**TIKTOK CLAIMS CLASSIFICATION PROJECT**

**Problem Statement:**

Build a machine learning model to classify video reports as claims or opinions.

**Business Requirements:**

TikTok users have the ability to submit reports that identify videos and comments that contain user claims. These reports identify content that needs to be reviewed by moderators. The process generates a large number of user reports that are challenging to consider in a timely manner.

TikTok is working on the development of a predictive model that can determine whether a video contains a claim or offers an opinion. With a successful prediction model, TikTok can reduce the backlog of user reports and prioritize them more efficiently.

**Steps Followed:**

1. Import necessary libraries and packages required for the project.
2. Load the dataset.
3. Gain necessary basic information about the dataset using EDA.
4. Check for missing values and remove them as they are very less in no as compared to the total no of records in the dataset.
5. Check for duplicate values.
6. Check for class balance of target variable.
7. Extract the length of each video transcription text and add this as a column named text length to the dataframe
8. Calculate the average text length for claims and opinions.
9. Visualize the same using a histogram.
10. Encode the categorical data into numerical data.
11. Split the data into features X and target variable y.
12. Split the data into training and testing sets.
13. Split the training data into training and validation sets.
14. Check the shape of each training, validation, and testing set.
15. Instantiate a random forest classifier, create a dictionary of hyperparameters to tune, creating a list of scoring metrics.
16. Set up a Count Vectorizer object, which converts a collection of text to a matrix of token counts.
17. Extract numerical features from video transcription text in the training set.
18. Place the numerical representation of video transcription text from training set into a dataframe.
19. Concatenate X train and count df to form the final dataframe for training data.
20. Extract numerical features from video transcription text` in the testing set.
21. Place the numerical representation of video transcription text from validation set into a dataframe.
22. Concatenate X val and validation count df to form the final dataframe for training data.
23. Extract numerical features from video transcription text` in the testing set.
24. Place the numerical representation of video transcription text from test set into a dataframe.
25. Concatenate X val and validation count df to form the final dataframe for training data.
26. Fit the model to the new training data after concatenation.
27. Check the best recall score and parameters.
28. Access the GridSearch results and convert it to a pandas dataframe.
29. Examine the GridSearch results dataframe at column mean test precision in the best index.
30. Similarly, build a XGBoost model to compare with the random forest model.
31. Use the random forest best estimator model to get predictions on the validation set.
32. Create classification report for random forest model.
33. Do the same for XGBoost.
34. Compare them and find the better model. Here it is the random forest model.
35. Use it to predict on test data.
36. Create confusion matrix to visualize its’ performance.
37. Plot feature importances of the model.

**Key Findings and Recommendations:**

1. Would you recommend using this model? Why or why not? Yes, one can recommend this model because it performed well on both the validation and test holdout data. Furthermore, both precision and F1 scores were consistently high. The model very successfully classified claims and opinions.
2. What was your model doing? Can you explain how it was making predictions? The model's most predictive features were all related to the user engagement levels associated with each video. It was classifying videos based on how many views, likes, shares, and downloads they received.
3. Are there new features that you can engineer that might improve model performance? Because the model currently performs nearly perfectly, there is no need to engineer any new features.
4. What features would you want to have that would likely improve the performance of your model? The current version of the model does not need any new features. However, it would be helpful to have the number of times the video was reported. It would also be useful to have the total number of user reports for all videos posted by each author.