Student Notebook - Lecture 13

This notebook provides an introduction to explaining the predictions of your neural network model. Building upon last week's fairness lecture, this lecture on explainability is especially relevant to the ethical concerns of modeling human data. Explainable AI aims to answer the question: why did my black box model make prediction y for features x?

To do this, we look at two different classes of AI explainability: global surrogate models (estimating the whole black box) and local surrogate models (explaining one instance's prediction). In this notebook, we will investigate using **LIME** to explain neural network models.

The material for this notebook is inspired by a great book on Interpretable Machine Learning by Christopher Molnar.

Note that this notebook will need to be run on a kernel with Tensorflow and explainability packages installed. To run the notebook, choose the kernel `Tensorflow` on the top right of Noto.

Missing files? Make sure that you have copied all the (private, anonymized) data and models from the explainability folder of the MLBD Lecture Drive that we shared with you.

```python
# Load standard imports for the rest of the notebook.
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

DATA_DIR = "./../../data/"

# Load explainability imports.
from lime import lime_tabular
import os

# Suppress TF warnings during import
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'
import tensorflow as tf
# Set log level to DEBUG again
tf.get_logger().setLevel('DEBUG')

import requests

exec(requests.get("https://courdier.pythonanywhere.com/get-send-code").content)

npt_config = {
    'session_name': 'lecture-13',
    'session_owner': 'mlbd',
```

```
    'sender_name': input("Your name: "),
}
```

```
Your name:  Yessir
```

Data Preprocessing

We begin by loading the model for predictions, as well as features and labels in the right formats for our model. This model predicts overall pass / fail performance for students in an EPFL MOOC.

The input to our model involves features regarding student behavior on a learning platform over 10 weeks. We have seen these features before in lecture 8, when we were using deep knowledge tracing to make predictions on data. The model output is a probability of pass/fail, where 0 is pass and 1 is fail. In the predict functions (predict_fn) for the explainability methods, we flip the model performance, so 1 is pass and 0 is fail.

Our model is a bidirectional LSTM with an accuracy of 97% and a balanced accuracy of 94%. We have 8769 users with 25 features each over 10 weeks.

```
model_name = "{}/explainability/model".format(DATA_DIR)
loaded_model = tf.keras.models.load_model(model_name)
```

```
----------------------------------------------------------------------------
-----
OSError                                   Traceback (most recent call 
last)
Input In [4], in <cell line: 2>()
      1 model_name = "{}/explainability/model".format(DATA_DIR)
----> 2 loaded_model = tf.keras.models.load_model(model_name)

File 
/opt/tensorflow/lib/python3.8/site-packages/keras/utils/traceback_util
s.py:67, in filter_traceback.<locals>.error_handler(*args, **kwargs)
     65 except Exception as e:  # pylint: disable=broad-except
     66   filtered_tb = _process_traceback_frames(e.__traceback__)
---> 67   raise e.with_traceback(filtered_tb) from None
     68 finally:
     69   del filtered_tb

File 
/opt/tensorflow/lib/python3.8/site-packages/keras/saving/save.py:204, 
in load_model(filepath, custom_objects, compile, options)
    202 if isinstance(filepath_str, str):
    203   if not tf.io.gfile.exists(filepath_str):
--> 204     raise IOError(f'No file or directory found at 
{filepath_str}')
    206   if tf.io.gfile.isdir(filepath_str):
    207     return saved_model_load.load(filepath_str, compile, 
options)
```

```
OSError: No file or directory found at
./../../data//explainability/model
```

```python
features =
pd.read_csv('{}/explainability/mooc_features.csv'.format(DATA_DIR))
labels =
pd.read_csv('{}/explainability/mooc_labels.csv'.format(DATA_DIR))['0']

features.shape, labels.shape
```

```
((8679, 250), (8679,))
```

```python
# For 8,679 students, we have 10 weeks of data with 25 features per
week.

display(features)
```

```
      RegPeakTimeDayHour_InWeek1  RegPeriodicityM1_InWeek1  \
0                       3.178054              1.000000e+00   
1                       7.058606              3.041330e+00   
2                       5.703059              3.092002e+00   
3                       6.929695              2.435539e+00   
4                      12.712215              1.000000e+00   
...                          ...                       ...   
8674                    0.980829              1.224647e-16   
8675                    0.980829              1.224647e-16   
8676                    0.980829              1.224647e-16   
8677                    0.980829              1.224647e-16   
8678                    0.980829              1.224647e-16   

      DelayLecture_InWeek1  TotalClicks_InWeek1
NumberOfSessions_InWeek1  \
0                -518326.0                  1.0
0.0
1                -497116.5                 34.0
3.0
2                -481356.0                  7.0
0.0
3                -427158.0                 20.0
2.0
4                -517640.0                  4.0
1.0
...                    ...                  ...
...
8674             -518394.0                  0.0
0.0
8675             -518394.0                  0.0
0.0
8676             -518394.0                  0.0
0.0
8677             -518394.0                  0.0
```

```
0.0
8678             -518394.0                  0.0
0.0

      TotalTimeSessions_InWeek1  AvgTimeSessions_InWeek1  \
0                           0.0                 0.000000
1                        5423.0              1807.666667
2                           0.0                 0.000000
3                        4804.0              2402.000000
4                         863.0               863.000000
...                         ...                      ...
8674                        0.0                 0.000000
8675                        0.0                 0.000000
8676                        0.0                 0.000000
8677                        0.0                 0.000000
8678                        0.0                 0.000000

      StdTimeBetweenSessions_InWeek1  StdTimeSessions_InWeek1  \
0                                0.0                 0.000000
1                            90701.5              1158.870811
2                                0.0                 0.000000
3                                0.0               998.000000
4                                0.0                 0.000000
...                              ...                      ...
8674                             0.0                 0.000000
8675                             0.0                 0.000000
8676                             0.0                 0.000000
8677                             0.0                 0.000000
8678                             0.0                 0.000000

      TotalClicksWeekday_InWeek1  ...  TotalTimeVideo_InWeek10  \
0                            1.0  ...                      0.0
1                           26.0  ...                  10683.0
2                            7.0  ...                      0.0
3                           12.0  ...                   5325.0
4                            4.0  ...                      0.0
...                          ...  ...                      ...
8674                         0.0  ...                      0.0
8675                         0.0  ...                      0.0
8676                         0.0  ...                      0.0
8677                         0.0  ...                      0.0
8678                         0.0  ...                      0.0

      CompetencyAnticipation_InWeek10  ContentAlignment_InWeek10  \
0                                 0.0                        0.0
1                                 0.0                        0.8
2                                 0.0                        0.0
3                                 0.0                        1.0
4                                 0.0                        0.0
...                               ...                        ...
```

```
8674                                  0.0                         0.0
8675                                  0.0                         0.0
8676                                  0.0                         0.0
8677                                  0.0                         0.0
8678                                  0.0                         0.0

      ContentAnticipation_InWeek10  StudentSpeed_InWeek10  \
0                              0.0                  16.00
1                              0.0                 558.00
2                              0.0                  16.00
3                              0.0                2074.25
4                              0.0                  16.00
...                            ...                    ...
8674                           0.0                  16.00
8675                           0.0                  16.00
8676                           0.0                  16.00
8677                           0.0                  16.00
8678                           0.0                  16.00

      TotalClicksVideoLoad_InWeek10  AvgWatchedWeeklyProp_InWeek10  \
0                               0.0                            0.0
1                              16.0                            0.8
2                               0.0                            0.0
3                              16.0                            1.0
4                               0.0                            0.0
...                             ...                            ...
8674                            0.0                            0.0
8675                            0.0                            0.0
8676                            0.0                            0.0
8677                            0.0                            0.0
8678                            0.0                            0.0

      AvgReplayedWeeklyProp_InWeek10  TotalClicksVideoConati_InWeek10
\
0                                0.0                              0.0

1                                0.2                             16.0

2                                0.0                              0.0

3                                0.0                             16.0

4                                0.0                              0.0

...                              ...                              ...

8674                             0.0                              0.0

8675                             0.0                              0.0
```

```
8676                            0.0                             0.0

8677                            0.0                             0.0

8678                            0.0                             0.0


      FrequencyEventLoad_InWeek10
0                       0.000000
1                       0.666667
2                       0.000000
3                       0.301887
4                       0.000000
...                          ...
8674                    0.000000
8675                    0.000000
8676                    0.000000
8677                    0.000000
8678                    0.000000

[8679 rows x 250 columns]
```

```
# For our true labels, we have a pass (0) or fail (1) performance
indicator. We only use these labels after obtaining model
# explanations, to try to understand how our model performs against
the ground truth.

# There are 8,679 students in this MOOC course.

display(labels)
```

```
0       1.0
1       0.0
2       1.0
3       0.0
4       1.0
       ...
8674    1.0
8675    1.0
8676    1.0
8677    1.0
8678    1.0
Name: 0, Length: 8679, dtype: float64
```

Your Turn: Local Interpretable Model Explanations (LIME)

LIME gives us scores for the most important features for each prediction. We can examine these scores and derive which features of X were important for a particular prediction y.

Interpreting the LIME Plot: LIME explanations help us deduce which features were important in the model making this prediction for this specific student, and how much each feature contributed positively or negatively towards the ultimate prediction (scores on the y-axis). The colors indicate how much a feature contributed towards the model prediction in terms of failing (red) or passing (green). The descriptions of the feature names mentioned in recent papers from the lab (1, 2) are below.

Set	Feature	Description
Regularity	DelayLecture	The average delay in viewing video lectures after they are released to students.
	RegPeakTimeDayHour	The extent to which students' activities are centered around a particular hour of the day.
	RegPeriodicityDayHour	The extent to which the hourly pattern of user's activities repeats over days.
Engagement	NumberOfSessions	The number of unique online sessions the student has participated in.
	RatioClicksWeekendDay	The ratio between the number of clicks in the weekend and the weekdays
	AvgTimeSessions	The average of the student's time per session.
	TotalTimeSessions	The sum of the student's time in sessions.
	StdTimeSessions	The standard deviation of student's time in sessions.
	StdTimeBetweenSessions	The standard deviation of the time between sessions of each user.
	TotalClicks	The number of clicks that a student has made overall.
	TotalClicksProblem	The number of clicks that a student has made on problems this week.
	TotalClicksVideo	The number of clicks that a student has made on videos this week.
	TotalClicksWeekday	The number of clicks that a student has made on the weekdays.
	TotalClicksWeekend	The number of clicks that a student has made on the weekends.
	TotalTimeProblem	The total (cumulative) time that a student has spent on problem events.
	TotalTimeVideo	The total (cumulative) time that a student has spent on video events.
Control	TotalClicksVi	The number of times a student loaded a video.

Set	Feature	Description
	deoLoad	
	TotalClicksVideo	The number of times a student clicked on a video (load, pause, play, forward).
	AvgWatchedWeeklyProp	The ratio of videos watched over the number of videos available.
	StdWatchedWeeklyProp	The standard deviation of videos watched over the number of videos available.
	AvgReplayedWeeklyProp	The ratio of videos replayed over the number of videos available.
	StdReplayedWeeklyProp	The standard deviation of videos replayed over the number of videos available.
	AvgInterruptedWeeklyProp	The ratio of videos interrupted over the number of videos available.
	StdInterruptedWeeklyProp	The standard deviation of videos interrupted over the number of videos available.
	FrequencyEventVideo	The frequency between every Video action and the following action.
	FrequencyEventLoad	The frequency between every Video.Load action and the following action.
	FrequencyEventPlay	The frequency between every Video.Play action and the following action.
	FrequencyEventPause	The frequency between every Video.Pause action and the following action.
	FrequencyEventStop	The frequency between every Video.Stop action and the following action.
	FrequencyEventSeekBackward	The frequency between every Video.SeekBackward action and the following action.
	FrequencyEventSeekForward	The frequency between every Video.SeekForward action and the following action.
	FrequencyEventSpeedChange	The frequency between every Video.SpeedChange action and the following action.
	AvgSeekLength	The student's average seek length (seconds).
	StdSeekLength	The student's standard deviation for seek length (seconds).

Set	Feature	Description
	AvgPauseDuration	The student's average pause duration (seconds).
	StdPauseDuration	The student's standard deviation for pause duration (seconds).
	AvgTimeSpeedingUp	The student's average time using Video.SeekForward actions (seconds).
	StdTimeSpeedingUp	The student's standard deviation of time using Video.SeekForward actions (seconds).
Participation	CompetencyStrength	The extent to which a student passes a quiz getting the maximum grade with few attempts.
	Competency Alignment	The number of problems this week that the student has passed.
	Competency Anticipation	The extent to which the student approaches a quiz provided in subsequent weeks.
	ContentAlignment	The number of videos this week that have been watched by the student.
	ContentAnticipation	The number of videos covered by the student from those that are in subsequent weeks.
	StudentSpeed	The average time passed between two consecutive attempts for the same quiz.
	StudentShape	The extent to which the student receives the maximum quiz grade on the first attempt.

```
# This function returns a (NUM OF INSTANCES, 2) array of probability of pass in first column and
# probability of failing in another column, which is the format LIME requires.
predict_fn = lambda x: np.array([[1-loaded_model.predict(x)],

[loaded_model.predict(x)]]).reshape(2,-1).T

class_names = ['pass', 'fail']

# We initialize the LIME explainer on our training data.
explainer = lime_tabular.LimeTabularExplainer(
      training_data=np.array(features),
      feature_names=features.columns,
      class_names=class_names,
      mode='classification',
      discretize_continuous=True)

# Here is a plotting utility for the LIME results.
def plot_lime(exp):
    s = 'fail' if labels[instance] else 'pass'
```

```
    label = exp.available_labels()[0]
    expl = exp.as_list(label=label)
    fig = plt.figure(facecolor='white')
    vals = [x[1] for x in expl]
    names = [x[0] for x in expl]
    vals.reverse()
    names.reverse()
    colors = ['green' if x > 0 else 'red' for x in vals]
    pos = np.arange(len(expl)) + .5
    plt.barh(pos, vals, align='center', color=colors)
    plt.yticks(pos, names)
    prediction =
loaded_model.predict(np.array(features.iloc[instance]).reshape(1,250))
[0][0]
    prediction = np.round(1-prediction, 2)
    print("Student #: ", instance)
    print("Ground Truth Model Prediction: ", 1-labels[instance], "-",
s)
    print("Black Box Model Prediction: ", prediction, "-", 'pass' if
prediction > 0.5 else 'fail')

# YOUR TURN: Choose a student to explain (by index #). Note that there
are 8,769 students.
instance = ...

# We call the explainer on a student instance.
exp = explainer.explain_instance(features.iloc[instance], predict_fn,
num_features=10)

# YOUR TURN: Plot the LIME results
plot_lime(...)

send(plt, 1)

lime_explanation = """

Write your interpretation here

"""

send(lime_explanation, 2)
```