Enhancing AI Equity in Education: Post Implementation Development

Overview

In this notebook, we delve into crucial aspects of AI application in education, focusing on explainability, customer feedback and UX testing, and post-launch assessment. Our goal is to ensure that AI tools not only support educational objectives but also do so transparently and equitably, incorporating user feedback and continuously monitoring performance and bias.

Topics Covered

1. **AI Explainability, Literacy & Education**: Understanding the "why" behind AI predictions is crucial for trust and usability in educational settings. We'll explore tools and techniques for enhancing AI explainability.
2. **Customer Feedback and UX Testing**: Direct input from users is invaluable. We'll examine how to collect and use feedback to improve AI tools in education.
3. **Post-Launch Assessment**: After deployment, the work isn't done. We'll cover strategies for monitoring AI tools to detect any performance issues or bias, ensuring they remain effective and fair over time.

In [1]:
```
import sys
!{sys.executable} -m pip install shap
```

```
Collecting shap
  Downloading shap-0.44.1-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux
_2_17_x86_64.manylinux2014_x86_64.whl.metadata (24 kB)
Requirement already satisfied: numpy in /home/nukpezah/anaconda3/lib/python3.9/site-pack
ages (from shap) (1.24.3)
Requirement already satisfied: scipy in /home/nukpezah/anaconda3/lib/python3.9/site-pack
ages (from shap) (1.11.3)
Requirement already satisfied: scikit-learn in /home/nukpezah/anaconda3/lib/python3.9/si
te-packages (from shap) (1.3.0)
Requirement already satisfied: pandas in /home/nukpezah/anaconda3/lib/python3.9/site-pac
kages (from shap) (1.4.4)
Requirement already satisfied: tqdm>=4.27.0 in /home/nukpezah/anaconda3/lib/python3.9/si
te-packages (from shap) (4.65.0)
Requirement already satisfied: packaging>20.9 in /home/nukpezah/anaconda3/lib/python3.9/
site-packages (from shap) (23.1)
Collecting slicer==0.0.7 (from shap)
  Downloading slicer-0.0.7-py3-none-any.whl (14 kB)
Requirement already satisfied: numba in /home/nukpezah/anaconda3/lib/python3.9/site-pack
ages (from shap) (0.58.0)
Requirement already satisfied: cloudpickle in /home/nukpezah/anaconda3/lib/python3.9/sit
e-packages (from shap) (2.2.1)
Requirement already satisfied: llvmlite<0.42,>=0.41.0dev0 in /home/nukpezah/anaconda3/li
b/python3.9/site-packages (from numba->shap) (0.41.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /home/nukpezah/anaconda3/lib/py
thon3.9/site-packages (from pandas->shap) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /home/nukpezah/anaconda3/lib/python3.9/si
te-packages (from pandas->shap) (2023.3.post1)
Requirement already satisfied: joblib>=1.1.1 in /home/nukpezah/anaconda3/lib/python3.9/s
ite-packages (from scikit-learn->shap) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /home/nukpezah/anaconda3/lib/pyth
on3.9/site-packages (from scikit-learn->shap) (2.2.0)
Requirement already satisfied: six>=1.5 in /home/nukpezah/anaconda3/lib/python3.9/site-p
ackages (from python-dateutil>=2.8.1->pandas->shap) (1.16.0)
Downloading shap-0.44.1-cp39-cp39-manylinux_2_12_x86_64.manylinux2010_x86_64.manylinux_2
_17_x86_64.manylinux2014_x86_64.whl (535 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 535.6/535.6 kB 3.7 MB/s eta 0:00:00[31m3.9 M
B/s eta 0:00:01
Installing collected packages: slicer, shap
Successfully installed shap-0.44.1 slicer-0.0.7
```

In [2]:
```python
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.datasets import make_classification
import shap  # For explainability
import matplotlib.pyplot as plt
```

In [3]:
```python
# Generate a synthetic dataset: features could represent student demographics, perfo
X, y = make_classification(n_samples=1000, n_features=20, n_informative=15, n_redundant=

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42

print("Synthetic dataset generated and split into training and testing sets.")
```

```
Synthetic dataset generated and split into training and testing sets.
```

In [4]:
```python
# Initialize and train a RandomForestClassifier
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Predict on the test set
```

```
y_pred = model.predict(X_test)

# Basic evaluation
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy:.2f}")
```

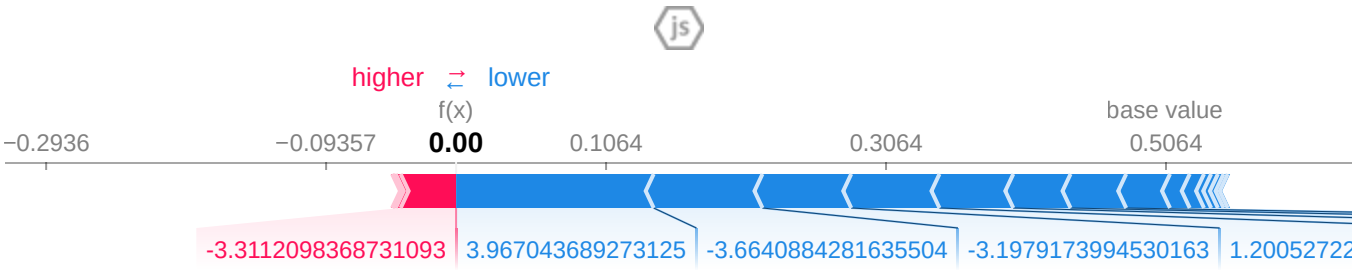
```
Model Accuracy: 0.98
```

In [5]:

```
# Using SHAP for explainability
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)

# Plot the SHAP values for the first prediction
shap.initjs()
shap.force_plot(explainer.expected_value[1], shap_values[1][0,:], X_test[0,:])
```

Out[5]:



Customer Feedback and UX Testing

Gathering and integrating customer feedback is essential for creating user-centered AI tools in education. Tools like surveys, interviews, and usability tests help understand user needs and preferences. Incorporating this feedback into the AI development process ensures the tool is effective and user-friendly.

Tools for Customer Feedback and UX Testing:

- **Surveys and Questionnaires**: Collect quantitative and qualitative data from users.
- **Interviews**: Deep dive into user experiences and expectations.
- **Usability Testing**: Observe real users interacting with the AI tool to identify usability issues.

In [7]:

```
# Performance and Bias Detection
# Assuming y_test has equal representation of classes for simplicity

# Confusion matrix to detect bias in predictions
conf_matrix = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:\n", conf_matrix)

# Performance monitoring could be done through logging and regular evaluation against ne
```

```
Confusion Matrix:
 [[104   0]
 [  4  92]]
```

Monitoring Tools and Resources

Ensuring the long-term success of an AI tool in education requires continuous monitoring for performance, fairness, and user satisfaction. Below are resources and tools that can help practitioners in these areas:

Regular Evaluation

- **Scikit-learn**: A foundational Python library that provides simple and efficient tools for data analysis and modeling, including performance metrics.
  - GitHub: https://github.com/scikit-learn/scikit-learn

Bias Detection

- **AI Fairness 360 (AIF360)**: An extensible open-source library to help detect, understand, and mitigate bias in machine learning models.
  - GitHub: https://github.com/Trusted-AI/AIF360
- **Fairlearn**: A toolkit that aims to empower data scientists and developers to assess and improve the fairness of their AI systems.
  - GitHub: https://github.com/fairlearn/fairlearn

User Feedback Loop

- **Prodigy**: An annotation tool for machine learning and natural language processing tasks. It's an efficient way to improve models based on user feedback, though it's not open source.
  - Website: https://prodi.gy/
- **Paperform**: A versatile form builder that can be used to collect user feedback, with easy integration into websites and apps.
  - Website: https://paperform.co/

While Prodigy and Paperform are not hosted on GitHub and might not be free, they represent practical tools for collecting and incorporating user feedback into the continuous improvement cycle of AI tools. For open-source alternatives, consider integrating GitHub issues or Google Forms as part of your feedback loop strategy.