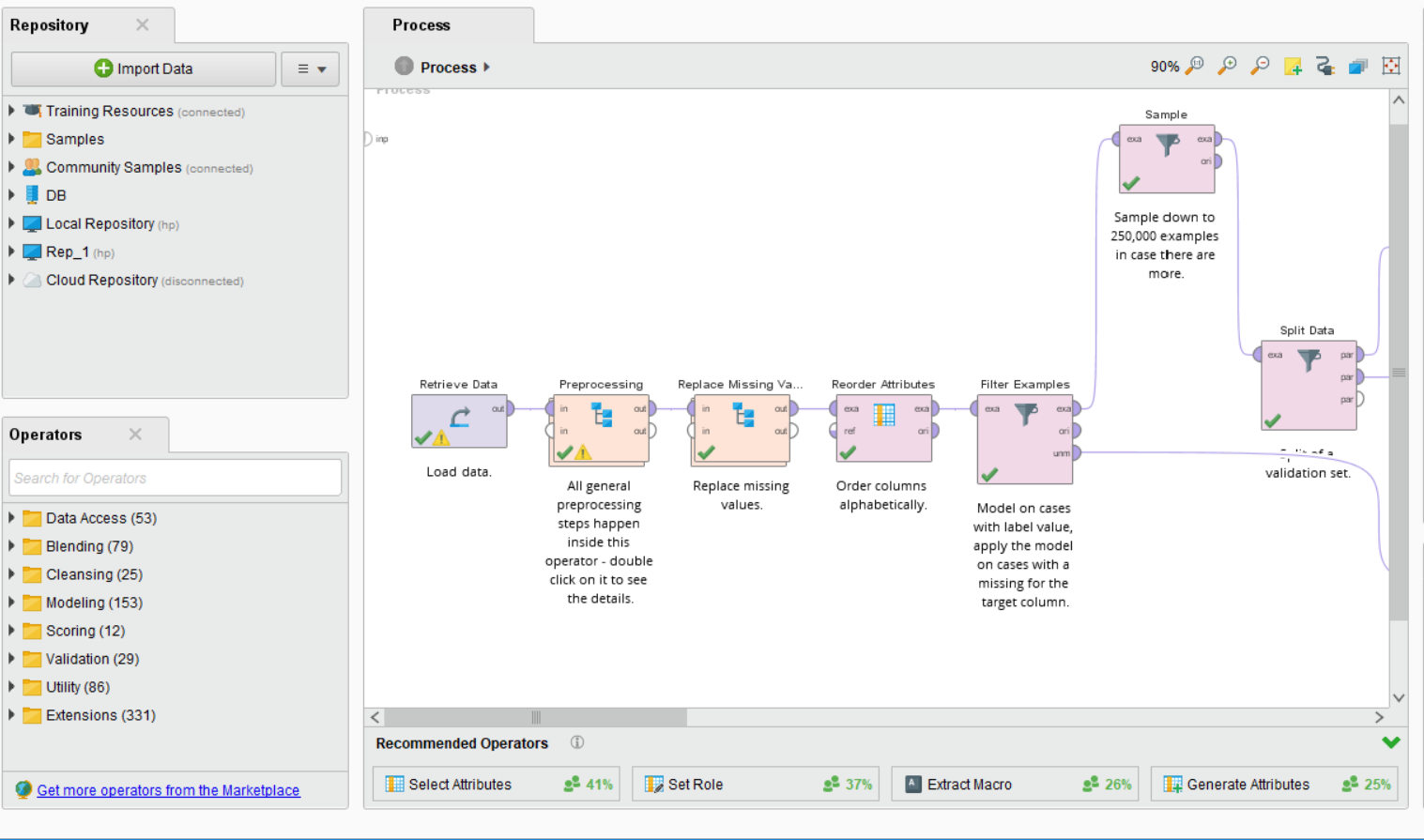
This is the final stage of the process. We have already created a model for implementation, but we have to implement it. This stage is for that. The implementation can be in any form. For example, visualization, making we application, web page etc.



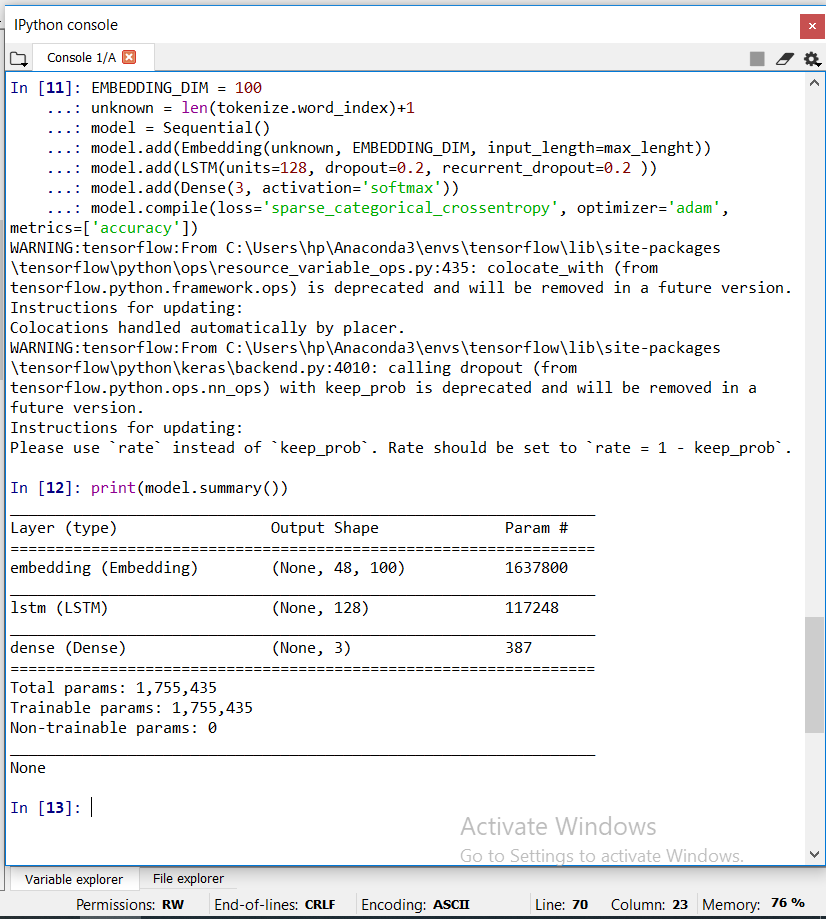
This is the Model type and parameter selection of Deep learning (Auto Model)



This following four images are the design of the Deep Learning process (Preprocessing, Filtering, Parameter selection, Model selection, etc)

****

In python, Deep learning we used LSTM model and represent three classification label (Using Dense). This is the summary of our model



**Conclusion**

Analyzing tone and extracting sentiments form user text input is not an easy task specially that we have sentiments, for example it is hard to distictish between from somewhat positive and positive classes. In the rest part of our project we investigated the current state-of-the-art for the problem of sentiment analysis in order to discover which are the most e‑cient approaches. We started by analyzing the dataset provided in this project, we extracted some statistics which can help us in better understanding the data. Thereafter we proceeded with preprocessing the dataset in order to obtain a good representation of the text input. Finally we explored some classification algorithms which could be employed for this task. We considered Deep Learning. in order to obtain a baseline result for the classification, as well as logistic regression with various kernel functions.

While concluding we can say that the project had successfully being driven by the CRISP-DM methodology in order to attain the final goal. We analyzed the model by giving new phases and the model was able to predict which class the phrase belongs to, with a greater accuracy . Overall the predications were of good accuracy

**References**

Wikipedia.org. (2019). *Wikipedia*. [online] Available at: https://www.wikipedia.org/ [Accessed 05 april. 2019]

RapidMiner. (2019). *Visual Workflow Designer for Data Scientists | RapidMiner Studio*. [online] Available at: https://rapidminer.com/products/studio/ [Accessed 01 April. 2019].

Kaggle.com. (2019). *Kaggle: Your Home for Data Science*. [online] Available at: https://www.kaggle.com/ [Accessed 05 April. 2019].

aylien.com. (2018). AYLIEN: Making Sense of the Incomprehensible. [online] Available at: https://aylien.com/ [Accessed 15 march . 2019]