Internship Report

A thesis Submitted in partial fulfillment

of the requirements for the degree of

Master of Technology in Data Science

(Business Analytics)

By

Aafreen Kazi

(SAP ID: 70271019018)

under the guidance of Mr. Tushar Kamble (Industry Mentor) & Prof. Sarada Samantaray

Department of Data Science

NMIMS University

Mumbai

April, 2021

CERTIFICATE

This is to certify that the thesis entitled **"Internship Report"** is a bonafide work of **"Aafreen Kazi (SAP ID:70271019018)"** submitted to the NMIMS University in partial fulfillment of the requirement for the award of the degree of **"Master of Technology"** in **"Data Science (Business Analytics)"**.

Prof. Sarada Samantaray Associate Dean NMIMS

ACKNOWLEDGEMENT

It is with a feeling of great pleasure that I would like to express my most sincere heartfelt gratitude to **Mr. Tushar Kamble (Assistant Manager – Corporate Technology Development, Crisil Ltd.)** for his steady and able guidance throughout my internship and writing of this thesis. The internship opportunity with CRISIL Limited was a great platform for learning about the financial industry. The projects were conducted under the guidance of our individual external mentors Mr. Manish Singh and Mr. Tushar Kamble. I would like to thank them for supporting and providing me with valuable insights throughout the journey of my internship in spite of their busy schedules. Also, I would like to thank Ms. Achsah Ruben, Ms. Hansini Manohar, Mr. Krishna Mishra and Ms. Kruti Desai for their continuous help and support throughout my internship.

I express my sincere thanks to **Prof. Sarada Samantaray (HOD, Data Science)** for his guidance and for providing the necessary facilities in the department. I am also thankful to all the staff members of the department of Data Science especially **Prof. Siba Panda** for their inspiration and help.

DISCLAIMER

This report has been prepared by Aafreen Kazi pursuant to their internship with CRISIL Limited. This report is solely for use by the Interns for academic purposes for submission to NMIMS, MPSTME in connection with their internship at CRISIL Limited and is not intended for public or commercial use or constitute any advice, recommendation or opinion on any Company or security. The opinions and comments expressed in this report do not reflect the opinions or comments of CRISIL Limited, its affiliates, or their personnel. No person is authorized to use this report, rely on it or quote it, for any purposes. CRISIL shall not be liable in any manner for any use of this report. All the data used in the report has been sourced from the public domain.

Abstract

This report describes the internship experience at CRISIL Ltd. An opportunity to work with the Product & DoS advanced analytics team at CRISIL Ltd. from December 2020 to May 2021 has provided organizational knowledge on how exactly is the field of data science helping business organizations to automate the traditional procedures. It gave an insight about the collection, generation and handling of unstructured industrial data along with the implementation of deep learning models as well as sequence models for NLP-related tasks. Some of the largest commercial data was explored and various insights were generated after performing methods of data wrangling and exploratory data analysis. Implementation of deep learning models on huge industrial image data as well as performing a number of pre-processing operations on the industrial text data really provided an understanding about how industries are integrating automation with data handling to yield data science products. Some of the most important methodologies that were performed in the internship have been included in this report.

TABLE OF CONTENTS

Chapter 1

Introduction to CRISIL Ltd.

1.1 About CRISIL Ltd.

CRISIL is an agile and innovative, global analytics company driven by its mission of making markets function better. It is an Indian analytical firm that provides ratings, research, and risk and policy advisory services and is a subsidiary of American company S&P Global. A strong track record of growth, culture of innovation and global footprint sets CRISIL apart from a traditional ratings company. Over the years, it has delivered independent opinions, actionable insights, and efficient solutions to over 100,000 customers. CRISIL's businesses operate from India, the US, the UK, Argentina, Poland, China, Hong Kong and Singapore. Currently, CRISIL has also extended its footprints in Australia.

CRISIL is majority owned by S&P Global Inc., a leading provider of transparent and independent ratings, benchmarks, analytics and data to the capital and commodity markets worldwide. CRISIL's clients range from micro, small and medium companies to large corporates, investors, to top global financial institutions. Clients associated with CRISIL include commercial and investment banks, insurance companies, private equity players and asset management companies globally. Apart from that, CRISIL also works with governments and policy makers in the infrastructure space in India and in other emerging markets. It provides analyses, insights and solutions to help lenders, borrowers, issuers, investors, regulators and intermediaries make sound decisions. Also, it helps clients to manage and mitigate risks, take pricing and valuation decisions, reduce time to market, generate more revenue and enhance returns. CRISIL has grown a lot in terms of data science and analytics over the years and is not just a traditional ratings company.

1.2 Work Culture at CRISIL Ltd.

The Product & DoS team at CRISIL Ltd. tries to adopt the agile work culture instead of the traditional macro-management techniques. An agile work culture focuses more on finding the best way to solve a problem rather than applying the same procedural approach to every situation. It tries to innovate new procedures in order to yield better approaches to a problem than the traditional ones. This flexibility applies not just to how work is done, but also to the work environment itself. It provides an experience to handle problems or situations in our own innovative ways instead of directly following the orders. One can not only learn how to perform a task but an agile environment enables one to explore different solutions to provide a solution to the task. Agile work environment not only provides the organization with a better innovative approach for a given problem but also enables the individuals working an opportunity to learn and grow.

Agile work environment enables the individuals working in the organization to work as a team instead of engaging in work politics. One works as a part of a cross-functional, self-organizing team that has end-to-end responsibilities through autonomous team culture. Autonomy enabled the team to learn quickly by enabling decisions to happen locally. Basically, aile work culture allows teams and individuals to be more adaptive, flexible, innovative and resilient when dealing with complexity, uncertainty and change.

Chapter 2

Intelligent Document Processing (IDP)

2.1 Introduction to Intelligent Document Processing (IDP)

Document processing is the technique of storing, transforming, extracting and managing the information present in the documents of an organization. Business organizations have been performing it daily for many years in order to handle huge volumes of documents. The document processing involves the manual labour of doing the same tasks again and again that are repetitive in nature. As the volume of documentation increases, these tasks not only become labour intensive but also less rewarding for the employees working in the organization as they are not learning any new skills. This in turn decreases the efficiency of an organization. Therefore, an automated process for document processing is the need of the hour. Intelligent Document Management is a technique that involves the application of artificial intelligence technologies like machine learning, natural language processing (NLP), intelligent character processing (ICR) in order to process documents that would generally require manual labour. In simpler terms, it is the process of removing the repetitive and boring tasks associated while dealing with the document data. It is a form of Intelligent Process Automation (IPA) that develops solutions which are more time and cost efficient than traditional manual processes [9].

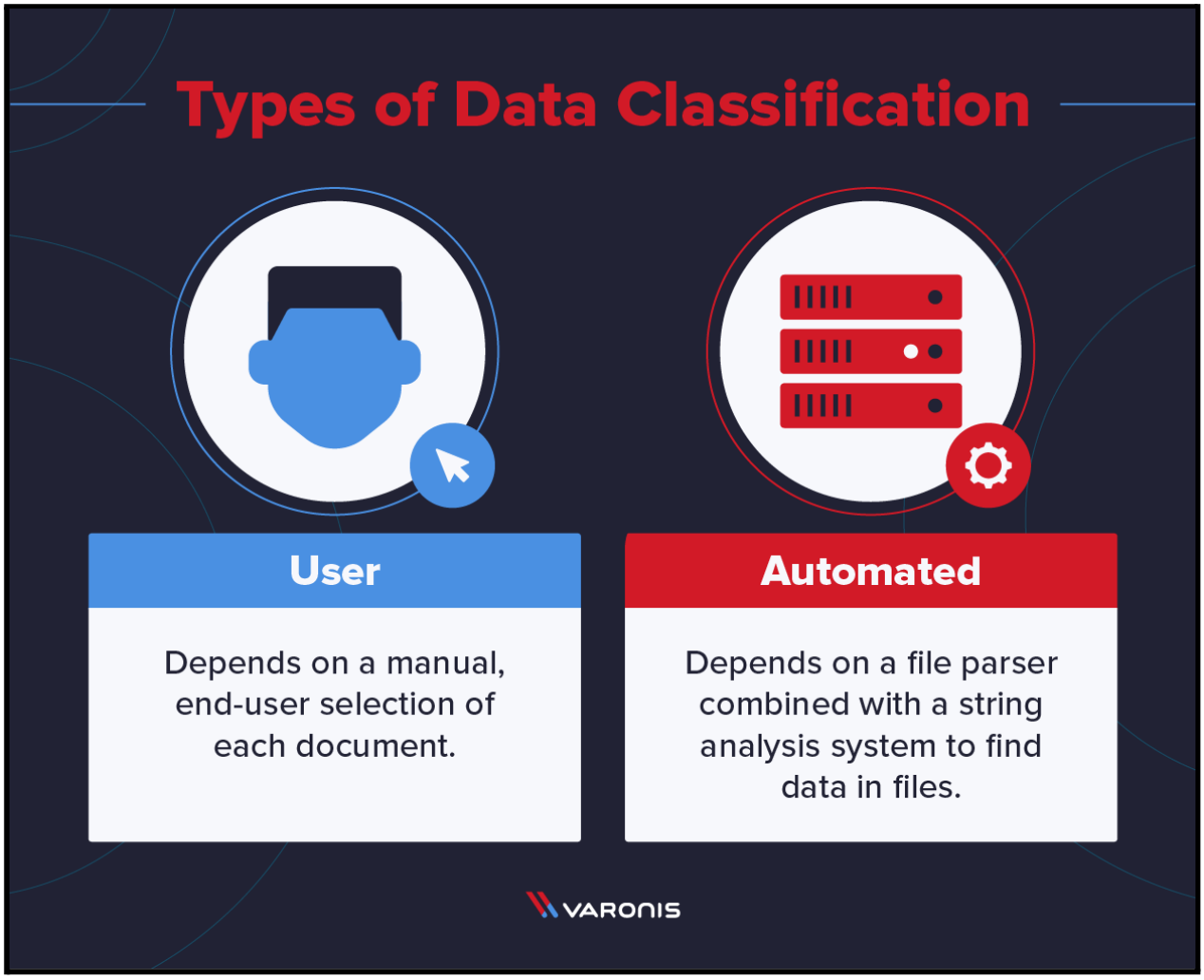
Image Source: Canva

Every company or business organization needs to have proper financial documentation. As the business grows, these financial documents increase which results in a huge volume of financial documents. Thus, the amount of tedious work done in every organization most importantly financial organizations is quite high that in turn creates high impact of intelligent document processing in hospitals and providers. Financial documentation is a process which involves sorting, labelling, storing and searching of documents that involves a lot of time [1]. The process of intelligent documentation could help the financial firms and its employees to deal with their large chunks of document data in order to manage their invoices, processes related to order management and understanding the financial data.

2.2 Relevance of Intelligent Document Processing in Business Organization

A process of proper document handling could not only result in an improvement in the smooth function of a business entity but could also result in an increased profitability by applying the actionable insights into the business decisions. The constraint of worker finiteness plays a major role in decreasing the quality of customer experience for a business organization. These constraints could be solved by shifting away from pull technology where the user has to actively initiate the request for information towards push technology where available information is automatically delivered without user intervention [2]. The intelligent document processing not only deals with improving the efficiency of document processing techniques in a business organization but also improves employee satisfaction as they do not have to deal with tedious, repetitive work thereby providing them the opportunity to learn something new and more productive.

A study has revealed that businesses spend an average of $20 in labor for filing a single document, $120 to search for each misplaced document and around $220 to recreate every lost document in the US [12]. Thus, a digital system that could help the firms to decrease these costs in order to increase their profitability becomes quite vital. The intelligent document processing could be used to decrease the time required to process document data with quality similar to the humans. Since the functioning of any business organization is largely dependent on the documents, a huge amount of document data is generated. Thus, a proper handling and management of the document data through automated intelligent systems becomes quite important.



Difference between intelligent document processing and automated document processing

The intelligent documentation process involves four stages. The first stage is the pre-processing that deals with uniformization and preparation to improve the quality of the raw data. The second stage involves the document classification that separates documents based on their contents whereas the third stage involves the extraction of relevant information from the classified documents. The last stage deals with improving the extraction results in order to get the information that provides actionable insights.

This project involved the first two steps of document processing. A taxonomy from the perspective of document management or content management is the task of classifying content into groups. Each group has its own unique characteristics as well as metadata. Automating text classification is the process of categorizing text documents into predefined classes through the application of algorithms. An actual industrial finance data was considered for the task of text classification.

Chapter 3

Data Generation and Pre-processing of the Industrial Data

3.1 Importance of dealing with unstructured data

It has been reported that the maximum amount of time in a data science project goes to the pre-processing of the data to prepare it for applying the data science models on it. If data pre-processing goes wrong then there are high chances of getting errors in the deep learning and machine learning models. Data is the lifeblood of business and it comes in a huge variety of formats which ranges from strictly formed relational databases to the complex text data. All of that data, in all different formats, can be sorted into one of two categories: structured and unstructured data. Structured data is highly specific and is stored in a predefined format, where unstructured data is a conglomeration of many varied types of data that are stored in their native formats. It is easy to handle and a lot of hands on experience was already gained in it throughout the previous syllabus. However, in industries, collecting properly structured data is extremely difficult. This internship has provided a hands on experience of how to process unstructured data to prepare it for data science related application. Around 80% of business data are unstructured which makes intelligent document management a need. Since the normal digital systems could not capture the semantics and syntactics of the document data, intelligent document management could be applied to give human-like results. It avoids the chain of tedious, boring and time-consuming paperwork to allocate free time and resources to spend on more important and productive tasks. Raw unstructured data was converted into the data type required by the project. Conversion of any format of the document into images and then using those images as an input for the text classification model.

3.2 Data Collection and Conversion

Generally, the data collected were in PDF format. Various types of PDF formats were collected. In order to further pre-process the data for running them into deep learning models, each page of the PDF was converted into an image using PyPDF2 library on Python. However, some PDFs were of corrupt type which resulted in a challenge to convert the given set of PDFs into images. This issue was solved through using the pikepdf. It is a Python library for reading and writing PDF files. It is based on QPDF, a powerful PDF manipulation and repair library. QPDF is a command-line program that does structural, content-preserving transformations on PDF files. It could have been called something like pdf-to-pdf. It includes support for merging and splitting PDFs through the ability to copy objects from one PDF file into another and to manipulate the list of pages in a PDF file.

3.3 Image Pre-processing

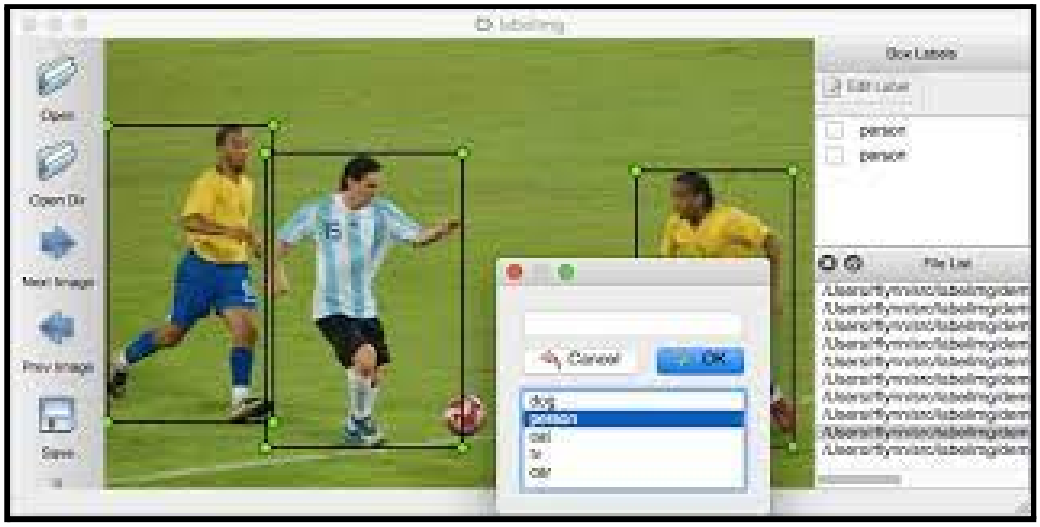
It is important to know what exactly image processing is and what is its role in the bigger picture before diving into its how's. Image Processing is most commonly termed as 'Digital Image Processing' and the domain in which it is frequently used is 'Computer Vision'. The aim of pre-processing is an improvement of the image data that suppresses unwilling distortions or enhances some image features important for further processing, although geometric transformations of images (e.g. rotation, scaling, translation) are classified among pre-processing methods here since similar techniques are used. Labelling of unstructured image data becomes quite important for making deep learning algorithms to learn the objects present in an image.

3.3.1 Image normalization and resizing using OpenCV

OpenCV (Open Source Computer Vision Library) is an open-source library that includes several hundreds of computer vision algorithms. It consists of an image processing module that includes linear and non-linear image filtering, geometric image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, histograms, and so on. Image Normalization is a process in which we change the range of pixel intensity values to make the image more familiar or normal to the senses, hence the term normalization. Data normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network. Since neural networks receive inputs of the same size, all images need to be resized to a fixed size before inputting them to the CNN. The larger the fixed size, the less shrinking required. Less shrinking means less deformation of features and patterns inside the image.

3.3.2 Data labelling with LabelImg

Labeled datasets help to train the corresponding machine learning or deep learning models to identify and understand the recurring patterns in the input fed into them for delivering accurate output. After being trained by annotated data, deep learning models can start recognizing the same patterns in the new unstructured data. Converted image data was labelled through the help of LabelImg. Relevant data was annotated based on the final categories of the document. Since object detection needed to be performed on the image data, the images were annotated using the bounding box. LabelImg is a free, open source tool for graphically labeling images. It's written in Python and uses QT for its graphical interface. It's an easy, free way to label a few hundred images.



Demonstration of LabelImg Tool

The LabelImg tool played a significant role in the process of data pre-processing. Basically, it provides the coordinates of the object detected. It supports labelling in Pascal VOC XML or YOLO text file format. The Pascal VOC format gives the minimum and maximum x and y coordinates for each class of object labelled in an image. The YOLO format provides float values relative to width and height of image and therefore the coordinates provided in the YOLO file format lies in the range of 0.0 to 1.0. The value of the x coordinate is equivalent to the ratio absolute value of the x coordinate and image width whereas the value of the y coordinate is equivalent to the ratio of absolute height of the object total image height. As per the experience at the internship, the default Pascal VOC XML format for creating labels is strongly recommended if the data has to undergo further pre-processing whereas if the task is just related to object detection, the YOLO format should be recommended. Two types of objects were labelled namely paragraphs and tables. The corresponding label of the image objects were texts that were detected using Pytesseract and the output of the Pytesseract algorithm was manually checked in order to avoid any discrepancies.

3.4 Text pre-processing

In any machine learning or deep learning task, cleaning or preprocessing the data is as important as model building if not more. And when it comes to unstructured data like text, this process is even more important. General pre-processing tasks include converting the words into a uniform case either upper or lower case. Generally, the text is converted into lowercase. The two major pre-processing steps involved in the text include tokenization and converting words into vectors.

3.4.1 Tokenizing text

Tokenization is the process of turning a meaningful piece of data, such as an account number, into a random string of characters called a token that has no meaningful value if breached. The text could be tokenized based on the sequence of words or character.

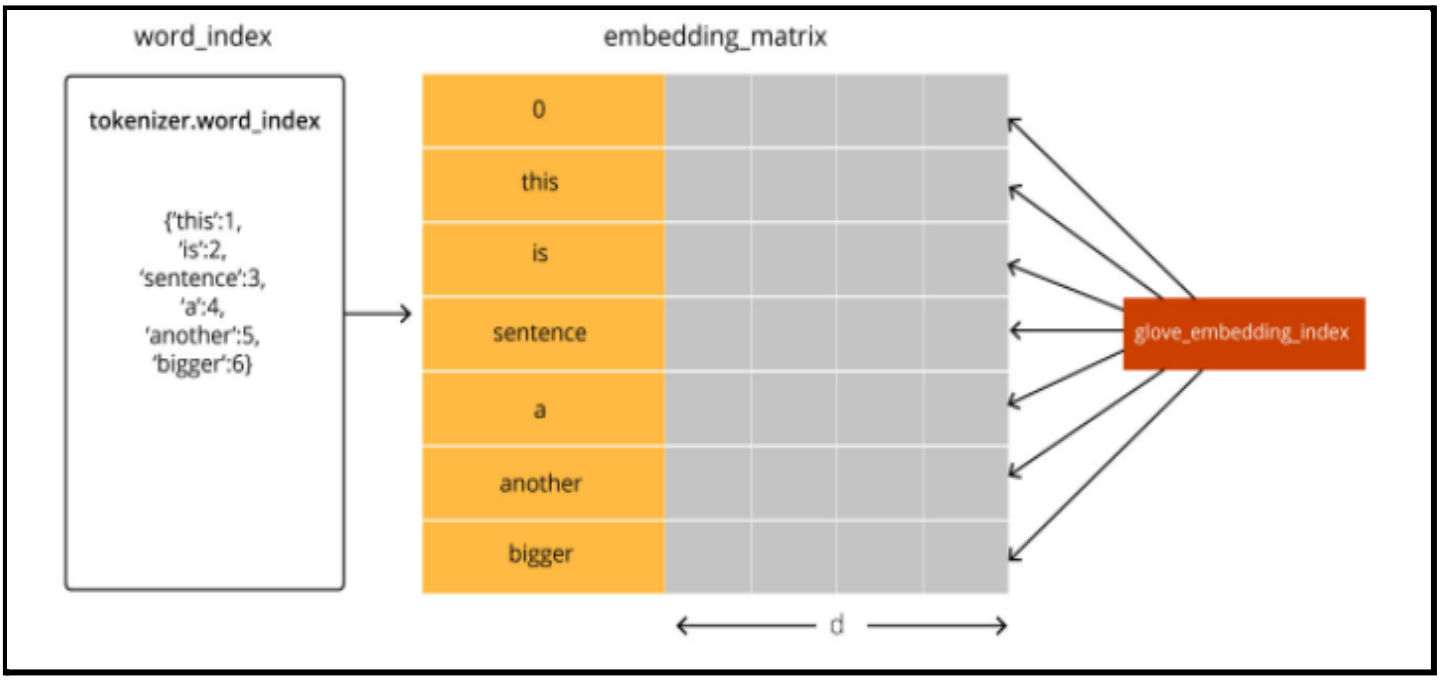
Word Tokenization: Tokenization is essentially splitting a phrase, sentence, paragraph, or an entire text document into smaller units, such as individual words or terms. Each of these smaller units are called tokens. In our financial data, each word was considered as a single unit (token). It should be noted that the text classification process does not involve the removal of special characters since they holded a lot of significance in the given financial data. A separate token was created for each special character encountered in the text. Each tokenized word was assigned an index that helps in the generation of final word embeddings.

Character Tokenization:Tokenizing a given sequence of words into their corresponding character solves the out-of-the-vocabulary problem but the length of the input and output sentences increases rapidly as we are representing a sentence as a sequence of characters. As a result, it becomes challenging to learn the relationship between the characters to form meaningful words.

3.4.2 Creating Word Embeddings

Vectorization process of a word is a natural language processing (NLP) process that uses language models to map words into vector space. A vector space represents each word by a vector of real numbers. It also allows words with similar meanings to have similar representations. Among various word embedding technologies, GloVe embeddings was implemented for the taxonomy classification model.

GloVe is an unsupervised learning technique used to obtain the corresponding vector representations for words. It was used for the text embedding and is a pre-trained model for generating word embeddings that functions on the principle of word co-occurrence matrix that tabulates how frequently words co-occur with one another in a given corpus. The generated word embeddings were treated as an initial input for NLP downstream tasks such as text classification.



Generation of the GloVe Word Embeddings from tokenized text

The embedding matrix used to train the text classification models were built from the Glove model and was integrated in the weight matrix of the neural network. The matrix dimensions were the size of the vocabulary by the number of features Glove generates for each word. Generally, the deep learning or sequence models expect that each text sequence (each training example) will be of the same length (the same number of words/tokens). Text sequences could be padded with the help of the max length which is the maximum length of the sequence of text which one is dealing with.

Chapter 4

Application of Deep Learning and Sequence Models to the Industrial Data

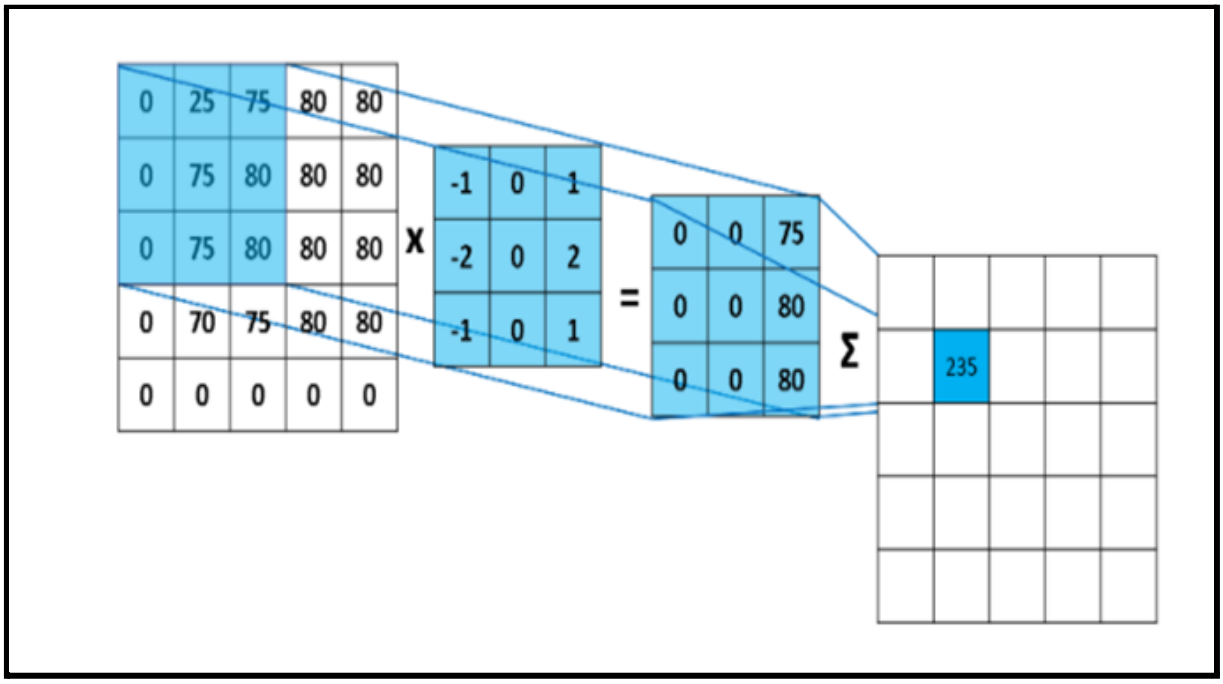
4.1. Introduction to the Models Applied

The internship has provided the opportunity to work with different deep learning and sequence models on organizational data. It gave an insight about how different fields of data science like computer vision, natural language processing could improve the way in which companies work. The previous workflows provided two outputs: a labelled pre-processed image data and a clean text data ready to be implemented in the deep learning model. Image to text conversion and text classification were the two deep learning algorithms that were implemented.

4.1.1 CNN Model

A Convolutional Neural Network (CNN, or ConvNet) are a special kind of multi-layer neural networks, designed to recognize visual patterns directly from pixel images with minimal preprocessing. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm — a Convolutional Neural Network. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. One major problem with computer vision problems is that the input data can get really big.

Suppose an image is of the size 68 X 68 X 3. The input feature dimension then becomes 12,288. This will be even bigger if we have larger images (say, of size 720 X 720 X 3). Now, if we pass such a big input to a neural network, the number of parameters will swell up to a huge number (depending on the number of hidden layers and hidden units). This will result in more computational and memory requirements – not something that can be dealt with.

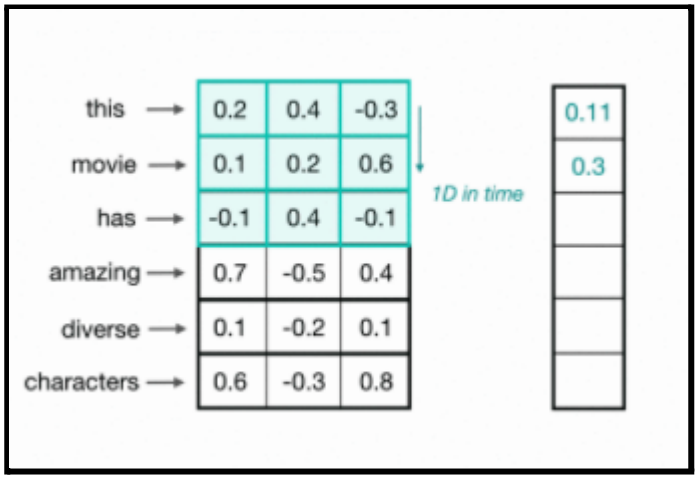


Working of the Convolutional Neural Network

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area. This study focuses on determining the freshness of three types of fruits using different architectures of the convolutional neural networks. This study focuses on two classical CNN models i.e. AlexNet, VGG-16 and a residual network ResNet50.

4.1.1.1 1D CNN Model

A 1D convolution neural network involves the moving of the kernel in only one dimension which is the time or sequence of words for a text. A single kernel will move one-by-one down a list of input embeddings, looking at the first word embedding (and a small window of next-word embeddings) then the next word embedding, and the next, and so on. The resultant output will be a feature vector that contains about as many values as there were input embeddings, so the input sequence size does matter.



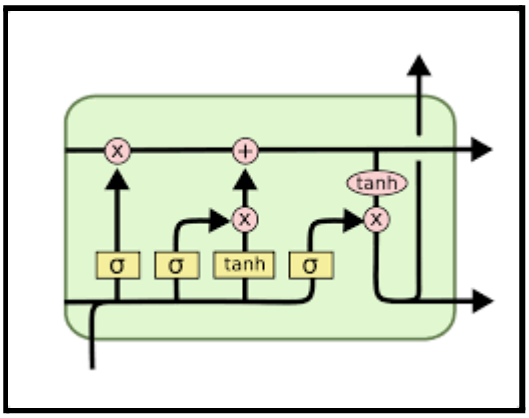
Working of 1D CNN for text data

Generally, CNN can detect edges, distribution of colours of text which makes these networks very robust in text classification and other similar data which contain similar spatial properties. 1D CNNs are also used on audio data since we can also represent the sound and texts as a time series data.

4.1.2 LSTM Model

LSTM is a recurrent neural network (RNN) architecture that remembers values over arbitrary intervals. One of the appeals of RNNs is the idea that they might be able to connect previous information to the present task, such as using previous video frames might inform the understanding of the present frame. If RNNs could do this, they'd be extremely useful. Sometimes, we only need to look at recent information to perform the present task. For example, consider a language model trying to predict the next word based on the previous ones. If we are trying to predict the last word in "the clouds are in the sky," we don't need any further context – it's pretty obvious the next word is going to be sky.

In such cases, where the gap between the relevant information and the place that it's needed is small, RNNs can learn to use the past information. However, if the sequence of words is large then LSTMs come into picture. An LSTM is well-suited to classify, process and predict time series given time lags of unknown duration.



Single Cell of a LSTM model

Relative insensitivity to gap length gives an advantage to LSTM over alternative RNNs, hidden Markov models and other sequence learning methods. The long-term memory is usually called the cell state. The cell state is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged. The looping arrows indicate the recursive nature of the cell. This allows information from previous intervals to be stored within the LSTM cell. Cell state is modified by the forget gate placed below the cell state and also adjusted by the input modulation gate. From the equation, the previous cell state forgets by multiplying with the forget gate and adds new information through the output of the input gates. The key to LSTMs is the cell state, the horizontal line running through the top of the diagram.

4.1.2.1 Bidirectional LSTMs

In traditional neural networks, all the inputs, and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words(as the next word will depend on your previous input). Bidirectional LSTMs are an extension of traditional LSTMs that can improve model performance on sequence classification problems. In problems where all timesteps of the input sequence are available, Bidirectional LSTMs train two instead of one LSTMs on the input sequence. As in NLP , sometimes to understand a word we need not just to the previous word , but also to the coming word , like in the example of the sentences given below. Here for the word “Teddy” , we can’t just say whether the next word is going to be “Bears” or “Roosevelt”, it will depend on the context of the sentence.

Example of a sentence in which the importance of Bidirectional LSTMs could be observed

Initially, the application of providing the sequence bi-directionally was justified in the domain of speech recognition since there was evidence which suggested that the context of the whole utterance could be used to interpret what is being said rather than a linear interpretation.

Bi-lstm is a general architecture that can use any RNN model. For a bi-directional LSTM, one should apply forward propagation twice, one for the forward cells and one for the backward cells. The first on the input sequence as-is and the second on a reversed copy of the input sequence. The application of bidirectional LSTMs may not make sense for all sequence prediction problems, but can offer some benefit in terms of better results to those domains where it is appropriate.

4.2 Implementing and Selecting the Configuration of Model Architecture

Architecture of a deep learning as well as sequence model plays the most important role in determining the overall performance of a model. After selecting which models to consider, the configuration of its architecture becomes quite important. 1D CNN architecture was used for text classification whereas the encoder-decoder architecture was applied for encoder decoder architecture. It has been reported that methods such as transfer learning and Multi-Task Learning make the component development process more efficient. Tasks in this project integrated the model generated from transfer learning and model generated at individual level to produce high-performance model architectures. The models were built using Anaconda Python IDE.

4.2.1 1D CNN Architectures for Text Classification

A text categorization task consists of a training phase and a text classification phase. The former includes the feature extraction process and the indexing process which was already performed in the pre-processing stage. In order to explore more about the application of 1D CNN in the process of text classification, AlexNet architecture was also explored. The model architecture of Alexnet was kept similar to the 2D CNN. However, instead of using 2D kernel and other layers, 1D CNN layers were used to build the model architecture. For text classification, it was observed that a simple architecture Similarly, a one dimensional architecture of VGG-16 was imitated. The 1D CNN worked better as compared to other complex models like AlexNet, VGG-16 model in one dimensional data.

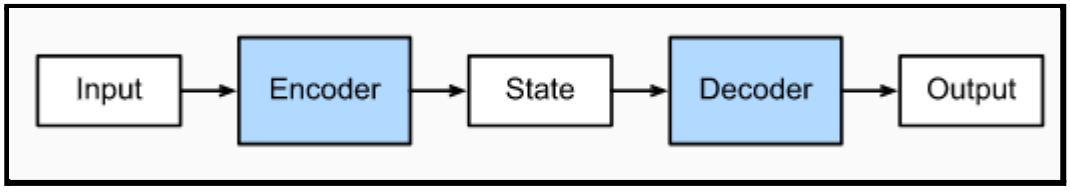
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding (Embedding)        (None, 2687, 100)         888300
_________________________________________________________________
conv1d (Conv1D)              (None, 2672, 16)          25616
_________________________________________________________________
max_pooling1d (MaxPooling1D) (None, 1336, 16)          0
_________________________________________________________________
flatten (Flatten)            (None, 21376)             0
_________________________________________________________________
dropout (Dropout)            (None, 21376)             0
_________________________________________________________________
dense (Dense)                (None, 5)                 106885
=================================================================
Total params: 1,020,801
Trainable params: 1,020,801
Non-trainable params: 0
_________________________________________________________________
```

Simple1D CNN Model for Text Classification

Total parameters generated by a simple CNN Model for text classification was 1,020,801. A simple architecture yields better results for one dimensional data as compared to the complex networks. It was also observed that as the number of layers were increasing for the 1D CNN architecture, the text accuracy decreased significantly. Model configuration of the 1D-CNN was used to classify. The accuracy was found to be around 97% for a generalized data and its implementation on the business data could have been done later.

4.2.2 Encoder-Decoder architecture for Image to Text Conversion

The encoder-decoder architecture handles the natural language conditions in which the inputs and their corresponding outputs are the sequences of variable-length. To handle this type of inputs and outputs, we can design an architecture with two major components. The first component is an encoder: it takes a variable-length sequence as the input and transforms it into a state with a fixed shape. The second component is a decoder: it maps the encoded state of a fixed shape to a variable-length sequence.



Encoder Decoder Architecture

Internship allowed the exploration of various strategies of the encoder-decoder architecture on financial data. After researching a number of strategies, the encoder-decoder architecture was found to be efficient. Encoder tries to detect various objects in an image and extract its corresponding features. Decoder takes in the input features provided by the encoder and tries to generate the corresponding text in sequence. Encoder-decoder architecture was implemented for converting images into the relevant text based on the objects detected.

Chapter 5

Evaluating the Performance of the Model based on Business Understanding

5.1. Evaluate the performance of the model based on model metrics

Evaluation of the image to text model becomes a bit challenging since it differs greatly from the general image classification or text classification task in which categorical evaluation metrics are applied. The closer an image to text translation is to a professional human translation, the better it is. However, human evaluations of each and every output for the image to text model would be extensive but expensive for the companies creating the product. Therefore, BLEUs score could be applied to evaluate the accuracy of the model. BLEU Score is an evaluation metric that is quick,inexpensive, and language-independent,that correlates highly with human evaluation. It was first tested on machine translation tasks to determine its accuracy. The encoder-decor architecture implemented was evaluated to determine its performance on image to text conversion for the testing data.

In order to determine the performance of the text classification model, classification-related metrics were used to evaluate the accuracy related to the classification of the breakdown labels. Here, the accuracy is calculated by comparing the output of the detector to that of the actual labels that were humanely-provided. The following metrics were used:

- Accuracy: The number of correctly classified labels divided by the total number of labels to be classified. It gives a measure about how well the classification model is working on the given set of data.

- Loss: Machines learn by means of a loss function. It's a method of evaluating how well specific algorithms model the given data. If predictions deviate too much from actual results, loss function would cough up a very large number. Lower the loss, better the model.

The model architecture and the hyper-parameters of the deep learning model was adjusted in such a way that the accuracy is the highest and the loss function is the lowest for a given set of training data. Validation data is used to tune these hyperparameters. Weights and biases were saved of the trained model and applied to the model predicting the test data. Actual performance of the model is determined through its performance on the test set which is the data not considered in the training phase of the model.

5.2. Evaluate the business performance of the model

Developing a model with high accuracy is just not enough for business applications. Models developed should have good performance based on the business requirements of the project. Evaluating the model only based on the model metrics performance does not hold good for business applications of data science. Analyzing the output generated by the computer models and trying to integrate and improve the current logic in order to generate better results.

The given output from the image to text model was used to create sentences in text format. Importance of converting it to text format is that the text sentences would make it easier for machine codes to find relevant texts. In order to find the relevant texts, the generated sentences were mapped on 30 different POC business field labels to provide the ease of searchability that would decrease time as well as money for the businesses.

5.3 Recommendations based on the model's performance on model metrics and business data

As the number of data increases for training, the performance of the model increases significantly. However, while creating a data science product it should be noted that enough data should be available so as to handle all of the exceptions that could be faced by the product. This could help in significantly improving the model performance by integrating exceptions in the model. The integration of business rules to generate sentences from the image to text code becomes quite vital for the development of data science products in business organizations. Different types of tables affected the performance of the document processing task which could be improved through adding more data having different types of tables.

References

1. Nemec, Kelley D., "The Application of Paperless Processes to Improve Data Management within Small to Medium Businesses", University Honors Program Theses, 2019. (https://digitalcommons.georgiasouthern.edu/honors-theses/438 visited on 05 April 2021).
2. Hind Zantout, Marir Farh, "Document management systems from current capabilities towards intelligent information retrieval: an overview", International Journal of Information Management, 1999. (https://www.sciencedirect.com/science/article/abs/pii/S0268401299000432 visited on 05 April 2021).
3. Sonka Milan, et. al., "Image pre-processing", *Image Processing, Analysis and Machine Vision,* 1993. (https://link.springer.com/chapter/10.1007/978-1-4899-3216-7_4 visited on 08 April 2021)
4. Singaravela Sundaravelpandian, et. al., "Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction", *Advanced Engineering Informatics*, 2018. (https://www.sciencedirect.com/science/article/pii/S1474034617305359#:~:text=Results%20indicate%20that%20deep%20learning,component%20development%20process%20more%20efficient visited on 08 April 2021).
5. Jochen Hartmann, et. al., "Comparing automated text classification methods", *International Journal of Research in Marketing*, 2019. (https://www.sciencedirect.com/science/article/pii/S0167811618300545 visited on 08 April 2021).

6. Kishore Papineni, et. al., “BLEU: a Method for Automatic Evaluation of Machine Translation”, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2002. (https://www.researchgate.net/publication/2588204_BLEU_a_Method_for_Automatic_Evaluation_of_Machine_Translation visited on 09 April 2021).

7. Soni Neha, et. al., “Impact of Artificial Intelligence on Businesses: from Research, Innovation, Market Deployment to Future Shifts in Business Models”, *arXiv*. (https://arxiv.org/ftp/arxiv/papers/1905/1905.02092.pdf visited on 09 April 2021).

8. https://www.automationanywhere.com/company/blog/product-insights/what-is-intelligent-document-processing-a-primer#:~:text=Intelligent%20document%20processing%20includes%20four,extraction%2C%20and%20post%2Dprocessing visited on 04 April 2021.

9. https://celaton.com/news/intelligentdocumentprocessing visited on 04 April 2021.

10. https://blog.adobe.com/en/publish/2019/10/22/state-of-ai-in-document-management.html#gs.kgu4z5 visited on 06 April 2021.

11. https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/convert-word-to-vector#:~:text=Converting%20words%20to%20vectors%2C%20or,similar%20meanings%20have%20similar%20representations.visited on 06 April 2021.

12. https://info.basicsafe.us/safety-management/blog/how-much-are-paper-records-costing-your-company visited on 06 April 2021.