# Course Six
## The Nuts and Bolts of Machine Learning

### Instructions

Use this PACE strategy document to record decisions and reflections as you work through the end-of-course project. As a reminder, this document is a resource that you can reference in the future and a guide to help consider responses and reflections posed at various points throughout projects.

### Course Project Recap

Regardless of which track you have chosen to complete, your goals for this project are:

- Complete the questions in the Course 6 PACE strategy document
- Answer the questions in the Jupyter notebook project file
- Build a machine learning model
- Create an executive summary for team members and other stakeholders

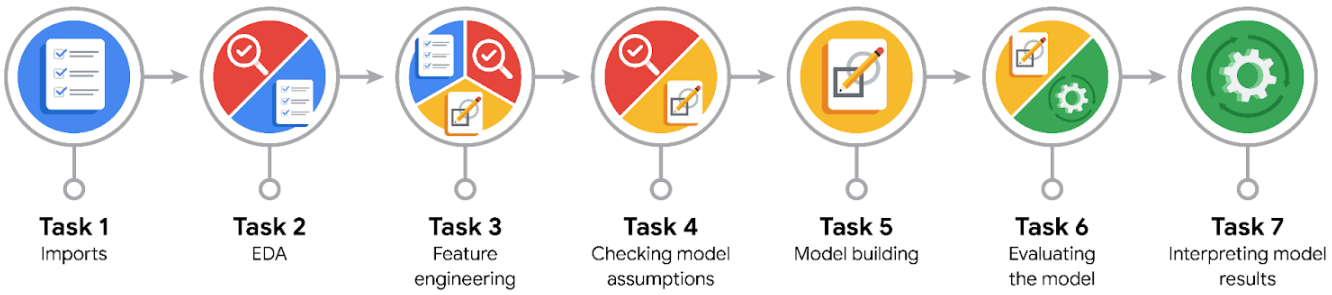
### Relevant Interview Questions

Completing the end-of-course project will empower you to respond to the following interview topics:

- What kinds of business problems would be best addressed by supervised learning models?
- What requirements are needed to create effective supervised learning models?
- What does machine learning mean to you?
- How would you explain what machine learning algorithms do to a teammate who is new to the concept?
- How does gradient boosting work?

## Reference Guide:

This project has seven tasks; the visual below identifies how the stages of PACE are incorporated across those tasks.




## Data Project Questions & Considerations

### PACE: Plan Stage

- What are you trying to solve or accomplish?

To build a Classification model for video claims classification for TikTok in order to distinguish between video reports that are either claims or opinions that are reported by users. This will help TikTok moderators with the immense amount of backlog of reported videos.

- Who are your external stakeholders that I will be presenting for this project?

The external stakeholders that I will be presenting for this project are Willow Jaffey (Data Science Lead), Mary Joanna Rodgers (Project Management Officer) and Maika Abadi (Operations Lead).

- What resources do you find yourself using as you complete this stage?

The resources I am using to complete this stage is the Pandas library in order to import the data as a DataFrame.

- Do you have any ethical considerations at this stage?

That a False Negative, may allow potentially harmful misinformation to remain unchecked on the platform, thereby reducing trust and possibly causing reputational or even real-world harm. If the video is classified as a False Positive, then this may waste moderation resources by unnecessarily

flagging opinion videos, causing inefficiencies in review. However, this is a less severe risk than letting harmful claims slip through.

- Is my data reliable?

Yes, the data is fairly reliable since it is first-party data from TikTok and directly reflects how videos were labeled and engaged with on the platform. However, it may still have some bias in labelling and doesn't capture every type of TikTok video. So, while reliable for this project, it isn't perfect and results should be interpreted carefully.

- What data do I need/would like to see in a perfect world to answer this question?

In a perfect world, I would like to have fact-checking data linked to each video in order to confirm whether claims are true or false, along with user credibility history such as whether an account often spreads misinformation. I would also want more detailed text features, such as sentiment, linguistic style or topic categories as well as temporal data showing how engagement changes over time. Together, these would give a much richer picture for classifying claims versus opinions.

- What data do I have/can I get?

I have access to the tiktok_dataset.csv, which includes information such as the video transcription text, claim status, video duration, engagement metrics (views, likes, comments, downloads and shares) and author details(ban status, verification status). These features provide a solid foundation for building models in order to classify claims versus opinions.

- What metric should I use to evaluate success of my business/organizational objective? Why?

The most important evaluation metric for this objective is recall, due to the priority is to correctly identify as many claims videos as possible. Missing a claim (False Negative) could allow misinformation to spread unchecked, which is riskier than mistakenly flagging an opinion (False Positive). Precision and F1 scores are also useful to track, but recall should be emphasized since it directly supports the goal of reducing the spread of misinformation.

## PACE: Analyze Stage

- Revisit "What am I trying to solve?" Does it still work? Does the plan need revising?

Yes the objective of building and evaluating a Machine Learning Model that can classify TikTok videos as either claims or opinions still works and the plan does not need revising.

- Does the data break the assumptions of the model? Is that ok, or unacceptable?

Due to the type of Models I will be using are Decision Tree Based Models (Random Forest and XGBoost), they don't assume linearity, normal distribution or homoscedasticity and so the data does not break any critical assumptions. This is acceptable and is one of the reasons that these Models are a strong choice. However, if I used Models like Logistic Regression, Skewed Distributions and Multicollinearity would need closer examination.

- Why did you select the X variables you did?

I selected the X features such as video view, like, share, comment, download count as well as text length and author status due to these are likely related to how users engage with content, which can indicate whether a post is more claim like or opinion like.

- What are some purposes of EDA before constructing a model?

EDA helps to understand the underlying structure of the dataset while also checking for missing values, outliers, feature distributions and explore the potential relationships between features and the target variable. It also helps guide feature engineering and ensures that the chosen model aligns with the data's characteristics.

- What has the EDA told you?

The EDA shows that the engagement metrics such as views, likes, comments, downloads and shares had outliers which were handled with. There was also null values that were dropped and the data was checked for duplicates. The class balance for the target variable was also checked, and the text length was generated from video transcription text and the average text length was calculated for each class of the target variable. The distribution of text length against claim status was also performed with a showcase of two histograms in the same plot, one for each class.

- What resources do you find yourself using as you complete this stage?

I find myself using the Pandas, NumPy, Seaborn and Matplotlib.pyplot libraries in order to conduct EDA.

**PACE: Construct Stage**

- Do I notice anything odd? Is it a problem? Can it be fixed? If so, how?

I noticed that both of the models achieved a near perfect performance which may suggest that the dataset is relatively easy to classify or that there could be a possibility of the model's overfitting the data. Therefore, the models should be evaluated on unseen holdout data to ensure they aren't overfitting the data. Rewrite this

- Which independent variables did you choose for the model, and why?

I selected numerical engagement features such as video views, likes, shares, comments and downloads; account status features (ban, verified) and video metadata such as duration and text length. These variables were chosen because they directly influence how claims versus opinions may be differentiated in TikTok content.

- How well does your model fit the data? What is my model's validation score?

The Random Forest Model achieved a recall score of 0.99 and a precision score of 0.999 and the XGBoost Model achieved a recall score of 0.99 and a precision score of 0.998. Therefore, this indicates that both of the models fit the data extremely well with the Random Forest Model performing just barely better than the XGBoost Model in Precision and Recall.

- Can you improve it? Is there anything you would change about the model?

Both of these Models perform outstandingly well but some improvements could be: Testing on unseen data in order to check for model overfitting, Performing more feature engineering and simplifying the Model if interpretability is a top priority.

- What resources do you find yourself using as you complete this stage?

I used Scikit-Learn's module's such as metrics, model selection and ensemble to get functions like train_test_split, GridSearchCV and Random Forest Classifier as well as evaluation metric functions. Pandas was useful for converting the cv_results_ NumPy array into a DataFrame in order to check the precision score of the best model. The XGBoost library was also useful in importing the XGBClassifier function in order to instantiate the XGBoost Classifier.

## PACE: Execute Stage

- What key insights emerged from your model(s)? Can you explain my model?

Both the Random Forest and XGBoost Models achieved a near perfect classification performance, with precision, recall and F1 scores close to 1.0 for both opinions and claims classes. This therefore indicates that the models are extremely effective at distinguishing between the two classes. The Random Forest Model in particular provides interpretability with being able to explain the predictions and also with feature importance, which helps to understand which features most strongly influence predictions.

- What are the criteria for model selection?

The criteria used for Model selection includes precision, recall, F1 score and interpretability and generalizability. Since both of the Models performed equally well in terms of metrics, the interpretability of the Model became the deciding factor as it means an ease of communication with Stakeholders. Random Forest Classification Model was preferred over the XGBoost Model due to GBMs are often considered black box Models, thereby making them less transparent.

- Does my model make sense? Are my final results acceptable?

Yes, the Model makes sense and the results are highly acceptable. With the Model's precision and recall being near 1.0, the Model has thereby demonstrated its ability to correctly classify claims and opinions with almost no errors. This confirms that the Model is therefore well trained and reliable for the project task.

- Do you think your model could be improved? Why or why not? How?

Even though the Model performs exceptionally well, there is always improvements to be made to the model, due to the high recall and precision scores, it is a good measure to test the Model on unseen data to make sure that the Model is not overfitting the data. Also experimenting with additional engineered features could provide more in detail signals for distinguishing between claims and opinions.

- Were there any features that were not important at all? What if you take them out?

Yes as some of the features had little to no impact on the classification, such as verified status, author ban status and video duration as evident on the Random Forest Feature Importance Plot. Removing these non important features can reduce Model complexity and training time without significantly affecting accuracy, although in this case with the already high performance means that the feature reduction would be more about efficiency than accuracy gains.

- What business/organizational recommendations do you propose based on the models built?

Both of the Models can be deployed to automatically detect claims versus opinions in TikTok content, thereby enabling improving content moderation, misinformation detection and trend analysis. TikTok can use this tool to filter, categorize and prioritize claim based content for fact-checking while allowing opinions to flow freely. Therefore, improving content integrity and user trust.

- Given what you know about the data and the models you were using, what other questions could you address for the team?

  Can the Model be extended to detect different types of claims? How does the Model perform on multilingual TikTok data? Can this approach be scaled for real-time content monitoring on a large volume of posts? What are the ethical implications of automating claim detection in social platforms?

- What resources do you find yourself using as you complete this stage?

  I used the Scikit-Learn, Matplotlib, Pandas and Seaborn in this stage with the classification_report, confusion_matrix, ConfusionMatrixDisplay functions and the .predict, .best_estimator_ and .feature_importances_ attributes from the GridSearchCV Function. These resources helped evaluate the Models and interpret the performance of the Classifiers.

- Is my model ethical?

  The Model is ethical as long as it is used transparently and responsibly. The Model's primary goal is to distinguish between claims and opinions, which supports fact-checking and combats misinformation. However, ethical considerations to consider are bias in the training data and their being a potential misuse for censorship. Clear communication and governance are essential to ensure fairness and transparency.

- When my model makes a mistake, what is happening? How does that translate to my use case?

  When the Model makes a mistake, it may misclassify a claim as an opinion or vice versa. In practice, this could mean that a claim might go unchecked, or an opinion could be flagged unnecessarily. While errors are minimal, organizations should have a human-in-the-loop system to review edge cases and ensure that misclassifications do not harm decision-making.