Group 31 – Machine Learning Assignment[i]

1. Description of computational learning experiments

1.1 Feature engineering[ii]

Our journey started with EDA. After filling all *missing values*, we *removed outliers*[iii]. We then feature engineered some variables by creating new ones and making sure they were all numerical[1]. Finally, we *scaled* the numerical variables. From 24 variables, we then dropped irrelevant ones, checked correlations with the target variable with *permutation feature importance* (after running some models), and selected the most important features. The final variables were: **CurrentSessionLength, Day, Time, StandardizedProgression, ResponseValue_mean, ResponseValue_median, ResponseValue_std**. An alternative pre-processing (mode, mean and one hot encoding) was conducted but had worse performances.

Feature	Data Type	Problem	Action
UserID	Text	Multiple entries per user	- Remove "p" to retain only numerical part, converting the data type to numeric - Group by this feature to compute statistics for other features (CurrentSessionLength, LevelProgressionAmount, ResponseValue)
QuestionTiming	Binary	Imbalanced	- Label encode - undersample the majority class
TimeUtc	Text	Too much information in a single value	- Convert data type to "datetime" - Split into two features: Day and Time - Day: From Monday to Sunday, lebal encoded and standardized - Time: we tried to make this feature balanced distributed, and cluster values into 3 groups: night, daytime, and evening, lebal encoded and standardized
CurrentGameMode	Categorical	Extremely imbalanced	Drop
CurrentTask	Text	- Many missing values - Too many classes	- According to the meaning of the feature, a missing value means that a user is not playing a game, therefore, we filled the missing values with "Not_Playing" - We created a list "special_tasks" for similar classes, and then clustered the values into 3 groups: 0 (Not_Playing), 1 (other), and 2 (special_task)
CurrentSessionLength	Numeric	Imbalanced: extremely right skewed	- Log transformation was ineffective; calculated statistics instead: mean, median, standard deviation, maximum
LastTaskCompleted	Text	- Too many missing values - Too namy classes	- According to the meaning of the feature, a missing value means that a user hadn't saved or finished any game, therefore, we filled the missing values with "Not_Saved" - We used the list "special_tasks" here again, and then clustered the values into 3 groups: 0 (Not_Saved), 1 (other), and 2 (special_task)
LevelProgressionAmount	Numeric	- Many missing values - Imbalanced: according to bar plot, there are too many 0s and 1s	- According to the meaning of the feature, a missing value means that a user hadn't started the game, therefore, we filled the missing values with 0 - We standardized and calculated statistics: mean, median, standard deviation, maximum
QuestionType	Categorical	Only contains a single class	Drop
ResponseValue	Numeric	- Imbalanced: left skewed	- Important to learn from target variable due to time series nature - Calculated statistics: mean, median, standard deviation

1.2 Learning algorithm(s)[iv]

The algorithms that performed the best were the following ones. We tested the training data on 10 models and selected the 4 best-performing ones. RNN and Linear Regression had a MSE > 100.

MODEL	MSE	MSE WITH KFOLD	HYPERPARMETERS
Random Forest	83,2151	83,1054	{'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 300}
LGBM	82,9705	81,3959	Best parameters: {'learning_rate': 0.1, 'n_estimators': 500, 'num_leaves': 100},
XGBoost	83,4048	82,0874	Best parameters: {'learning_rate': 0.1, 'max_depth': 10, 'n_estimators': 100, 'subsample': 0.8}
CatBoost	82,1456	81,0609	Best parameters: {'depth': 10, 'iterations': 500, 'learning_rate': 0.1}

1.3 Hyperparameter tuning

We performed hyperparameter tuning using *GridSearchCV*[v] with a *StratifiedKFold* cross-validation. The best parameters were determined based on the mean absolute error scoring metric. The final model, trained with these optimal parameters, achieved a MAE [81.06] on the validation set.

1.4 Solution Performance[vi]

Our most successful submission (MAE: 107, 859 on test set) used the highest performing learning algorithm (**CatBoost**)[vii] with stratified k-fold (k = 10) to tackle class imbalances. We divided the testing dataset between users that were present also in the training one (**shared_users_data**), from the new users that appeared in the test dataset for the first time (**new_users_data**). We applied CatBoost on the shared users[viii], then filled the new users' ResponseValue with the *mean* of the predicted score. Finally, we adjusted the index. Failed attempts involved using SMOGN to augment the data in order to tackle class imbalances (MAE: 108,188) while using KNN to avoid overfitting[ix], or adding some more variables that could have been relevant according to permutation feature importance (108,804).[x]

---

[1] Source: Stackoverflow.

2. Name on Codalab: YueW

3. Task Division

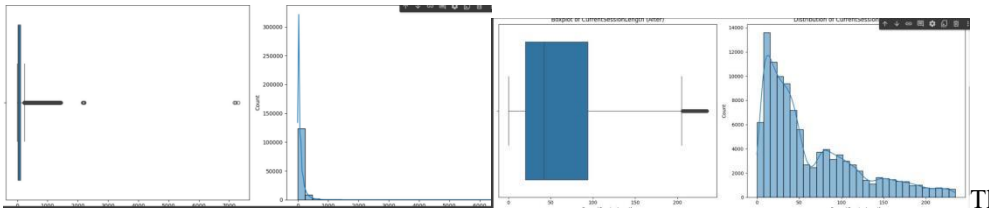
Group	Members	Snr.	Task
31	Angela Barone	2128404	Preprocessing (outliers, missing values), feature engineering (scaling), hyperparameter tuning, permutation feature importance, CatBoost, idea about separating shared users/new users + LGBM, XGBoost, SMOGN (not used)
	Evangelia Santorinaiou	2134847	Feature engineering (encoding, not used), Random Forest, Hyperparameter tuning, code about separating shared users from new ones, augmentation (not used)
	Yue Wang	2111066	Feature engineering (feature importance), random forest, logistic regression, SVR, GradientBoosting, SGDR, MLPR (not used), hyperparameter tuning (Random Forest), test dataset processing (ensure data order), cleaned the final code for submission
	Feysal Reka	2010916	RNN coding (not used), preprocessing with mode, median and one hot encoding (not used). The unused codes were submitted less than 3 days away from the deadline and they were not pertinent with what the rest of the group was trying to achieve.

4. References or appendices

---

[i] From our analysis, the experiment used a game (probably "PowerWashSimulator") in order to trigger some emotional changes in the users, as washing items and listening to the sound of water would presumably stimulate positive feelings. The changes in mood were assessed with some questions ("Wellbeing" status in the case of the training and test data) and calculated with a score ("Response Value"). The game contained two game modes, career mode or free play (9 special tasks).

[ii] Extra sources for the code and FE ideas could be found in this video

[iii]  This is an example of how the removal of outliers affected one of the most important features: CurrentSessionLength. The code was found and adapted from Stackoverflow.

[iv] LGBM, XGBoost and CatBoost (along with the codes) were studied from this video.

[v] Together with permutation feature importance, the codes were available from the labs of other exams (Data Mining, Spatiotemporal Data Analysis).

[vi] As only three submissions were left after our first successful attempt, we decided not to use the supplemental data and focus on optimizing our solid result. If we had more attempts left, we would have used the supplemental data as many shared users with both datasets were present in the file.

[vii] "*As a Decision Tree based algorithm, CatBoost is well-suited to machine learning tasks involving categorical, heterogeneous data*" Hancock, J.T., Khoshgoftaar, T.M. CatBoost for big data: an interdisciplinary review. *J Big Data* **7**, 94 (2020). https://doi.org/10.1186/s40537-020-00369-8

[viii] This is the source we used to identify the new and shared users between the two datasets: Python | set() Function - GeeksforGeeks

[ix] As it was extremely computationally expensive (more than 14 hours), it was only tried once. SMOGN was used to augment the data while also taking care of unbalanced classes

[x] Feature ablation was then performed.