**Developing a Multi-Output Deep Learning Algorithm for Sentiment Analysis and Categorization for Enhancing Brand Recognition**

**Design and Methodology**

**Research Design**This study uses a thorough research approach, seamlessly integrating both qualitative and quantitative methodologies to explore the effectiveness of data analytics techniques and neural network structures in augmenting brand recognition. The focus of the study revolves around the Amazon Electronics Dataset, which contains a wide range of customer reviews and comments, in textual form.

The decision to choose the Amazon Electronics Dataset was based on two important factors,  
Firstly we were drawn to the dataset's size and diverse nature, which provided a great opportunity to delve into the customer’s opinions and categorizing different topics. With thousands of reviews covering a range of electronic products, it served as a solid foundation for conducting comprehensive analyses.

Secondly, by focusing on reviews within the Amazon Electronics domain we can gain insights, into customer preferences, product performance, and brand sentiment in a highly relevant context that mirrors the competitive electronics industry.

**Qualitative Component: Expert Interviews**

The study will engage in conversations with experts who have hands-on experience using advanced data analysis techniques to enhance brand recognition specifically within the Machine Learning context. These experts were selected based on their backgrounds and expertise to ensure a comprehensive understanding.

During these interviews, there were structured discussions focusing on the aspects of employing data-driven methods, algorithms, and machine learning tools to improve brand visibility. The project also delves into the utilization of Deep Learning techniques that enable computers to comprehend data as well as sentiment analysis methods and models for organizing text into categories.

The goal is to gather insights from these discussions about how experts rely on model development, preprocessing the data, how they think it can be further developed, and how they tackle challenges such as data noise reduction for analysis and managing imbalanced data distributions. This perspective will provide an understanding of how data analytics and machine learning are put into practice.

This qualitative data will serve as a foundation, for our subsequent quantitative efforts. This wisdom was applied later in the algorithm development process. By incorporating these observations alongside thorough analysis, it enhanced the ability to develop a Multi Output Deep Learning Algorithm (MODLA) that is not just technically robust but also highly applicable, in real world scenarios. *(MIMMO, UCL, Ferianc & Rodrigues, M. 2021)*  
  
  
**Quantitative Component: Multi-Output Deep Learning Algorithm (MODLA)**

The first part of this study focuses on developing a Multi Output Deep Learning Algorithm (MODLA) specifically designed for datasets like Amazon Electronics Dataset. This algorithm is a state of the art tool that can handle two tasks simultaneously; analyzing sentiment and categorizing text. By processing the collection of customer reviews and comments in the dataset the MODLAs neural network architecture expertly identifies complex patterns and uncovers hidden relationships in the text.

Powered by deep learning techniques the MODLAs neural network is trained using a diverse range of textual data. This training process involves refining its internal parameters through iterative analysis and adjustment based on the dataset, this is similar to how humans improve their skills over time *(Deep Learning, Ian Goodfellow, MIT Press, 2016*). As a result, the MODLA becomes skilled, at recognizing both positive and negative sentiment orientations expressed in reviews. It also gains the ability to classify reviews into categories contributing to a comprehensive understanding of their content. And this will be happening simultaneously at the same time.

The MODLAs effectiveness lies in its ability to identify linguistic patterns that traditional methods may overlook. The MODLA excels at handling amounts of data by intelligently distinguishing between different categories and tracking trends in sentiment. *(MIMMO, UCL, Ferianc & Rodrigues, M. 2021)*  By utilizing neural network architecture and leveraging machine learning techniques the goal is to convert raw textual data into valuable insights and meaningful categorizations.

Ultimately the MODLA goes beyond the limitations of traditional sentiment analysis models and basic classifiers. Its ability to perform tasks provides a comprehensive understanding of textual data that closely resembles human comprehension. *(Deep Learning, LeCun, Y., Bengio, Y., & Hinton, G., Nature, 2015).*  This enables decision making aimed at enhancing brand recognition.

**Data Collection**

**Qualitative Data Collection: Expert Interviews**

During the data collection phase individuals are carefully selected who have a proven track record in the domain of Data analytics and Machine Learning. Through structured interviews the aim was to explore the strategies in detail as well as the challenges they have faced and the outcomes they have achieved. The interview was conducted in a systematic manner allowing for an in-depth exploration of various aspects that arise from real world applications.

The qualitative insights gained from these interviews formed a foundation for the subsequent phases of the study. Through analysis the interview transcribe was thoroughly looked into. The process involved identifying recurring patterns, thematic clusters and intricate relationships, within the collected data. This analysis extracted themes that contributed to the comprehensive analysis.

**Quantitative Data Collection: Preprocessing the Amazon Electronics Dataset**

This investigation heavily relyed on an diverse dataset known as the Amazon Electronics Dataset. This dataset contains a range of valuable information in the form of customer reviews and comments. However before using this data to train the learning model it needs to be carefully preprocessed. During this phase, a variety of tasks were conducted to ensure the quality and appropriateness of the data for training purposes. Meticulous elimination of information or noise from the dataset is carried out, *(Springer, 2014)* ensuring the utilization of solely pertinent data. It is also essential to standardize formats across the dataset to enable integration and analysis. Furthermore, data cleansing is undertaken to eradicate any inconsistencies or anomalies that could impact the accuracy of model training.

This processed dataset serves as a representation of customers opinions expressed through their reviews and comments. It forms the foundation for training the Multi Output Deep Learning Algorithm (MODLA). Through the utilization of learning techniques on this dataset, valuable insights were unearthed regarding sentiments and categorizations inherent in textual content.  
  
**Data Collection - Scraping the Data**

The initial phase of collecting data is crucial for implementing the project as it provides the raw material for analysis and developing models. At first the plan was to scrape data from the Amazon platform expecting it to offer insights. However due to excessive bot sniping there were restrictions in place against data scrapping.

In response to these scraping restrictions imposed by platforms the project took a flexible approach. This pursuit of alternatives led to collecting the dataset from Stanford University, which granted access to the Amazon Electronics dataset.

The dataset acquired through this collaboration formed a part of the research endeavor. It not only provides a substantial amount of data for analysis but also highlights the ability to adapt in the face of unforeseen challenges. This experience underscores the real world obstacles often encountered in projects and emphasizes the importance of resilience and creative problem solving.  
  
**Data Preprocessing**

The success of the following analysis depends on how it is preprocessed and how the textual data have been collected *(Data Preprocessing in Data Mining, Batista, G. E., Prati, R. C., & Monard, M. C., Springer, 2014)* This crucial step ensures that the input data is prepared in a way that's suitable, for training the Multi Output Deep Learning Algorithm (MODLA). Since the textual data comes from sources it is important to have a careful and detailed preprocessing strategy to extract valuable insights. In this project the initial preparation of data forms the basis for training the Multi Output Deep Learning Algorithm (MODLA) on the Amazon Electronics Dataset. Each step of preparation, which is explained in the following sections has a purpose; text tokenization breaks down textual data into smaller elements removing stop words cleanses the dataset and encoding enables numerical processing. By performing these steps MODLA can extract valuable insights from the dataset both qualitatively and quantitatively. This phase plays a role, in our research (Géron, A., 2017).  
  
**Text Tokenization**

The first step in analyzing data involves tokenization, which is a crucial process that breaks down the text into individual tokens or words. By segmenting the text in this way tokenization sets the groundwork for analysis *(Géron, A., 2017)* This detailed representation allows MODLA to understand the connections, between words and phrases making it easier to perform accurate sentiment analysis and categorization.  
  
**Stop-Word Removal**

One important step in preprocessing is getting rid of stop words. These are words, like "the " "is,". And" that don't have much meaning on their own. Removing these words helps clean up the dataset so that the MODLA can concentrate on the words that truly express sentiment and determine categories. This improvement makes the algorithm better at recognizing patterns.

**Encoding and Vectorization**

Converting the text into numerical values marks the final step in the data preprocessing process. Methods such as one-hot encoding and word embedding will be employed to represent words as vectors within multi-dimensional spaces. This conversion allows the MODLA system to effectively handle information leading to reliable results, in sentiment analysis and categorization tasks. *Géron, A. (2017)*

Essentially the data preprocessing stage plays a role in preparing the raw text data for the MODLA. It involves techniques such as tokenization removing unnecessary words, stemming and encoding. These techniques help ensure that the research can extract insights, from the textual dataset both qualitatively and quantitatively.

**Model Architecture**

The basis of the Multi Output Deep Learning Algorithm (MODLA) rests on a designed structure that can handle both sentiment analysis and categorization tasks simultaneously. This technical framework smoothly integrates cutting edge machine learning methods to effectively handle the nature of textual data. At its core MODLA uses embedding layers to give meaning to text allowing for better understanding of word relationships and context. By incorporating layers MODLA excels at recognizing complex patterns in text sequences, which helps in interpreting emotions and assigning appropriate categories. The dense layers in its architecture further enhance MODLAs ability to uncover characteristics and subtle details, for categorization purposes. This comprehensive approach enables MODLA to effectively navigate the complexities of data resulting in accurate sentiment analysis and categorization outcomes. (Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning)

**Embedding Layers**

The architecture starts by using embedding layers which're crucial for adding meaning to the text. Through the process of embedding, words and phrases are transformed into vectors in vector spaces. This allows the algorithm to understand the relationships between words and their context. This important step creates a foundation for tasks, like sentiment analysis and categorization.

**Convolutional Layers**

Incorporating convolutional layers significantly improves the MODLAs capability to discover intricate patterns in sequences of text. These layers excel at recognizing small scale characteristics and hierarchies present, in the data. By capturing structures and connections convolutional layers enhance the MODLAs ability to understand emotions and assign relevant categories to various types of text inputs.

**Dense Layers**

The architecture reaches its peak with layers that take advantage of the information gathered by previous layers. These layers are excellent at performing calculations allowing the MODLA to discover more advanced characteristics and reveal hidden emotions and subtle categorization details. The interaction between layers, in the neural network enables thorough sentiment analysis and precise categorization.

**Evaluation**

**Qualitative Data Analysis: Unveiling Insights from ML and Data Analytics Experts**

Concluding the data collection phase, the journey of thematic analysis was initiated, engaging with insights from experts in Machine Learning (ML) and Data Analytics. *(Smith, J., & Brown, A., 2020)* This approach involved examining the content of interviews to uncover not only patterns but also profound insights rooted in ML and Data Analytics expertise. This analytical journey closely mirrors how ML algorithms are trained—revealing trends and shedding light on valuable themes through data-driven methods.

During transcription the collected data undergoes analysis. Like ML frameworks thematic analysis delves deep into expert narratives moving beyond surface interpretations to explore the intricate layers of practical strategies, challenges and outcomes encountered by experts as they utilize ML and Data Analytics for brand recognition.

Thematic analysis functions as a model similar to ML frameworks by extracting knowledge, from qualitative data in a systematic manner. The synthesized themes encompass the multifaceted dimensions of harnessing ML and Data Analytics techniques providing an understanding of the complex nuances that drive effective brand recognition strategies within the realm of ML and Data Analytics expertise.

**Quantitative Performance Metrics: MODLA Assessment**

To evaluate the Multi Output Deep Learning Algorithm (MODLA), attention is directed towards two aspects: sentiment analysis and categorization tasks. A set of metrics, including accuracy, precision, recall, and F1 score, is employed to evaluate the algorithm's performance.

In the field of machine learning and data analysis there are important performance metrics that help us evaluate how well models and algorithms work. Precision, Recall and the F1 Score are metrics in this regard. Precision measures how accurate positive predictions are by comparing them to the positive predictions helping prevent false positives. Recall, also known as sensitivity shows how well the model can correctly identify all instances among the actual positive instances. The F1 Score combines precision. Recall to provide a balanced assessment of both metrics. Additionally loss functions are crucial for assessing the performance of machine learning models by quantifying the difference, between predicted values and actual values. These metrics act as guides in navigating the world of model evaluation helping practitioners fine tune their algorithms for optimal results. *(Provost, F., & Fawcett, T., 2013)*

For categorization tasks, these metrics are also utilized to assess the effectiveness of MODLA in classifying data into predefined categories. A high accuracy score along, with precision, recall and F1 scores indicates that the algorithm effectively categorizes input data.

By employing an evaluation process that integrates qualitative insights and quantitative assessments, the effectiveness and accuracy of the developed MODLA in concurrently conducting sentiment analysis and categorization tasks can be gauged.

**IMPLEMENTATION**

**Introduction**

The implementation phase is a step, in the project process where the transition is made from ideas to putting them into practice. During this stage the projects attention shifts towards aspects such as gathering and preparing data analyzing it and developing models. This phase acts as a connection, between the framework established in stages and the tangible results that represent the projects objectives. *(James et al.2013)*

Following the planning phase and attaining a robust comprehension of the challenge at hand, the implementation stage involves engaging directly with real-world data and cutting-edge technology to tackle the task at hand. Our focus here encompasses two aspects; hands on coding and model development on one hand and conducting interviews with individuals, on the other hand. Both aspects are crucial and play vital roles in ensuring the successful completion of our project.

Regarding coding, the initial step involves the collection of data. Web scraping techniques were employed to gather information from online sources. During the stage of data collection efforts were made to gather information from the Amazon Electronics platform. However was faced with challenges along the way such as scraping limitations imposed by Amazon to detect bots and manage high traffic. To overcome this obstacle, contact was made with Stanford University, which possessed a comprehensive dataset on Amazon Electronics. This strategic decision not only allowed to obtain the necessary data smoothly but also showcased our ability to adapt in the face of unforeseen hurdles.

After collecting the dataset, the next steps were focused on preprocessing and processing. This phase involves cleaning and transforming the data well as conducting exploratory analysis to uncover insights and patterns. To get an idea about the dataset we are working on and to know more about relation with different markers operations such as basic sentiment analysis, classification and more advanced techniques, like topic modeling and emotion analysis, Time series analysis was done. These serve as components of the projects foundation and contribute greatly to its overall success.

However coding alone cannot fully grasp the understanding of user experiences and needs. That's where the interview process comes in. Interviews allows to engage with individuals who have knowledge of using data analytics techniques to improve brand recognition on social media. By conducting interviews insights that complement the quantitative results obtained through coding are found. These interviews provide context, personal stories and real world challenges that significantly enhance the understanding and decision making. *(Rapley, 2004)*

The combination of code driven outcomes and interview based insights is crucial in presenting an overview of the project. Integrating technology with experiences brings depth and subtlety to the findings. Together these elements result in a rounded implementation that showcases the technical skills, adaptability and empathy towards the target users.

The implementation stage reflects the maturity and progress of the project. As the project navigate through the code and conversations, it not only address the technical aspects but also emphasize the human centered dimensions that give meaning and impact to the work. It was made sure that the project aligns with both data driven insights and real world experiences.

**Reading JSON File**

We got the dataset in JSON format, A JSON file, also known as JavaScript Object Notation is a format used to store and share structured data. It is an readable data interchange format that can be easily understood by both machines and people *(Johnson, 2019)*.The project starts by reading the Amazon Electronics dataset from a JSON file format. JSON is selected because it can represent hierarchical data structures in a way that is easily understood, making it perfect for datasets of different levels of complexity.

After parsing the JSON file the data is processed in a manner allowing for a thorough understanding of its contents. This step reveals how the data fields are organized, their relationships to each overall composition of the dataset. Having this understanding sets the foundation, for making decisions during subsequent preprocessing stages.

**Converting it into an Excel File**

By converting the data into an Excel format several advantages are realized. The tabular structure of Excel improves clarity and simplifies the representation of information. It helps in identifying trends, anomalies, and significant patterns within the data *(Berk, K.N., & Carey, P. 2019). Data Analysis with Microsoft Excel*l) Moreover, Excel's user-friendly interface makes it accessible to team members who may not have specialized skills promoting collaborative decision making.  
  
The project recognizes the significance of having structured data for effective analysis. To achieve this, the dataset is converted from JSON into an organized Excel spreadsheet. This conversion process utilizes the pandas library, which is a powerful tool in Python for manipulating data.

Preprocessing plays a crucial role in this project as it involves reading the JSON file and converting it into an Excel spreadsheet. This step establishes a foundation for subsequent stages, like exploratory data analysis and model development. It highlights how adaptable our project is when dealing with data formats to extract meaningful insights effectively.

**Statistics**

During the phase of this project's data exploration journey, to uncover the fundamental characteristics of the dataset a thorough and organized analysis was conducted. This analysis involves examining aspects that provide valuable insights, for the rest of the project.

As the data-driven approach begins, a view of how the dataset is structured was obtained. This important factor gives us an idea of its composition, including the number of rows and columns that define its shape. By understanding the structure of the dataset an understanding of its size and scope was gained.  
  
Moving forward the attention shifted towards examining the information contained within the dataset. The focus was on understanding the metadata, which revealed details about the types of data in each column and whether there were any missing values. By exploring these data attributes insights were gained into the nature of the dataset, which formed a strong foundation for further analysis.

The project moved on to summary statistics for further exploration. These key metrics offered a view of how the data was distributed and its central tendencies. Descriptive statistics played a role in summarizing attributes by providing measures, like mean, median and dispersion. These statistics helped to gain an understanding of how the data behaved, guiding the subsequent analysis and aiding in decision making.

In the evolving journey of exploring data the initial phase of diving into information and statistics went beyond being just a procedural requirement.This exploration served as a guiding compass for the project, mapping out the path for pursuits and forming the core of the investigation.

In total there were 1,035,845 null values for the 'overall' attribute and 887,548 non null entries for 'verified'. The 'ReviewTime' had 1,042,266 null values while 'reviewerID' contained a complete set of 1,048,575 entries. As for the 'asin' it had 1,047,571 null values. Moving on to the 'style' attribute which featured 902,959 null entries and the 'reviewerName' with 1,013,386 non null values. The 'reviewText' had a total of 1,046,804 entries. Additionally,'summary' included a count of 1,048,484 null values whereas 'unixReviewTime' had a count of 799614 entries. The attribute called 'vote' exhibited a count of 275785 values. Finally the attribute named 'image' was represented by a count of 72,538.

Moreover the summary statistics provided insights into how the data is distributed. It revealed attributes with unique values,frequencies and other important metrics.This in depth analysis helped establish an understanding about the datasets characteristics which guided further analysis.Within these statistics it was found that there were exactly;1035845 occurrences for the ‘overall’ attribute; 887548 occurrences for ‘verified’; 1042266 occurrences for ‘reviewTime’; 1048575 occurrences, for ‘reviewerID’ and 1047571 occurrences for ‘asin’. The ‘style’ attribute featured 902959 unique entries while there were 669843 unique values for ‘reviewerName’. There were 836,949 entries in the 'ReviewText' section and 447,538 unique values, in the 'summary' section. The 'UnixReviewTime' had 72,620 values and 'vote' had 14,412 different entries. Lastly the 'image' category was represented by 9,546 values.

Upon analyzing the metadata it was discovered that the dataset contains both numerical and categorical data. For instance columns like 'overall'. Verified' are categorical in nature representing factors such as product ratings and verification status are numerical. On the other hand columns like 'reviewTime' and 'unixReviewTime' are numerical indicating timestamps. Recognizing these data types is crucial as it helps determine which analytical techniques are most appropriate.

Furthermore examining the metadata helped to identify columns with values. For instance the 'vote' and 'image' columns have a number of missing values. Understanding the extent of this missing data assists in making decisions regarding data imputation or whether to exclude these columns from analyses.

By examining summary statistics insights were gained on how the statistics were distributed. These statistics included details, about the unique values, frequencies and other important metrics. They helped to grasp the characteristics of the dataset effectively.

**Dropping Unnecessary Columns**

To ensure the accuracy of the data we carefully removed columns such as 'reviewerName' 'vote' and 'image' from the dataset. This trimming allows us to maintain focus, on our core objectives during analysis and model development eliminating any distractions. By selecting these columns we demonstrate our dedication to a streamlined and effective analysis process.

. **Sentiment Analysis using NLTK**As a first step towards future advancements of the project we implemented basic sentiment analysis using NLTK so that we can label the huge amount of data we have later on to develop the neural network and to get a basic idea about the sentiment distribution of the dataset. (Pang and Lee, 2008) We utilized the Sentiment Analyzer from the nltk library to assess sentiment scores providing an understanding of the underlying polarity in each review.

It's important to highlight that this initial exploration into sentiment analysis and categorization not enhances our current analytical insights but also establishes a foundation for a more advanced multi output neural network in subsequent stages. This strategic progression demonstrates our approach ensuring that each phase serves as a building block, towards delivering a comprehensive and high impact solution.

The project used NLTKs Sentiment Intensity Analyzer, which integrates the VADER lexicon created specifically for analyzing social media text. In this code there is a custom function called 'get\_sentiment' that categorizes sentiment scores as 'Positive' 'Negative,' or 'Neutral.' Every text review in the DataFrames 'cleaned\_review' column goes through sentiment analysis. A sentiment score is calculated using VADERs compound score. These scores are then stored in a column called 'sentiment\_score.' Afterward we utilize the 'get\_sentiment' function to classify these scores into sentiments, like 'Positive' 'Negative,' or 'Neutral,'. The results are saved in a column named 'sentiment.'

The provided sentiment metrics include Accuracy (43%) Precision (51.4) Recall (53.2) F1 Score (51.8) and Loss (17.10%). These metrics evaluate the performance of the sentiment analysis model giving insights into its accuracy, precision, recall and error rate in classifying sentiments

**Categorization Model using Classifier:**When the project was started, one of our goals was to develop a neural network that can predict both sentiment and category labels simultaneously. To make this complex model possible focus was on creating a text categorization system. This initial model played a role, in categorizing textual data into three distinct and meaningful classes; "Care," "Leads," and "General." a keyword based approach was used for this categorization, where carefully curated lists of keywords to each class were utilized. For example the "Care" category included words like "complaint " "problem," and "disappointed," while the "Leads" category encompassed terms such as "buy," "purchase," and "product." The default option was the "category when no specific keywords, from the classes were identified.

There were two reasons, for creating this categorization model. Firstly it was necessary for labeling the dataset, which's essential for training the neural networks that follow. And more importantly it helped to simultaneously work on developing the multi output neural network by providing real time labeled data. Additionally this categorization provided insights into the distribution of content in the dataset, which was crucial, for guiding stages of our project. (Sebastiani, 2002)

The initial categorization model showed outcomes providing a foundation, for the overall goals. The model achieved an accuracy of 62% with a precision of 60.1% a recall rate of 62% and an F1 score of 0.587. Additionally the log loss, which is a metric for evaluating the models performance was recorded at 9.13. These findings highlight the possibility of text categorization. Offer insights, into the underlying structure of the dataset further motivating us to continue developing the multi output neural network. This network will build upon these categorizations to predict both sentiment and category labels.

**Neural Networks for Sentiment and Category Classification:**

One of the stages, in this project is to create neural networks for sentiment and category classification. Developing these networks is a step towards building a output neural network that can predict both sentiment and category labels simultaneously.  
  
These individual networks serve as building blocks forming the foundational structure for the comprehensive neural network with multiple outputs. By constructing them in this way we create specialized models for analyzing sentiment and categorizing data. This ensures that each component of the output network is finely tuned to its specific task resulting in improved accuracy when they are integrated together. Additionally taking a step by step approach during development allows to proactively identify and address challenges early on reducing the risk of unexpected issues during integration. It also helps to make progress in data preprocessing, feature engineering and model optimization.*(Kim, Y. 2014)*. Furthermore these separate networks enhance our understanding of the dataset by examining sentiment and category analysis providing valuable insights that contribute to effectively handling the complexities of the dataset.

**a)** **Sentiment Analysis Neural Network:**

The neural network has been designed to perform the sentiment analysis. By utilizing the TensorFlow library the project delve into the computational sentiment analysis. The process starts with tokenization breaking down the dataset into tokens setting the stage for linguistic analysis. Embedding layers then convert these tokens into vectors enabling the network to comprehend word relationships like humans do.The dataset labeled by the NLTK sentiment model was used for the neural network training.

The true strength of this network becomes evident as it explores connected layers unraveling complex patterns, *(Kim, Y. 2014)* within the dataset to effectively understand underlying sentiments. Training involves optimization through exposing the model to the refining internal parameters resulting in more accurate interpretation of sentiments. It is crucial to preserve this trained model to ensure its value for phases and highlight the dedication towards integrating advanced technology.

**b)**  **Category Classification Neural Network:**

In the exploration of machine learning techniques, the project delved into developing a neural network specifically designed for categorizing different types. Like its predecessor in sentiment analysis this project utilized the powerful TensorFlow library, which is widely recognized in the field of deep learning. The dataset labeled by the classification model was used for the neural network training.

The process of constructing this network closely followed the steps happened in the sentiment analysis. It started with tokenization, where elements of the dataset was converted into tokens. This intricate transformation formed the basis for conducting analyses, where words were embedded within multidimensional layers. By doing the neural network gained the ability to understand contextual relationships between words.

The strength of this network became evident as it moved through a series of interconnected layers. These complex layers worked together to uncover patterns and connections within the dataset resulting in an impressive ability to accurately categorize reviews. This process resembled sentiment analysis, also involved fine tuning the model’s internal parameters based on the analysis.

In this stage of our project it was intentionally chosen not to calculate traditional accuracy metrics for both the sentiment and category neural networks. There are two reasons behind this decision. Firstly these neural networks serve as steps towards building a multi output neural network that can make simultaneous sentiment and category predictions. Secondly the focus during this phase is on training optimizing and validating the networks to ensure their effectiveness and reliability in later stages.

Traditional accuracy calculations will be performed in stages when the multi output neural network is fully established. This will allow to conduct a comprehensive and contextually relevant evaluation of the entire system.

**Exploratory Insights: Time Series Analysis, Named Entity Recognition (NER), Emotion Analysis**

In the pursuit of constructing a neural network that can predict sentiment and category labels simultaneously, The project embarked on a journey filled with important detours each contributing in its own unique way. The project also delved into Named Entity Recognition (NER) a task in natural language processing (NLP) which provided with valuable insights into the underlying structure of the dataset. By identifying and categorizing named entities like individuals organizations and locations NER added an extra layer to the data analysis helping to understand the context and relationships within the text. (Nadeau, 2017)

Time Series Analysis, a crucial step on the journey helped to explore the patterns of sentiment and category changes over time. *(Shumway,2017)* This analytical approach shed light on how the dataset evolves, revealing trends and variations in sentiments and categories. Although Time Series Analysis didn't directly contribute to the multi output neural network it provided vital insights into the dynamic nature of sentiments and categories.

Emotion Analysis, is when analysis delve into the emotional context of the textual data. It goes beyond sentiment, it will help to understand the emotion behind the text, The project explore emotion analysis as well to further know the possibility of ML techniques. *(Ekman,1971)*

The detours took through NER, Time Series Analysis and Emotion Analysis and they were carefully executed. They added depth to the dataset, it gave some background information and armed us with knowledge that will ultimately improve the accuracy and flexibility of the multi output neural network. These exploratory actions though different from each other are components towards building a comprehensive and efficient predictive model.

**Time Series Analysis**

Time Series Analysis played a role in uncovering how sentiment trends change over time. This investigation explored the relationship between evolving sentiments and the temporal dimension *(Shumway,2017)* By aggregating sentiment scores within specific time periods our project aimed to identify meaningful patterns, fluctuations and shifts in sentiments.

To lay the foundation for this analysis, the project utilized the sentiment scores assigned to each review earlier in the project. These scores were then organized chronologically to create a dataset that represents a timeline. The Python code grouped these sentiment scores into time intervals, such as days, weeks or months based on our desired level of detail.

Next statistical and computational techniques were applied to these sentiment scores within each time interval. The goal was to extract aggregated metrics like the average sentiment score for each period providing a representation of changing sentiment tendencies.

However numerical insights alone couldn't fully capture the variations of sentiments over time. To address this limitation data visualization techniques were used. Using Python libraries, like Matplotlib and Seaborn line plots that visually depicted how sentiments rise and fall over time was introduced. Peaks, valleys, periods of stability and sudden changes in sentiment were all clearly depicted through these visuals.

**Named Entity Recognition (NER)**

Named Entity Recognition commonly referred to as NER is a technique, in the field of Natural Language Processing (NLP) that aims to identify and categorize named entities in text. These named entities can be classified into predefined categories such, as peoples names organizations, locations, expressions of time quantities, monetary values, percentages and more *(Nadeau, 2017)*  
  
 Named Entity Recognition (NER) plays a role in this project. Spacy library was used to extract entities from carefully processed review texts. This phase goes beyond sentiment analysis and delves into a domain where the texts essential elements, such as names of people, landmarks, time references and more are thoroughly understood.

The code implements Spacys NER module to examine the review texts. Through its capabilities the Spacy library can identify and categorize various types of named entities found in the text. These entities can range from names, to geographical locations organization names and chronological references.  
During the analysis of named entities we identified themes present in the reviews. These themes are represented by repeated keywords that indicate topics. Categorizing the reviews based on these themes helps us better understand the sentiments expressed by customers and gain insights into their experiences and opinions, about aspects of electronic products. Here is how the topics are distributed;

* Topic: Positive Sentiments and Product Attributes (e.g., "perfect," "love," "great")
* Topic: Product Excellence and Appreciation (e.g., "excellent," "product," "great")
* Topic: Ease of Use and Functionality (e.g., "use," "easy," "camera")
* Topic: Quality and Performance (e.g., "quality," "good," "great")
* Topic: Ratings and Customer Experience (e.g., "stars," "great," "product")
* Topic: Performance and Value (e.g., "works," "value," "speakers")
* Topic: Satisfaction and Meeting Expectations (e.g., "good," "worked," "expected")

**Emotion Analysis and Sentiment-Based Emotion Derivation**

Emotion analysis, which is also referred to as emotion detection is a technique used in natural language processing (NLP) to identify and examine the tone, sentiment or subjective information conveyed within data *(Ekman,1971)*

Using the sentiment scores obtained from the analysis of sentiments we conducted an examination to uncover the underlying emotional nuances present in the reviews. These sentiment scores formed the basis for identifying emotions, like happiness, sadness, anger, surprise and neutrality. Each emotion was associated with thresholds of sentiment intensity. The analysis followed predetermined criteria as outlined below;  
  
***Joy:*** Reviews were associated with the emotion of joy if their sentiment score exceeded a threshold of 0.3. This classification denoted highly positive sentiments, capturing instances where customers expressed substantial satisfaction and elation with the product.

***Sadness:*** Reviews received the label of sadness when their sentiment score fell below -0.3. This demarcated profoundly negative sentiments, signifying instances where customers conveyed pronounced dissatisfaction and disappointment with the product.

***Surprise:*** Sentiment scores above 0 (but below 0.3) corresponded to the emotion of surprise. This encompassed moderately positive sentiments, suggesting customers' pleasant astonishment or unexpected satisfaction with the product.

***Anger:*** Sentiment scores below 0 (but above -0.3) were aligned with the emotion of anger. This encapsulated moderately negative sentiments, indicative of customers' discontent or frustration with the product.

***Neutral***: Reviews exhibiting a sentiment score precisely at 0 were attributed to the emotion of neutrality. This encompassed instances where sentiments were neither overtly positive nor negative, reflecting a balanced or unbiased view.

**Conclusion of Preliminary Analysis and Preprocessing:**

In this phase of the project all the necessary groundwork for the upcoming stages of research was completed. The journey began by collecting data specifically focusing on obtaining the Amazon Electronics dataset which's crucial for this project. Then the dataset is transferd into a structured Excel format to enable efficient analysis. During the exploratory data analysis unnecessary columns were removed to streamline the dataset.

To ensure clean textual data text preprocessing tasks were conducted to properly format it. By conducting sentiment analysis and categorization an understanding of the overall sentiment and catagory of the dataset was understood.

The advancement of networks in sentiment analysis and category classification has demonstrated a more thorough level of analysis. By utilizing TensorFlow, a deep learning framework we were able to train models for sentiment assessment and categorization. These models serve as tools for automatically assessing sentiment and categorizing in future stages.

Time series analysis was also explored to understand how sentiment changes over time. *(Shumway,2017)* Through visualizations representations were gained of the fluctuations in sentiment giving a unique perspective on the evolution of customer sentiment.

In addition named entities recognition was also explored to get an idea of the mentioned entities in reviews, which could be crucial for further analysis.

Lastly by combining sentiment scores with emotion analysis the project added an understanding of emotions conveyed in customer reviews beyond sentiment itself.

Overall these initial stages have laid a foundation for future advancements in the project. With a preprocessed dataset sophisticated models for sentiment assessment and categorization were trained.This will significantly enhance the depth and analytical power of our research endeavor.

**MULTI-OUTPUT NEURAL NETWORK IMPLEMENTATION**

Multi-Output Neural Network also referred to as Multi Task Neural Network is a type of neural network architecture specifically designed to predict multiple outputs or labels at the same time using a single input. In this network the final layer branches out into output layers each corresponding to a specific task or prediction. Each output layer can have its loss function which enables the network to optimize for multiple objectives concurrently.

Multi Output Neural Networks find utility in machine learning tasks where there is a requirement to predict multiple variables that are related or dependent, on each other using the same input data *(Caruana, R. 1997)*

**Importing Libraries and Reading Data**

The first phase of the project started by importing the necessary libraries and extracting relevant data from an Excel file. This important step sets the foundation for the following processes. By using these imported libraries the code creates a proficient environment for upcoming operations. The dataset consists of reviews, corresponding sentiments and previously predicted categories, which are crucial for the analysis.This strategic combination of code and data, at this point prepares for the stages of implementing a multi output neural network.

**Data Preparation**

Data preparation is a part of the implementation process *(Springer, 2014)* and great care was taken to ensure that it seamlessly integrates into our multi output neural network. During this phase, transformation of sentiments and categories was done which're key components into numerical values to lay the foundation for further analysis. To help the model understand the data better Keras Tokenizer class was used and it was effective for tokenization. Additionally reviews are uniformly padded to standardize their lengths and ensure correct input for the model. This meticulous data preparation guarantees that subsequent phases can extract insights, from the neural network model.

**Model Architecture**

The model architecture that has been selected is carefully crafted with attention to detail aiming to extract valuable information from the data while effectively dealing with the complexities arising from the multi output nature of the problem.

***Embedding Layer:*** Using an embedding layer in the beginning is important because it helps convert words into vectors allowing the model to understand the connections between words. This is crucial, for capturing the meanings embedded in the reviews. Since words that are related within a given context typically have embeddings, this layer ensures that the model can interpret the hidden contextual meaning.

***LSTM Layer:*** The LSTM layer, which comes after the embedding layer is highly effective in modeling data. This is especially crucial when dealing with text data because the arrangement of words carries meaning. LSTMs excel at capturing relationships that span across a sequence making them a reliable option for discovering complex emotions and patterns present, in reviews.  
  
***Dual Output Layers:*** The models structure is enhanced with two output layers, each designed to handle a specific prediction task; sentiment and category. This decision is based on the understanding that sentiment and category predictions present challenges each requiring its own specialized predictive layer. By separating them the model can learn patterns related to sentiment and category classifications improving its ability to offer precise and detailed predictions.  
  
***Rationale Behind Architecture:*** The decision to incorporate an architecture based on LSTM is well founded. LSTM layers are particularly effective, in handling sequences, which aligns seamlessly with the nature of text data. This capability ensures that the model can capture not immediate context but also long range dependencies leading to a more comprehensive understanding of the reviews. The design of having outputs acknowledges the multifaceted nature of the problem, where predicting sentiment and category require separate insights. This intentional design choice empowers the model to not learn intricate textual nuances but also differentiate between different categories ultimately enhancing overall predictive accuracy.  
  
To summarize, this architectural setup combines advanced methods in a carefully chosen manner to tackle the complexities of analyzing text data and predicting sentiment and categories. It optimizes the models capacity to derive insights, from the data resulting in more precise and insightful predictions.

**Validation: Ensuring Real-World Performance**

When it comes to developing a neural network that can handle multiple outputs the validation phase becomes a crucial checkpoint. During this phase we go beyond using the training data and assess how well the model performs in real world situations that it hasn't encountered before

To start this process an Excel file specifically for validation purposes was imported. These reviews act as a new dataset that the model hasn't seen during its training. This uniqueness is important because it provides us with a measure of how well the model can generalize *(Caruana, R. 1997)*

Just like during training we apply preprocessing steps to tokenize and pad the validation reviews to match the input requirements of the model. Once preprocessed the data the model predicts both the sentiment and category associated with each validation review.

However the validation process goes beyond automated measurements. To gain an understanding of the models effectiveness a manual validation procedure was conducted, carefully selecting 100 reviews from the validation set. Comparing the models predictions of sentiment and category with the actual values for each review was also done.  
 This manual validation approach provides insights into the practical accuracy and precision of the model capturing nuances that automated measures might miss.

The results of this validation was presented using a pivot table, which is an analytical tool that helps to calculate important metrics such as accuracy and precision. Accuracy tells us how many predictions for sentiments and categories are correct while precision gives us insight into how the model accurately classifies instances, within specific sentiments or categories.

The reason behind including validation is that it helps uncover subtle patterns and differences that automated methods might miss. Human evaluators have the ability to notice these nuances, which enriches the evaluation process by providing insights that contribute to a more comprehensive understanding of the models strengths and areas for improvement.

The validation phase serves as a link between development and practical application *(Ruder, S. 2017)* It confirms the models effectiveness, in real world situations. Ensures consistent performance. By combining automated metrics with validation using pivot table analysis a well rounded assessment of the models capabilities leading to informed insights and potential improvements can be found out.

**Summary: Multi-Output Neural Network**

The development of a neural network that can handle multiple outputs has been a crucial aspect of this project. By combining sentiment analysis and category classification and implementing them meticulously solution for complex issues were addressed.

The journey began with importing the data and activating libraries to extract insights from the dataset. To ensure consistency of the model, Keras Tokenizer class for tokenization and padding of reviews was used.

The architecture of the network helped to effectively uncover intricate patterns within the data. With the help of Keras functional API a model consisting of an embedding layer, a LSTM layer and two separate output layers for sentiment and category predictions was designed. The choice of 'adam' optimizer and 'sparse\_categorical\_crossentropy' loss function was driven by the commitment, to optimization and accuracy.

The validation phase demonstrates the models resilience by introducing a set of reviews. These reviews are carefully processed to predict sentiment and category. A method called pivot table analysis was used to measure accuracy and precision through manual validation ensuring a thorough evaluation of the models capabilities.

In this project the multi output neural network incorporates meticulous code implementation, strategic model architecture and comprehensive validation procedures. Its significance goes beyond being a technological achievement; it represents sophisticated solutions to real world challenges. As this chapter concludes, the network markers will provide valuable insights and improved efficiency.

**Result**

**Introduction to Results**

The journey into the world of Multi-Output neural network is an evolving story that seamlessly combines technical expertise with expert insights. Equipped with a dataset that includes 1,048,575 entries in various categories the expedition navigates through the vastness and complexity of this extensive data landscape. The initial chapters of the exploration reveal statistics that provide a comprehensive overview of the dataset encompassing crucial aspects such, as 'overall' 'verified,' 'summary,' 'unixReviewTime,' 'vote,' and 'image.'

With a thorough approach to refining data, The Project set out on a path of preparing the data paying close attention to address any missing values and shaping the dataset into a refined version that is well suited for analysis. The transformational journey involved steps such as converting text to lowercase removing punctuation marks, special characters, numbers and unnecessary stopwords. Leveraging the Natural Language Toolkit (NLTK) library the project embarked on sentiment analysis by skillfully categorizing reviews into 'Positive' 'Negative,' or 'Neutral' sentiments. This task serves as a foundation, for the subsequent analyses.  
  
The efforts in analyzing sentiments produced important outcomes, The project achieved a Sentiment Accuracy rate of 78.00% Sentiment Precision of 77.04% Sentiment Recall of 78.00% and a Sentiment F1 Score of 77.49%. However this was the beginning of the journey. The next phase involved classifying reviews into three categories; 'General,' 'Care' and 'Leads.' The Project accomplished this by using specific keywords as indicators. In this stage the project attained an Accuracy rate of 0.62 a Precision rate of 0.601 a Recall rate of 0.62 and an F1 Score of 0.587 which laid the groundwork for more, in depth analyses.

However the journey is not about achieving numbers. Its also copuled with captivating stories that shed light on the nuances of the dataset. From exploring 'Review Count vs. Sentiment' to analyzing 'Frequency vs. Words' and 'Count vs. Categories' these visual tales guide towards an understanding of the data.

In the quest to comprehend the data effectively the project delved into statistics that revealed the distribution of sentiment scores and the prevalence of different categories. The project uncovered the relationship, between sentiment and category through meticulous analysis bringing attention to their interconnectedness.

However the true masterpiece of the project lies within the output neural network. Going beyond its complexities it embodies the distilled knowledge we gained from interviewing industry experts. Engaging in conversations with figures like "Person A" from "Company A" and "Person B" from "Company A" provided with invaluable insights into implementing multi output neural networks. These discussions covered topics such as architectural choices designing loss functions, overcoming convergence challenges and ensuring scalability. These interactions greatly enriched the understanding.

As the project delves deeper into the core of this neural network it becomes apparent that the creation is not just, about data; it represents a herald of transformation that has the potential to reshape the landscape of the ML industry. Equipped with both proficiency and expert wisdom it stands poised to make a significant impact.

**Interview Findings**

Through the pursuit to comprehend the complexities of multi output neural networks and their real world significance we had enlightening discussions, with two professionals affiliated with "Company A." These individuals, namely "Person A" and "Person B " shared invaluable perspectives on the approaches, obstacles and effective techniques involved in implementing multi output neural networks.

**Strategies for Implementing Multi-Output Neural Networks**

In our conversation with "Person A " we discovered some strategies used in implementing multi output neural networks. We found two approaches; the "branched" approach and the "direct" approach. According to "Person A " the choice between these approaches depends on how tasks are related. In the architecture there are separate sub networks that extend from a shared initial layer. This is useful when tasks share features or connections. On the other hand the direct architecture predicts all outputs based on a shared representation, which is more suitable when tasks are unrelated to avoid any interference.

This strategic insight emphasizes the importance of customizing network architecture to fit tasks and their interdependencies. It shows how multi output neural networks can be adaptable and versatile, in tackling challenges.

**Designing Loss Functions for Multi-Output Neural Networks**

Designing loss functions for networks that handle both classification and regression tasks in the same model requires careful consideration. According to "Person A " it's important to approach this aspect with attention to detail. They explained that when it comes to classification tasks, categorical cross entropy loss is commonly preferred while squared error is widely used for regression tasks.

One major challenge arises when dealing with objectives of different scales. In cases it becomes crucial to balance these losses. This can be achieved by u normalizing the outputs. "Person A" also emphasized the usefulness of loss or class-weighted loss functions when working with imbalanced datasets.

Additionally combining losses, in output neural networks can be complex. "Person A" suggested assigning weights to each loss term based on their importance compared to others. This careful approach ensures optimization of multiple objectives by the network.

**Overcoming Convergence Challenges in Multi-Output Neural Networks**

Training networks with multiple outputs presents unique challenges as there can be conflicts between the different objectives. We were introduced to techniques by "Person A" that help ensure convergence in such scenarios. Additionally regularization methods like stopping, dropout and batch normalization contribute to the stability of the training process.

Another effective strategy is prioritization, where we focus on training the important outputs first to streamline the learning process. Transfer learning is also beneficial in output problems; it involves initializing the network with weights, from a related task and then fine tuning it specifically for our target outputs. By employing these strategies we can effectively achieve convergence in a multi output neural network.

**Conclusion**

**Appendix A**

**Interview With “Person A” from “Company A”**

* Hi “Person A” Can you please provide an explanation of the strategies used to implement multi output neural networks? Also, in what situations would one approach be more suitable, than the others?

**ans)** When working with output neural networks the design of the architecture relies on the specific tasks at hand and their interconnections. One common approach is known as the "architecture, where separate sub networks extend from a shared initial layer. Another option is the "architecture, where all outputs are predicted directly based on a shared representation. The decision between these two approaches largely depends on whether the tasks have any relationships or if they're completely independent. If the tasks are connected and share features opting for a branched architecture can prove efficient. On the hand if the tasks are unrelated a direct architecture may be more suitable to avoid any interference, between them.

* How can one go about designing or selecting loss functions for a neural network that has multiple outputs and handles both classification and regression tasks within the same model?

**ans)** It is important to carefully design loss functions in output networks. When dealing with classification tasks categorical cross entropy loss is often employed. On the other hand mean squared error is commonly used for regression tasks. In situations where the scales of the outputs differ it becomes necessary to balance these losses by either using weighted loss functions or normalizing the outputs. For datasets that have imbalanced data, focal loss or class weighted loss functions can be beneficial. Combining losses can be challenging; one approach is to assign weights to each loss term based on its relative importance compared to the others.

* How do you make sure that a neural network with multiple outputs converges, during training taking into account the difficulties that may arise when optimizing multiple objectives?

**ans)** Training a network with multiple outputs can pose challenges due to conflicting objectives. To address this we can employ techniques like task learning (MTL) that leverage shared representations. Regularization methods such as stopping, dropout and batch normalization are beneficial. When dealing with a number of outputs it's advisable to initially prioritize and focus on the most crucial ones during training. Another approach is transfer learning, where we initialize the network with trained weights from a related task and then fine tune it, for our specific multi output problem.

**Interview With “Person B” from “Company A”**

* When assessing the effectiveness of a output neural network how do you ensure that each outputs metrics are appropriately weighted, particularly when one output carries more significance, than the others?

**ans)** When assessing the effectiveness of a network that produces multiple outputs it's important to take into account the relative significance of each output. If one output holds importance than the others you may assign varying weights to the outputs while calculating an overall performance measure. For example you could compute an average of metrics like F1 score or precision recall. Alternatively you might utilize domain loss functions that give prominence to specific outputs. Additionally engaging in communication, with stakeholders is crucial to grasp their priorities and adapt the evaluation approach accordingly.

* Are there any particular transfer learning methods or pre trained models that you find helpful, for tasks involving outputs? If so how do you modify them to suit your problem domain?

**ans)** Transfer learning can bring advantages when dealing with tasks that have multiple outputs. Utilizing trained models such, as BERT or ALON which have been fine tuned on a wide range of data is often a reliable starting point. However to adapt these models to problem domains it is necessary to retrain the upper layers using your own dataset. The key challenge lies in selecting the layers for fine tuning and determining the optimal number of training examples required to avoid overfitting. Additionally customization of the output layers may be necessary to align with your desired target outputs.

* How can we guarantee that a multi output neural network is scalable and computationally efficient, in real world scenarios particularly when confronted with a quantity of outputs?

**ans)** It is vital to ensure that multi output neural networks are scalable and computationally efficient especially when dealing with a number of outputs. One approach is to employ model parallelism or distributed computing, which allows for training and deployment of models across multiple devices or clusters. Techniques such as model distillation, where a smaller network learns from an one can also enhance efficiency. Additionally leveraging hardware accelerators, like GPUs or TPUs can expedite both training and inference processes enabling the model to be used in time or high throughput applications.

**Interview With “Person C” from “Company B”**

* Person C, When developing a network that can predict sentiment and categories simultaneously how do you determine the optimal model structure?. Why is this decision important, for both the project and the coding process?

**ans)** Well selecting the appropriate model structure is a step. You must consider the connection between sentiment and category predictions. If they have a relationship to how the various branches of a tree share a common trunk it is efficient to have an initial layer that branches into sub networks. However if they are like separate trees, independent or "direct" architectures might be more suitable. This decision can significantly impact how the neural network operates and can either make or break the project. In coding terms it's akin, to determining the blueprint for a rocket before constructing it. If you get it right your journey will be smoother.

* How can incorporating data through pseudo-labeling be beneficial, for project development? What are the important steps or factors to consider when implementing this technique in coding tasks?

**ans)** Using pseudo labeling in your project is akin to giving it a boost. It helps broaden your dataset ultimately leading to precise predictions. However there's a catch – you need to exercise caution. Take the time to evaluate the quality of the data and only utilize it if you're reasonably confident, in its reliability. Additionally it's wise to establish a confidence threshold before applying pseudo labeling;

* Optimizing model performance through hyperparameter tuning is an aspect of code development. Could you please provide some insights on how to effectively carry out this tuning in the context of a project, like ours and how it can contribute to achieving our project goals?

**ans)** Tuning hyperparameters is so important in achieving the needed accuracy markers. You begin by exploring values for factors such as learning rates batch sizes and more. Cross validation becomes your ally; it's comparable to running simulations to determine the effective approach. And don't overlook evaluation metrics like precision and recall—they provide insight into whether you're hitting the target or veering off course. Keep tuning until you attain that optimal performance.

* The Project is planning on utilizing metrics to evaluate the performance of our model. Do you think these particular metrics are important, for the project and coding endeavors? Furthermore how do they assist in refining the output neural network?

**ans)** Absolutely. These metrics are similar to the dashboard instruments. They provide feedback on the performance of your neural network. Accuracy indicates whether you're on the track while precision and recall act as your navigational aids. They help you make decisions regarding the trade off between positives and false negatives which is particularly important for tasks like autonomous driving. About the F1 score, Well it's like finding the balance, between speed and safety; it's that ideal point you aim for. These metrics serve as your GPS on the journey of refining your network ensuring that it guides your project in the correct direction.

* Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning. MIT Press.
* Ferianc, M., & Rodrigues, M. (2021). MIMMO: Multi-Input Massive Multi-Output Neural Network. University College London
* LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444.
* Batista, G. E., Prati, R. C., & Monard, M. C. (2014). Data Preprocessing in Data Mining. Springer.
* Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow. O'Reilly Media
* Smith, J. A., & Osborn, M. (2008). Interpretative phenomenological analysis. In J. A. Smith (Ed.), Qualitative psychology: A practical guide to research methods (pp. 53-80). Sage Publications
* Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. O'Reilly Media, Inc.
* Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. Information Processing & Management, 45(4), 427-437.
* Davis, J., & Goadrich, M. (2006). The relationship between precision-recall and ROC curves. In Proceedings of the 23rd international conference on Machine learning (pp. 233-240).
* Rapley, T. (2004). Interviews. In Qualitative Research Practice (pp. 15-33). SAGE Publications Ltd
* Johnson, T. (2019). Understanding JSON: The Ultimate Guide. Data World
* (Berk, K.N., & Carey, P. 2019). Data Analysis with Microsoft Excel.
* Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1-2), 1-135. doi:10.1561/1500000011
* Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, 2(1-2), 1-135. doi:10.1561/1500000011
* Sebastiani, F. (2002). Machine learning in automated text categorization. ACM Computing Surveys (CSUR), 34(1), 1-47.
* Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 1746-1751.
* Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. Linguisticae Investigationes, 30(1), 3-26
* Shumway, R. H., & Stoffer, D. S. (2017). Time series analysis and its applications: With R examples. Springer.
* Caruana, R. (1997). Multitask Learning. Machine Learning, 28(1), 41-75.