Activity_Course 6 TikTok project lab

July 18, 2023

1 TikTok Project

Course 6 - The Nuts and bolts of machine learning

Recall that you are a data professional at TikTok. Your supervisor was impressed with the work you have done and has requested that you build a machine learning model that can be used to determine whether a video contains a claim or whether it offers an opinion. With a successful prediction model, TikTok can reduce the backlog of user reports and prioritize them more efficiently.

A notebook was structured and prepared to help you in this project. A notebook was structured and prepared to help you in this project. Please complete the following questions.

2 Course 6 End-of-course project: Classifying videos using machine learning

In this activity, you will practice using machine learning techniques to predict on a binary outcome variable.

The purpose of this model is to increase response time and system efficiency by automating the initial stages of the claims process.

The goal of this model is to predict whether a TikTok video presents a "claim" or presents an "opinion".

This activity has three parts:

Part 1: Ethical considerations * Consider the ethical implications of the request

- Should the objective of the model be adjusted?

Part 2: Feature engineering

- Perform feature selection, extraction, and transformation to prepare the data for modeling

Part 3: Modeling

- Build the models, evaluate them, and advise on next steps

Follow the instructions and answer the questions below to complete the activity. Then, you will complete an Executive Summary using the questions listed on the PACE Strategy Document.

Be sure to complete this activity before moving on. The next course item will provide you with a completed exemplar to compare to your own work.

3 Classify videos using machine learning

4 PACE stages

Throughout these project notebooks, you'll see references to the problem-solving framework PACE. The following notebook components are labeled with the respective PACE stage: Plan, Analyze, Construct, and Execute.

4.1 PACE: Plan

Consider the questions in your PACE Strategy Document to reflect on the Plan stage.

In this stage, consider the following questions:

1. **What are you being asked to do? What metric should I use to evaluate success of my business/organizational objective?**

2. **What are the ethical implications of the model? What are the consequences of your model making errors?**

- What is the likely effect of the model when it predicts a false negative (i.e., when the model says a video does not contain a claim and it actually does)?

- What is the likely effect of the model when it predicts a false positive (i.e., when the model says a video does contain a claim and it actually does not)?

3. **How would you proceed?**

==> ENTER YOUR RESPONSES HERE

4.1.1 Task 1. Imports and data loading

Start by importing packages needed to build machine learning models to achieve the goal of this project.

[ ]:

```
# Import packages for data manipulation
### YOUR CODE HERE ###
import pandas as pd
import numpy as np

# Import packages for data visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Import packages for data preprocessing
from sklearn.feature_extraction.text import CountVectorizer

# Import packages for data modeling
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import classification_report, accuracy_score, precision_score, \
recall_score, f1_score, confusion_matrix, ConfusionMatrixDisplay
```

```
from sklearn.ensemble import RandomForestClassifier
from xgboost import XGBClassifier
from xgboost import plot_importance

# Import packages for data visualization
### YOUR CODE HERE ###


# Import packages for data preprocessing
### YOUR CODE HERE ###


# Import packages for data modeling
### YOUR CODE HERE ###
```

Now load the data from the provided csv file into a dataframe.

Note: As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

[ ]:

```
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

4.2 PACE: Analyze

Consider the questions in your PACE Strategy Document to reflect on the Analyze stage.

4.2.1 Task 2: Examine data, summary info, and descriptive stats

Inspect the first five rows of the dataframe.

[ ]:

```
# Display first few rows
### YOUR CODE HERE ###
data.head(5)
```

Get the number of rows and columns in the dataset.

[ ]:

```
# Get number of rows and columns
### YOUR CODE HERE ###
data.shape
```

Get the data types of the columns.

[ ]:

```
# Get data types of columns
### YOUR CODE HERE ###
data.dtypes
```

Get basic information about the dataset.

[ ]:

```
# Get basic information
### YOUR CODE HERE ###
data.info()
```

Generate basic descriptive statistics about the dataset.

[ ]:

```
# Generate basic descriptive stats
### YOUR CODE HERE ###
data.describe()
```

Check for and handle missing values.

[ ]:

```
# Check for missing values
### YOUR CODE HERE ###
data.isna()
```

[ ]:

```
# Drop rows with missing values
### YOUR CODE HERE ###
data.dropna(axis = 0 )
```

[ ]:

```
# Display first few rows after handling missing values
### YOUR CODE HERE ###
data.head(5)
```

Check for and handle duplicates.

[ ]:

```
# Check for duplicates
### YOUR CODE HERE ###
data.isna().sum()
```

Check for and handle outliers.

[ ]:

```
### YOUR CODE HERE ###

data = data.dropna()
data.duplicated().sum()
```

Check class balance.

[ ]:

```
# Check class balance
### YOUR CODE HERE ###
data['claim_status'].value_counts(normalize = True)
```

4.3 PACE: Construct

Consider the questions in your PACE Strategy Document to reflect on the Construct stage.

4.3.1 Task 3: Feature engineering

Extract the length of each `video_transcription_text` and add this as a column to the dataframe, so that it can be used as a potential feature in the model.

[14]:
```
# Extract the length of each `video_transcription_text` and add this as a␣
 ↪column to the dataframe
### YOUR CODE HERE ###
data['text_length'] = data['video_transcription_text'].str.len()
data['text_length']
```

[14]:
```
0         97
1        107
2        137
3        131
4        128
        …
19079     65
19080     66
19081     53
19082     80
19083     70
Name: text_length, Length: 19084, dtype: int64
```

Calculate the average text_length for claims and opinions.

[15]:
```
# Display first few rows of dataframe after adding new column
### YOUR CODE HERE ###
data.head(5)
```

[15]:
```
   # claim_status    video_id  video_duration_sec  \
0  1        claim  7017666017                  59
1  2        claim  4014381136                  32
2  3        claim  9859838091                  31
3  4        claim  1866847991                  25
4  5        claim  7105231098                  19

                            video_transcription_text verified_status  \
0  someone shared with me that drone deliveries a…    not verified
1  someone shared with me that there are more mic…    not verified
2  someone shared with me that american industria…    not verified
3  someone shared with me that the metro of st. p…    not verified
4  someone shared with me that the number of busi…    not verified

  author_ban_status  video_view_count  video_like_count  video_share_count  \
0      under review          343296.0           19425.0              241.0
1            active          140877.0           77355.0            19034.0
2            active          902185.0           97690.0             2858.0
```

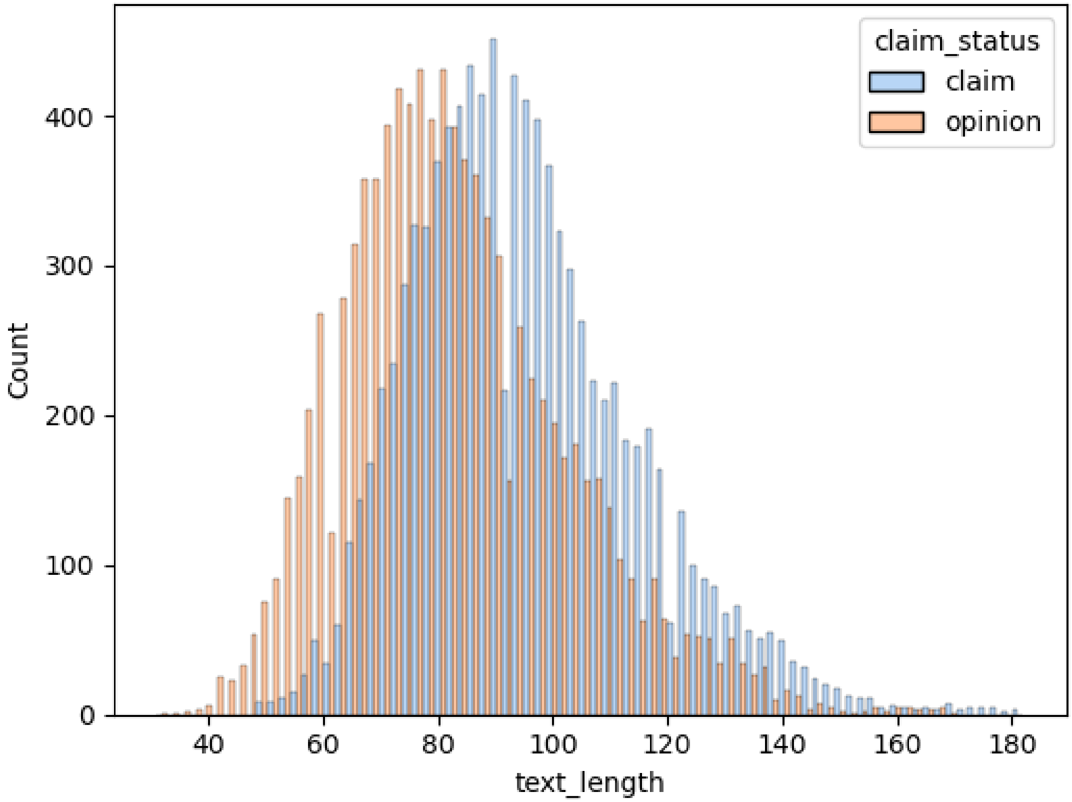
```
3           active          437506.0          239954.0             34812.0
4           active           56167.0           34987.0              4110.0

   video_download_count  video_comment_count  text_length
0                   1.0                  0.0           97
1                1161.0                684.0          107
2                 833.0                329.0          137
3                1234.0                584.0          131
4                 547.0                152.0          128
```

Visualize the distribution of `video_transcription_text` length for claims and opinions.

[16]:

```python
# Visualize the distribution of `video_transcription_text` length for claims and opinions
# Create two histograms in one plot
### YOUR CODE HERE ###
sns.histplot(data = data,stat = "count",multiple = "dodge",kde = False,palette = "pastel",x= 'text_length',hue="claim_status")
plt.show()
```



Create a heatmap to visualize how correlated variables are. Consider which variables you're interested in examining correlations between.

[ ]:

```
# Create a heatmap to visualize how correlated variables are
### YOUR CODE HERE ###
data_encoded = pd.get_dummies(data)

sns.heatmap(data_encoded.corr(),vmax=.3, center=0,
            square=True, linewidths=.5, cbar_kws={"shrink": .5})
```

One of the model assumptions for logistic regression is no severe multicollinearity among the features. Take this into consideration as you examine the heatmap and choose which features to proceed with.

4.4 PACE: Construct

Consider the questions in your PACE Strategy Document to reflect on the Construct stage.

4.4.1 Task 3. Feature engineering

Select the outcome variable.

[ ]:

```
# Select outcome variable
X = data.copy()
y = data['claim_status']
X = X.drop(['claim_status','#','video_id'],axis = 1)
X['claim_status'] = np.where(X['claim_status']== 'claim',1,0)
X = pd.get_dummies(X,
                   columns=['verified_status', 'author_ban_status'],
                   drop_first=True)
X.head(5)

### YOUR CODE HERE ###
```

Feature selection and transformation

Encode target and catgorical variables.

[ ]:

```
### YOUR CODE HERE ###
data_encoded = pd.get_dummies(data)
# Display first few rows
### YOUR CODE HERE ###
```

4.4.2 Task 4. Split the data

Assign target variable.

[ ]:

```
### YOUR CODE HERE ###
X = data_encoded.drop('claim_status',axis = 1)
```

Isolate the features.

[ ]:

```
#Isolate features
### YOUR CODE HERE ###
y = X['claim_status']
X = X.drop(['claim_status'],axis = 1)

# Display first few rows of features dataframe
### YOUR CODE HERE ###
```

Task 5: Create train/validate/test sets Split data into training and testing sets, 80/20.

[ ]:

```
### YOUR CODE HERE ###
X_train , X_test, y_train,y_test = train_test_split(X, y, stratify =␣
 ↪y,test_size = 0.2,random_state = 0  )
```

Split the training set into training and validation sets, 75/25, to result in a final ratio of 60/20/20 for train/validate/test sets.

[ ]:

```
### YOUR CODE HERE ###
X_train, X_val, y_train, y_val = train_test_split(X,y,stratify = y,test_size =␣
 ↪0.25,random_state = 0)
```

Confirm that the dimensions of the training, validation, and testing sets are in alignment.

[ ]:

```
### YOUR CODE HERE ###
X_train.shape, X_val.shape, X_test.shape, y_train.shape, y_val.shape, y_test.
 ↪shape
```

4.4.3 Task 6. Build models

4.4.4 Build a random forest model

Fit a random forest model to the training set. Use cross-validation to tune the hyperparameters and select the model that performs best on recall.

[ ]:

```
# Instantiate the random forest classifier
### YOUR CODE HERE ###

# Create a dictionary of hyperparameters to tune
### YOUR CODE HERE ###



# Define a dictionary of scoring metrics to capture
### YOUR CODE HERE ###

# Instantiate the GridSearchCV object
### YOUR CODE HERE ###
```

[ ]:
```
# Examine best recall score
### YOUR CODE HERE ###
```

[ ]:
```
  # Get all the results from the CV and put them in a df
### YOUR CODE HERE ###

  # Isolate the row of the df with the max(mean precision score)
### YOUR CODE HERE ###
```

[ ]:
```
# Examine best parameters
### YOUR CODE HERE ###
```

Question: How well is your model performing? Consider average recall score and precision score.

4.4.5 Build an XGBoost model

[ ]:
```
# Instantiate the XGBoost classifier
### YOUR CODE HERE ###

# Create a dictionary of hyperparameters to tune
### YOUR CODE HERE ###

# Define a dictionary of scoring metrics to capture
### YOUR CODE HERE ###

# Instantiate the GridSearchCV object
### YOUR CODE HERE ###
```

[ ]:
```
  # Get all the results from the CV and put them in a df
### YOUR CODE HERE ###

  # Isolate the row of the df with the max(mean precision score)
### YOUR CODE HERE ###
```

Question: How well does your model perform? Consider recall score and precision score.

4.5 PACE: Execute

Consider the questions in your PACE Strategy Document to reflect on the Execute stage.

4.5.1 Task 7. Evaluate model

Evaluate models against validation criteria.

Random forest

[ ]:
```
# Use the random forest "best estimator" model to get predictions on the␣
 ↪encoded testing set
### YOUR CODE HERE ###
```

Display the predictions on the encoded testing set.

```
[ ]: # Display the predictions on the encoded testing set
     ### YOUR CODE HERE ###
```

Display the true labels of the testing set.

```
[ ]: # Display the true labels of the testing set
     ### YOUR CODE HERE ###
```

Create a confusion matrix to visualize the results of the classification model.

```
[ ]: # Create a confusion matrix to visualize the results of the classification model

     # Compute values for confusion matrix
     ### YOUR CODE HERE ###

     # Create display of confusion matrix
     ### YOUR CODE HERE ###

     # Plot confusion matrix
     ### YOUR CODE HERE ###

     # Display plot
     ### YOUR CODE HERE ###
```

Create a classification report that includes precision, recall, f1-score, and accuracy metrics to evaluate the performance of the model.

```
[ ]: # Create a classification report
     # Create classification report for random forest model
     ### YOUR CODE HERE ###
```

Question: What does your classification report show? What does the confusion matrix indicate?

XGBoost

```
[ ]: #Evaluate XGBoost model
     ### YOUR CODE HERE ###
```

```
[ ]: # Compute values for confusion matrix
     ### YOUR CODE HERE ###

     # Create display of confusion matrix
     ### YOUR CODE HERE ###

     # Plot confusion matrix
     ### YOUR CODE HERE ###
```

```
# Display plot
### YOUR CODE HERE ###
```

[ ]:

```
# Create a classification report
### YOUR CODE HERE ###
```

Question: Describe your XGBoost model results. How does your XGBoost model compare to your random forest model?

4.5.2 Use champion model to predict on test data

[ ]:

```
### YOUR CODE HERE ###
```

[ ]:

```
# Compute values for confusion matrix
### YOUR CODE HERE ###

# Create display of confusion matrix
### YOUR CODE HERE ###

# Plot confusion matrix
### YOUR CODE HERE ###

# Display plot
### YOUR CODE HERE ###
```

Feature importances of champion model

[ ]:

```
### YOUR CODE HERE ###
```

Question: Describe your most predictive features. Were your results surprising?

4.5.3 Task 8. Conclusion

In this step use the results of the models above to formulate a conclusion. Consider the following questions:

1. **Would you recommend using this model? Why or why not?**
2. **What was your model doing? Can you explain how it was making predictions?**
3. **Are there new features that you can engineer that might improve model performance?**
4. **What features would you want to have that would likely improve the performance of your model?**

Remember, sometimes your data simply will not be predictive of your chosen target. This is common. Machine learning is a powerful tool, but it is not magic. If your data does not contain predictive signal, even the most complex algorithm will not be able to deliver consistent and accurate predictions. Do not be afraid to draw this conclusion.

==> ENTER YOUR RESPONSES HERE

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.