I'm Benazir. I will discuss data trends and data preprocessing methods that can help us achieve our future goals. Let's start with the trends we've analysed. The group 5 was interested in knowing the number of pawned users per year. We plotted a line graph with breach year versus pawn count. This is how it looked. As can be seen from this graph, the highest breach occurred in the year 2019. Additionally, we can see that a large number of breaches occurred during the pandemic. A number of data security tools were available on the market after 2019 Macfee Mvision cloud, google cloud security command etc) could have resulted in the dip in the number of pawned count after 2019. Additionally, we were curious as to whether any domain was more vulnerable than others. And we found out that this hacking site has been breached three times and they were in news for some time as well. There were also some other domains that were breached twice. Linkedin is one of the most popular domains among them. A second check was performed to determine whether there had been any delay in reporting that could have resulted in higher compromised user accounts. A line graph was then plotted with the breach year versus the delay in reporting days, which was derived by subtracting the modified date from the breach date. Moreover, we found that the highest number of unnoticed breaches occurred in 2016. Though it is not showing any similarity between the previous graph, the relation can be further investiggated in the next part. Also we have collected another databreach dataset from Kaggle and will be verifying of the trends we analysed is correct or not in next part of this project.

Let's move on to the preparation of the data. We have prepared a few pieces of data and attempted to create one master copy that can be altered for our specific modeling needs later on. First of all, there were three date columns which did not follow the same format. Therefore, we updated them in the same format. In order to determine the delay in reporting, we subtracted the breach date from the modified date. We have also extracted the year from the dates for our machine learning model, since the dates won't be taken and we want to predict the frequency of the breach. Secondly, we removed a few columns that were unnecessary for our objectives. As the title domain and names contained similar information, we decided to keep the domain and drop the other two. Logopath led to the picture image, and we also found another way to know the industry. We decided it wouldn't help in further investigation, so we dropped it as well. Lastly, the added date and modified date were almost identical for most records, so we decided to take the modified date and then extract only the year from it. We decided to search for missing values and duplicates after removing unnecessary columns. Although we did not find any duplicates, we did find 16 missing values, which were all deleted since they did not affect the sample size significantly.

The following step involved creating new columns to obtain valuable information for our objectives. We searched for datasets in Kaggle and other repositories to determine the industry types of the domain, but we were unable to find a dataset that included all the domains in our dataset. Therefore, we manually created a dataset by gathering information from Google on each domain and merged it with our existing dataset. We used text filtering in Python to categorize the domains into 10 broad categories, which are: Gaming industry, Entertainment industry, Finance industry, Government, Health Industry, Manufacturing industry, Retail industry, Service Industry, Marketing industry, and Technology.

Our objective was to identify the most vulnerable industries. We plotted a clustered bar chart for each year to find the top 5 vulnerable industries. We discovered that the technology industry was the most vulnerable, particularly from 2013 to 2022. The marketing industry followed closely behind, with the entertainment and gaming industries trailing shortly after.

The criteria of categorisation has been attached in the appendix for the reference.

We have also added three new columns, such as breach year, reported year, and reported days. By extracting the year from the breach date and the report date, the breach year and reported year were created. In order to determine the delay in reporting, the breach date was subtracted from the modified date and named as the reported days. We will continue to work on the preprocessing of the data in the next part as well. We can further analyze the description field, which contains more valuable information regarding the type of attack, severity of attack, and type of industry. As part of our text-based analysis, we will first remove the HTML tags and punctuation from the text field. Next, we will tokenize the text by breaking it into individual words using libraries such as NLTK or SpaCy. Afterwards, stop words such as "the" and "and" will be removed. After that, we will do stemming in order to retain words in their root form. For example, converting runs running to run. To obtain more meaningful words, we might also consider lemmatizing instead of stemming. After preprocessing the "Description" column, you can use the resulting tokens to extract features for text classification .One common approach is to use the bag-of-words model, which represents each document as a vector of word frequencies. WE will use a machine learning algorithm like Naive Bayes, SVM, or Decision Trees to classify the text data into different categories. For example, the type of data that was compromised or the severity of the data breach.

We had categorical values that needed to be transformed into numerical values for our machine learning models. To accomplish this, we used one hot encoding, which creates new columns for each unique value in the categorical column and assigns a binary value (0 or 1) to each row based on the presence or absence of that value. This results in a sparse matrix with many new columns, but it allows us to use categorical data in our models. The increase in column numbers can help in PCA, which stands for Principal Component Analysis. PCA is a statistical technique that reduces the dimensionality of a dataset by identifying the most important features and combining them into fewer components. Having more columns can provide more information for the PCA algorithm to work with and potentially improve the accuracy of our models.

In order to improve the performance of our machine learning models, it is essential to identify the important features. We ran a linear regression model with pawn count as the dependent variable and found that the r-value was negative, indicating a poor model fit. We then attempted forward feature selection by plotting the number of features versus the r-squared value to determine the optimal number of features. From the graph, it was determined that six features would provide the highest r-squared value with the lowest mean square error value.

We selected these six features and reran the regression model, resulting in an increased r-squared value of 0.024. However, this value is still too low to be statistically significant, so we may attempt standardization and PCA to further improve the model performance.