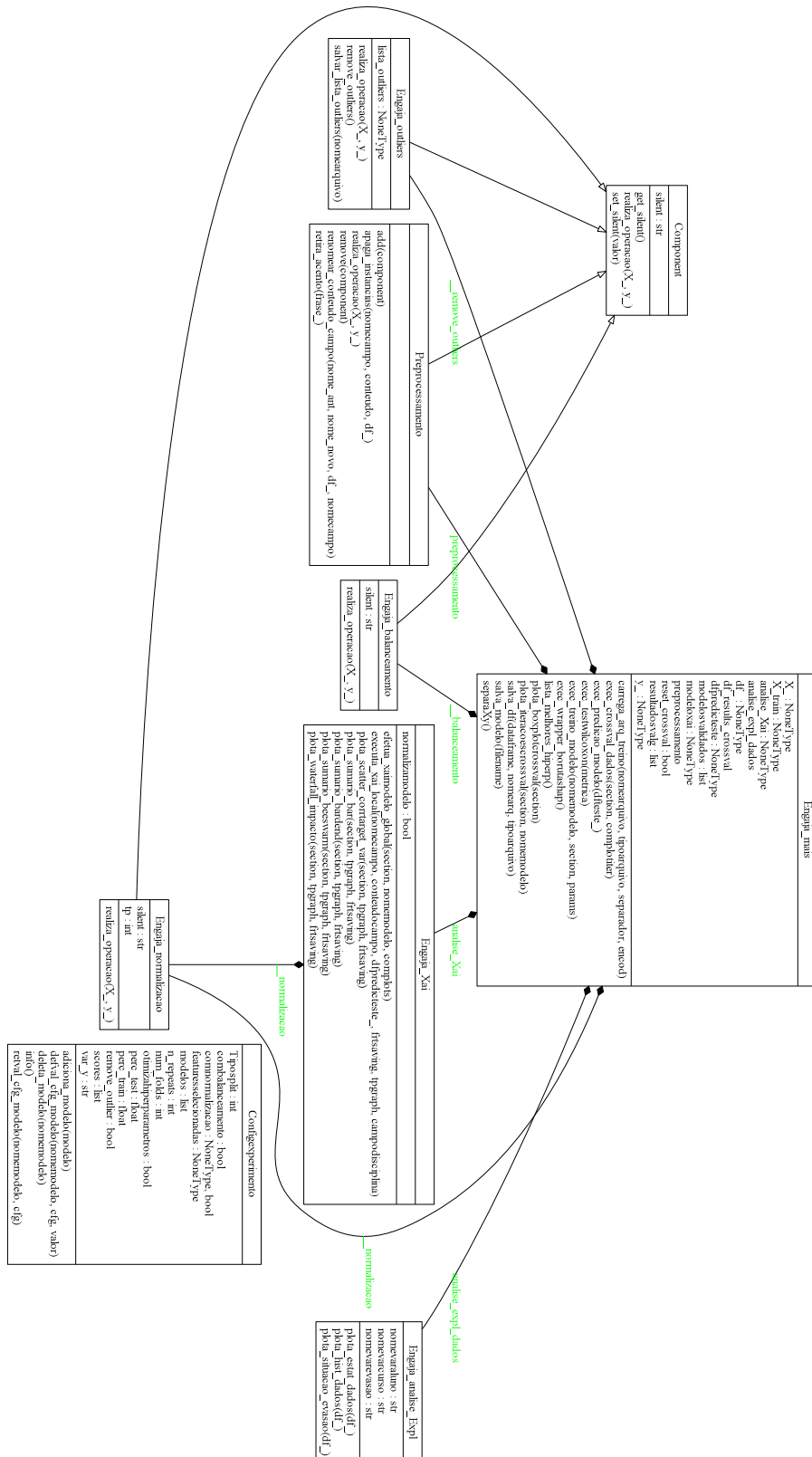


APÊNDICE B – ARQUIVOS DO ARCABOUÇO ENGAJAMAIS NO GITHUB

Nesta seção descrevemos os arquivos do arcabouço disponibilizados no GitHub. O uso do arcabouço para as Fase 1, 2, 3 e 4, é suportado por um conjunto de classes desenvolvido em linguagem Python, cujo o diagrama e as especificações técnicas das principais classes, estão disponíveis no **Apêndice C** e seu código fonte no arquivo “**engajamaisclasses.py**”. O arquivo “**Notebook com exemplo de uso do Engajamais - fases 1_2_3_4.ipynb**” foi criado para exemplificar o uso do arcabouço aplicado à base de dados da UPE. Para a implementação da Fase 5, foi disponibilizado no arquivo “**Apiengajamais_fase_5.zip**”, o projeto completo da API, tendo suas especificações técnicas descritas no **Apêndice D**.

Endereço GitHub dos arquivos do arcabouço: < https://github.com/herbert1703/Engajamais >	
Arquivo	Descrição
engajamaisclasses.py	Arquivo Python com todas as classes que compõem o arcabouço
Notebook com exemplo de uso do Engajamais - fases 1_2_3_4.ipynb	Jupyter Notebook com exemplos de utilização do arcabouço para as Fases 1, 2, 3 e 4
evasao_UPE.csv	Dataset utilizado nos exemplos de utilização do arcabouço
Apiengajamais_fase_5.zip	Projeto da API desenvolvido em Python através da ferramenta PyCharm (Fase 5)
Documentação Arcabouço Engajamais.pdf	Documentação composta pela descrição dos arquivos do arcabouço, diagrama de classes (Fases 1 a 4), especificações técnicas das principais classes e especificações técnicas dos <i>endpoints</i> da API desenvolvida (Fase 5)
Comando SQL - AVA do projeto engajamais.sql	Arquivos com instruções SQL para criação da tabela base (que contém informações do curso, disciplina e do aluno), criação de índices e comando <i>select</i> para consulta das interações do AVA-MOODLE



C.2 ESPECIFICAÇÕES TÉCNICAS DAS PRINCIPAIS CLASSES

Principais classes do arcabouço Engjamais		
Configexperimento (Classe Singleton para configurações gerais do arcabouço)		
Nome componente	Tipo	Descrição
Tiposplit	prop	Define o tipo de split utilizado para separação dos dados: 1 = RepeatedStratifiedKFold ou 2 = StratifiedKFold
combalanceamento	prop	Define balanceamento dos dados antes do treinamento
comnormalizacao	prop	Define normalização dos dados. Esta propriedade é alterada automaticamente pela configuração de cada classificador
featuresselecionadas	prop	Define lista de variáveis que serão utilizadas pelo arcabouço para treinamento e testes
modelos	prop	Define lista com informações sobre os classificadores utilizados (nome, classificador, parâmetros, normalização, params_shap, etc)
n_repeats/num_folds	prop	Define respectivamente, o número de repetições para Tiposplit = 1 e o número de kolds para o treinamento
otimizahiperparametros	prop	Define otimização de hiperparâmetro no treinamento
perc_train/perc_test	prop	Define respectivamente, o percentual de separação dos dados em treino e teste para cada uma das 30 iterações realizadas
remove_outlier	prop	Define remoção de outliers detectados
scores	prop	Define a lista de métricas utilizadas para avaliação
var_y	prop	Define o nome da variável "target"
adiciona_modelo	método	Adiciona classificadores na lista de treinamento e teste
defval_cfg_modelo	método	Altera configurações dos classificadores adicionados à lista de treinamento
deleta_modelo	método	Retira um classificador específico da lista de treinamento
info	método	Lista todas as configurações definidas para o arcabouço
retval_cfg_modelo	método	Recupera uma configuração do classificador especificado
Engaja_mais (Classe principal do arcabouço para executar préprocessamento, cross_val, treino e teste dos modelos, além dos recursos de XAI)		
Nome componente	Tipo	Descrição
X_ e y_	prop	Respectivamente representam o conjunto de dados para treinamento, validação e testes (X_) e a variável target (y_)
X_train	prop	Conjunto de dados utilizado para treinamento final do modelo escolhido
df_	prop	Conjunto de dados carregado pela função "carrega_arq_treino"
analise_Xai	prop	Instância da classe "Engaja_Xai", que disponibiliza funções SHAP para análises de importância e impacto local/global das variáveis do modelo escolhido
analise_expl_dados	prop	Instância da classe "Engaja_analise_Expl", que disponibiliza funções para obtenção de informações estatísticas sobre o conjunto de dados (df_) importados
dfpredicteste	prop	DataFrame com os dados (dfteste_) e o resultado das previsões realizadas pela função "exec_predicao_modelo"
modelosvalidados	prop	Lista dos modelos que passaram por validação cruzada na instância desta classe
modeloxai	prop	Modelo treinado através da função "exec_treino_modelo"
preprocessamento	prop	Instância da classe composta chamada "Preprocessamento" que possui como padrão a composição para execução das classes "Engaja_outliers", "Engaja_balanceamento" e "Engaja_normalizacao". Pode-se a critério, adicionar outras classes de interesse.
reset_crossval	prop	Define se a lista de resultados da validação cruzada, realizada através da função "exec_crossval_dados", será reiniciada ou acrescentará apenas os classificadores que ainda não foram validados
resultadosvalg	prop	Lista com os resultados das métricas dos modelos validados
df_results_crossval	prop	Dataframe com os resultados de cada métrica da validação cruzada por modelo (Coluna 0 = resultado do crossval, coluna 1 = resultado do teste realizado nos melhores estimadores encontrados em cada uma das 30 iterações propostas)
carrega_arq_treino	método	Carrega dataset para execução do arcabouço (aceita xlsx e csv)
exec_crossval_dados	método	Executa validação cruzada de acordo com as configurações definidas na classe "Configexperimento"
exec_testwilcoxon	método	Executa teste de wilcoxon com os resultados obtidos através da função "exec_crossval_dados"
lista_melhores_hiperp	método	Lista a frequência dos hiperparâmetros mais utilizados na validação cruzada
exec_treino_modelo	método	Realiza treinamento de um classificador específico com os dados carregados pela função "carrega_arq_treino"
exec_wrapper_borutashap	método	Executa o método borutaSHAP para avaliação da importância de variáveis para o modelo treinado (modeloxai) pela função "exec_treino_modelo"
exec_predicao_modelo	método	Realiza previsões com o modelo treinado, sendo os dados passados como parâmetros para esta função
plota_boxplotcrossval	método	Plota boxplot dos resultados da validação cruzada. F1-score é a métrica utilizada neste gráfico
plota_iteracoescrossval	método	Plota um gráfico de tendências para cada um dos resultados de treinamento e teste
salva_df	método	Salva um DataFrame em arquivo (Formatos: xlsx ou csv)
salva_modelo	método	Salva o modelo treinado (modeloxai) em um arquivo do tipo pickle e o arquivo X_train.csv, utilizado na API para gerar os gráficos de explicabilidade
SeparaXy	método	Separa o conjunto de dados em X_ e y_. Esta função pode ser utilizada após alguma alteração no dataframe df

APÊNDICE D – ESPECIFICAÇÕES TÉCNICAS DA API PROPOSTA NA FASE 5

Métodos (endpoints) da API python Engajamais				
Endpoint	Método	Conteúdo	Content-type envio	Content-type resposta
/verifica_api_online	GET	N/A	N/A	String
/predevasao	POST	[{}.{...}]	application/json	application/json
/summarypng/<job_id>	POST	[{}.{...}]	application/json	application/json
/status/<job_id>	GET	N/A	N/A	application/json
/recuperasummary/<job_id>	GET	N/A	N/A	image/png
/apagasesummary/<job_id>	DELETE	N/A	N/A	application/json
/plotforcewall.png	POST	[{}]	application/json	image/png
/plotforce.png	POST	[{}]	application/json	image/png
Arquivos para predição e XAI				
Descrição				
Arquivo com o modelo preditivo criado na Fase 4 do arcalbouço				
Arquivo que foi utilizado no treinamento do modelo para geração dos gráficos SHAP				
Arquivo com informações sobre os pacotes python necessários para uso da API				
Requisitos para o MySQL ou MariaDB				
Descrição				
MySQL INFO = { "sql_host": "255.255.255.255", "sql_uname": "xyz", "sql_password": "xyz", "sql_db": "xyz" }		Informações sobre a conexão com o banco de dados para armazenamento de informações sobre a execução das jobs		
CREATE TABLE 'job_status_tracker' ('uid' varchar(40) COLLATE utf8_unicode_ci NOT NULL, 'job_status' varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL, PRIMARY KEY ('uid')) ENGINE=InnoDB AUTO_INCREMENT=14706 DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci		Instrução para criação da tabela que servirá para realizar o controle de execução das jobs		