1. Write about any difficult problem that you solved. (According to us difficult - is something which 90% of people would have only 10% probability in getting a similarly good solution).

Ans:

1. Business Context:

- Malware is any piece of software that was written with the intent of doing harm to data, devices or to people.
- In the past few years, the malware industry has grown so rapidly that the syndicates invest heavily in technologies to evade traditional protection, forcing the anti-malware groups/communities to build more robust software to detect and terminate these attacks. The major part of protecting a computer system from a malware attack is to identify whether a given piece of file/software is a malware.
- Microsoft has been very active in building anti-malware products over the years and it runs its anti-malware utilities on over 150 million computers around the world. This generates tens of millions of daily data points to be analyzed as potential malware. In order to be effective in analyzing and classifying such large amounts of data, we need to be able to group them into groups and identify their respective malware families.
- Machine Learning can be a simple, less expensive, and effective approach to help cyber security teams to be more proactive in preventing threats by classifying malware.

2. Problem Statement:

- This dataset provided by Microsoft contains 9 classes of malware.
- For every malware, we have two files
    a. .asm file (consist of a sequence of operations that are executed by an assembler to generate object code)
    b. .bytes file (the raw data contains the hexadecimal representation of the file's binary content, without the PE header)
- Total dataset consist of 200GB data out of which 50Gb of data is .bytes files and 150GB of data is .asm files:
    a. There are total 10,868 .bytes files and 10,868 asm files total 21,736 files
- There are 9 types of malware in our given data namely:
    a. Ramnit
    b. Lollipop
    c. Kelihos_ver3
    d. Vundo
    e. Simda
    f. Tracur
    g. Kelihos_ver1

h. Obfuscator.ACY
i. Gatak

3. Solution Developed:

- Performed feature extraction on .bytes and .asm files to collect file size and other textual features like unigrams, bigrams and opcodes. Leveraged batch processing and parallel processing for extracting features to prevent memory overshoot and reduce runtime respectively.
- Performed EDA to understand distribution of classes, file sizes, and opcodes.
- Leveraged T-SNE for dimensionality reduction during multivariate analysis.
- The total bigram features are 66049, to reduce the dimensionality removed the zero variance features and then performed feature selection using Chi-Squared test to get the top 5000 bigram features.
- Transformed asm files to images, analyzed images of different classes and added image pixel values as features.
- Designed and Calibrated (wherever needed) KNN, Logistic Regression, Random Forest and XGBoost classifiers on different subsets of features .
- Achieved a Multi-Class log loss of 0.018.

4. Improvements to the Solution:

- Developed multiple Machine Learning Models on different subsets of features (small to large subsets) and observed that as we add more meaningful features to the set the model performance improves by a significant fraction.
- This reinforces the classic Machine Learning concept of garbage in, garbage out (GIGO).
- To improve the performance further we can try adding more meaningful features like Trigrams and implementing TFIDF Vectorization technique on n-grams instead of basic Bag-of-Words (BOW).

5. Link to the project:

- **https://github.com/ChiragPritmanii/microsoft-malware-classification**

2. Explain back propagation and tell us how you handle a dataset if 4 out of 30 parameters have null values more than 40 percentage.

Ans:

1. Backpropagation is a major step in training Gradient based algorithms. Any machine learning cost function which has a well defined first derivative gradient can be optimized using backpropagation.

Backpropagation is the process of updating weights by using derivatives of weights w.r.t the cost function, this process is done to the point where the minimum cost function value is achieved.
When looking at an eagle's view perspective of where backpropagation stands in the complete training process, it's done after each forward pass the forward pass may contain a single data point, a batch of data points or the complete train data which is also known as stochastic gradient descent, mini batch gradient descent and batch gradient descent respectively.
Each time backpropagation is conducted during the process, the weights adjust such that for the next pass the performance of the model is improved this is done to the point where the performance can not improve any further.

2. First approach to handle a dataset with missing values would be to check the missing data behaviour - missing at random, missing completely at random or missing not at random? We can decide to drop or use imputation techniques accordingly.

Second, we check how many rows have missing values, if the amount is inconsiderable compared to our whole data then we may drop them.

Third, we if the missing data is of considerable amount we may try KNN based imputation, this should be the last approach because we can't vouch for our data to be completely organic in nature which means that models on this data may or may not be able to generalize well.