WIH3001 Data Science Project

Influential Factors Identification in Alcohol Addiction of Adolescents using RELIEF

Presented by:

Chim Tingshing

U2005412/1

Supervised by:

Associate Prof. Ts. Dr. Sri Devi

A/P Ravana

Overview

- Introduction
- Literature Review
- Problem Statement
- Objectives
- Data Science Methodology
- Conclusion
- References
- Demonstration

Introduction

Alcohol Addiction

- In Malaysia, 11.8% of adults aged 18 and above drink alcohol, and 1 in 10 of those adults practice heavy episodic drinking. (National Health Morbidity Survey, 2019)
- Higher rates of binge drinking equates higher risk of developing alcohol addiction (Gowin et al., 2017)
- A study done on secondary school students in Miri, Sarawak showed that 42.2% of adolescents had problematic alcohol use (Chhoa et al., 2019)

Introduction

Diagnosing Alcohol Addiction

Traditional

- Identified using AUDIT
- Formal diagnosis using DSM-5
- Require discussion and answering questions regarding symptoms

Machine Learning

- Using machine learning algorithms to classify Alcohol Addiction
- Process large amounts of data in a short time
- Allow medical professionals to give immediate treatment

Literature Review

Author	Aim	Dataset	Feature extraction method	Model	Result	Knowledge Gap
Palaniappan et al. (2017)	- To predict alcohol addiction among the secondary school students given a set of parameters or factors leading to alcohol addiction.	- Survey data from Portugal schools for Maths and Portuguese class	- Preselected 15 features as input features from 33 features	- MLP - AutoMLP	- Accuracy of 61.78% MLP and 64.54% from AutoMLP	- Manual selection of features
Ismail et al. (2018)	- To predict alcohol consumption behaviors among secondary school students via a classification experiment	- Survey data from Portugal schools for Maths and Portuguese class	- None	- Decision Tree - Random Forest - KNN - Naïve Bayes	- Random forest showed highest weighted mean accuracy of 98.77%, weighted mean recall of 47.07% and weighted mean precision of 58.99%	- No feature selection methods used at all
Arshad et al. (2015)	- To study the contributing factors and the effects of alcoholism among youth.	- Questionnaire answer from 150 youths aged 15 – 24 sampled using snowball technique	- N/A	- N/A	- Positive relationship between interpersonal problems, availability of alcohol and lack of knowledge of alcohol with alcoholism	- No modelling to predict addiction based on significant features obtained

Problem Statement

01.

Limited to manual feature selection in most alcohol addiction studies

- Palaniappan et al. (2017) manually preselected 15 features from 33 features as inputs without using any feature selection techniques.

02.

Limited to flat machine learning classification for alchol addiction studies

- Ismail et al. (2018) trained machine learning models to predict alcohol consumption behaviour but directly on the entire dataset using flat classification

Objectives

1

The First Objective

To obtain the influential factors that cause alcohol addiction among adolescents using RELIEF based algorithms

2

The Second Objective

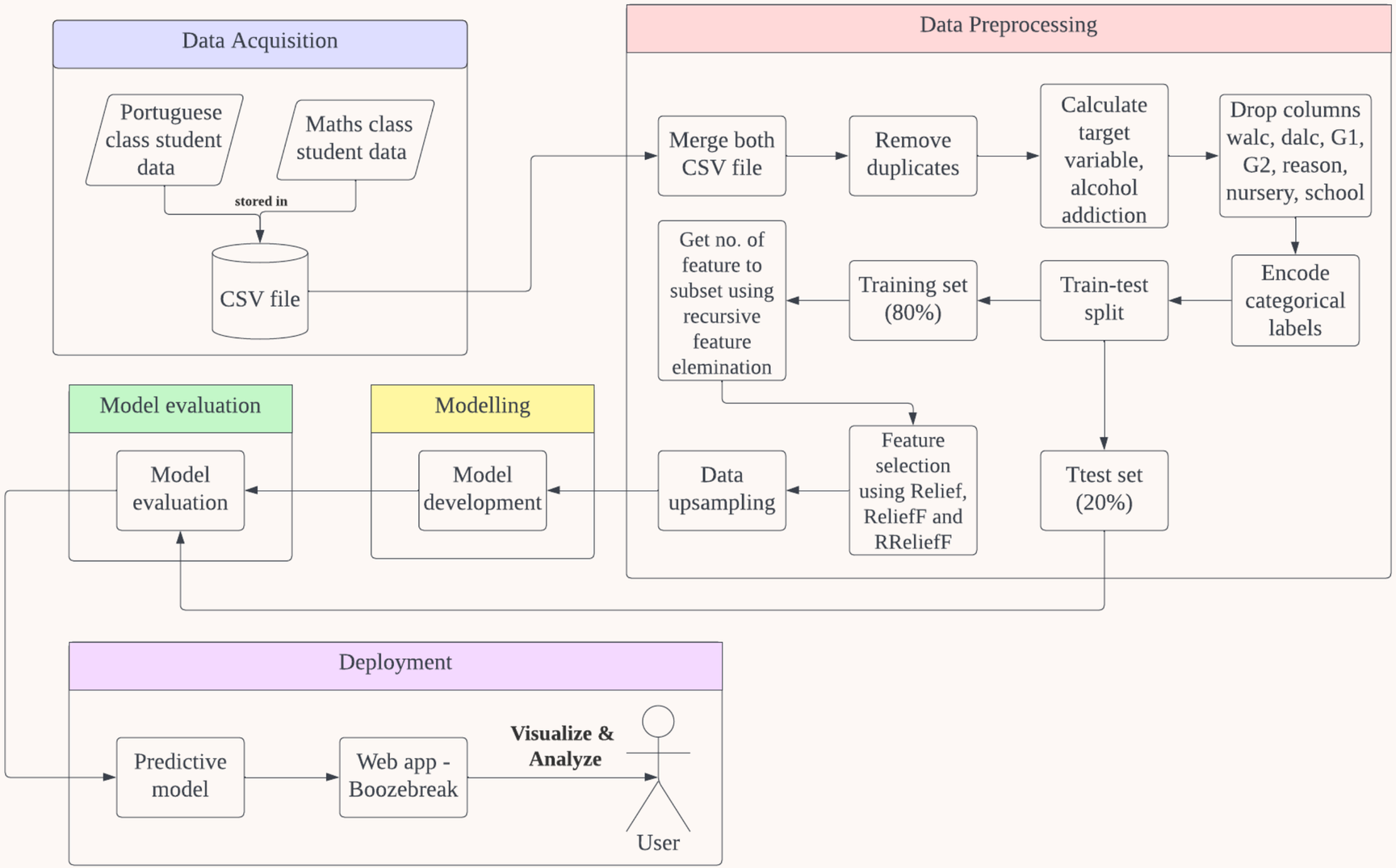
To develop classification models to classify alcohol addiction among adolescents using the obtained influential factors

3

The Third Objective

To assess and report the results of the classification models using evaluation metrics

Data Science Methodology



Data Acquisition

kaggle

- "Student Alcohol Consumption" from Kaggle
- Originally composed by P. Cortez and A. Silva (P. Cortez & A. Silva, 2006)
- Contains 33 attributes and consists of questionnaire data from secondary school students
- 395 students from the mathematics class and 649 students from the Portuguese language class

Data Preprocessing

Merging both datasets

```
df_mat = pd.read_csv("student-mat.csv")
df_por = pd.read_csv("student-por.csv")
df_merge = pd.merge(df_mat, df_por, how = "outer")

print("No. of students in Maths class:", df_mat.shape[0])
print("No. of students in Portuguese class:", df_por.shape[0])
print("No. of students after merging:", df_merge.shape[0])
```

```
No. of students in Maths class: 395
No. of students in Portuguese class: 649
No. of students after merging: 1044
```

Removing duplicates

Remove duplicates based on all attributes except results in the class and extra class for the subject

```
df_merge = df_merge.drop_duplicates(subset = subset, keep = 'first')
print("No of unique students after removing duplicates:", df_merge.shape[0])
```

```
No. of unique students after removing duplicates: 959
```

Data Preprocessing

Calculating target variable

- Calculate addiction based on formula below:

addiction = [(Walc × 2) + (Dalc × 5)] / 7

Walc = weekend alcohol consumption

Dalc = weekday alcohol consumption

- Map to binary outcome to measure alcohol addiction

addiction > 3 = 1 = addicted to alcohol

aaddiction ⩽ 3 = 0 = not addicted to alcohol

Dropping attributes

- Drop attributes Walc and Dalc
- Drop other attributes G1, G2, nursery, reason, school

```
df_merge = df_merge.drop(['G1', 'G2', 'Walc', 'Dalc',
                          'reason', 'school', 'nursery'],
                          axis = 1)
```

Exploratory Data Analysis (EDA)

Overview

Dataset Statistics

Number of Variables	27
Number of Rows	959
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	891.5 KB
Average Row Size in Memory	951.9 B
Variable Types	Categorical: 25 Numerical: 2

Dataset Insights

`absences` is skewed	Skewed
`G3` is skewed	Skewed
`sex` has constant length 1	Constant Length
`age` has constant length 2	Constant Length
`address` has constant length 1	Constant Length
`famsize` has constant length 3	Constant Length
`Pstatus` has constant length 1	Constant Length
`Medu` has constant length 1	Constant Length
`Fedu` has constant length 1	Constant Length
`traveltime` has constant length 1	Constant Length

1 2

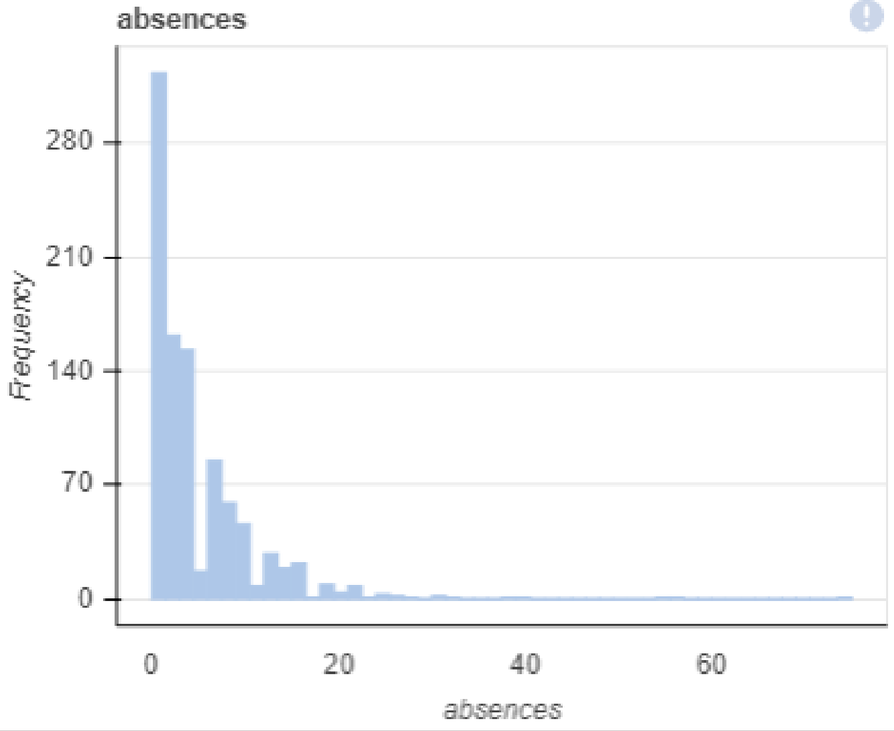
Exploratory Data Analysis (EDA)

absences
numerical

Show Details

Approximate Distinct Count	35
Approximate Unique (%)	3.6%
Missing	0
Missing (%)	0.0%
Infinite	0
Infinite (%)	0.0%
Memory Size	15344

Mean	4.6601
Minimum	0
Maximum	75
Zeros	308
Zeros (%)	32.1%
Negatives	0
Negatives (%)	0.0%

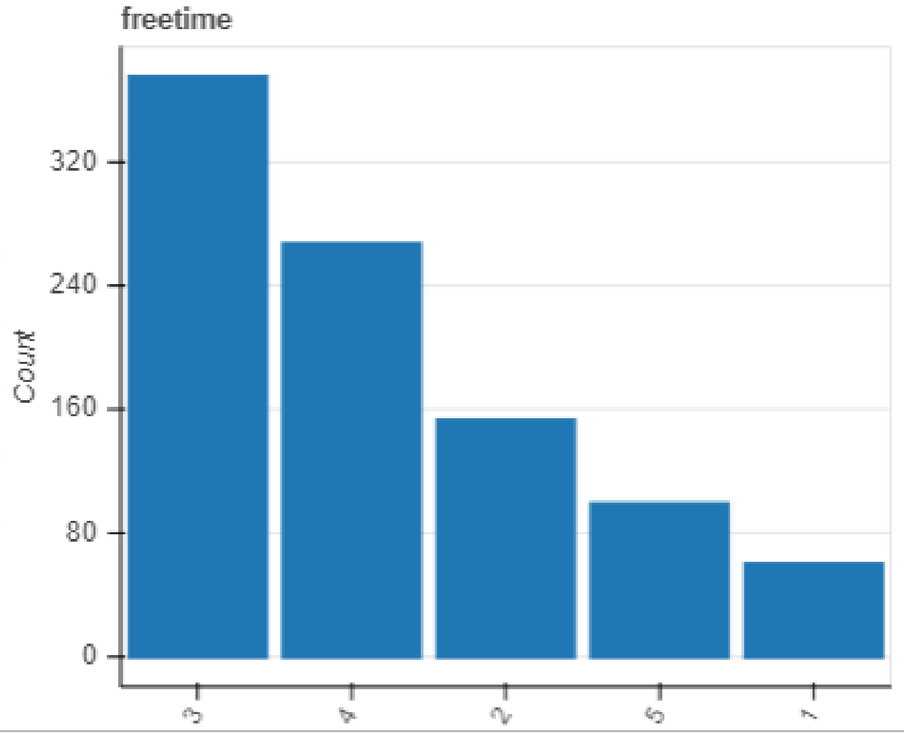


Exploratory Data Analysis (EDA)

freetime
categorical

Show Details

Approximate Distinct Count	5
Approximate Unique (%)	0.5%
Missing	0
Missing (%)	0.0%
Memory Size	63294



Data Preprocessing

Encoding categorical variables

- Machine learning models cannot handle categorical data, so transformation to numerical format is needed

```
# define category order
famsize_category = ['LE3', 'GT3']

# column transformer to encode data
column_transformer = ColumnTransformer(
    [('categorical', OrdinalEncoder(), ['sex', 'address', 'Pstatus', 'Mjob',
                                        'Fjob', 'guardian', 'schoolsup',
                                        'famsup', 'paid', 'activities',
                                        'higher', 'internet', 'romantic']),
    ('nominal', OrdinalEncoder(categories = [famsize_category]), ['famsize'])],
    remainder = 'passthrough'
)

column_transformer.set_output(transform="pandas")

df = column_transformer.fit_transform(df_merge)
```

Data Preprocessing

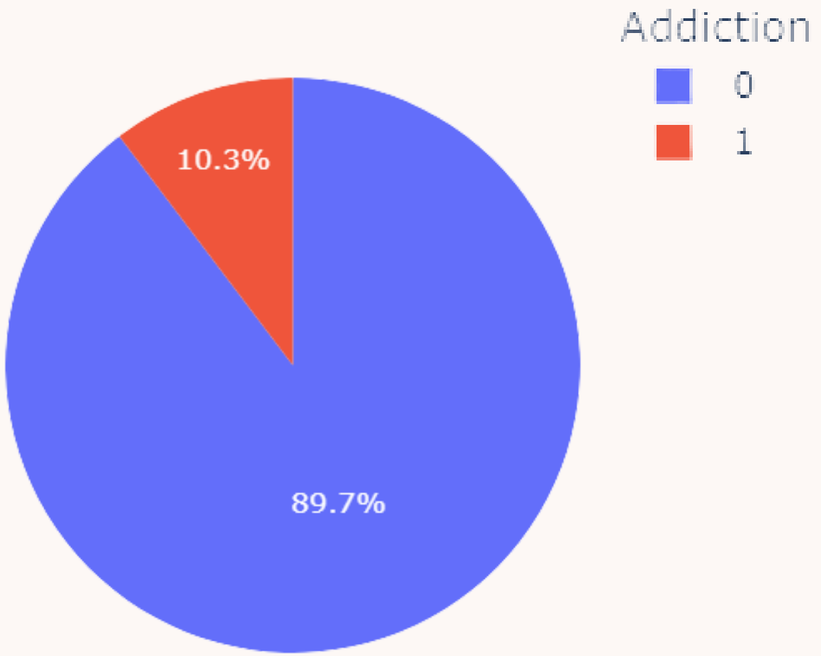
Splitting data

- Training and testing set with ratio of 8:2
- Stratified sampling to ensure equal representation of target variable, addiction in training and testing set due to imbalanced dataset

```
print("X_train shape:", X_train.shape)
print("X_test shape:", X_test.shape)
print("y_train shape:", y_train.shape)
print("y_test shape:", y_test.shape)
```

```
X_train shape: (767, 26)
X_test shape: (192, 26)
y_train shape: (767,)
y_test shape: (192,)
```

Imbalanced dataset



Data Preprocessing

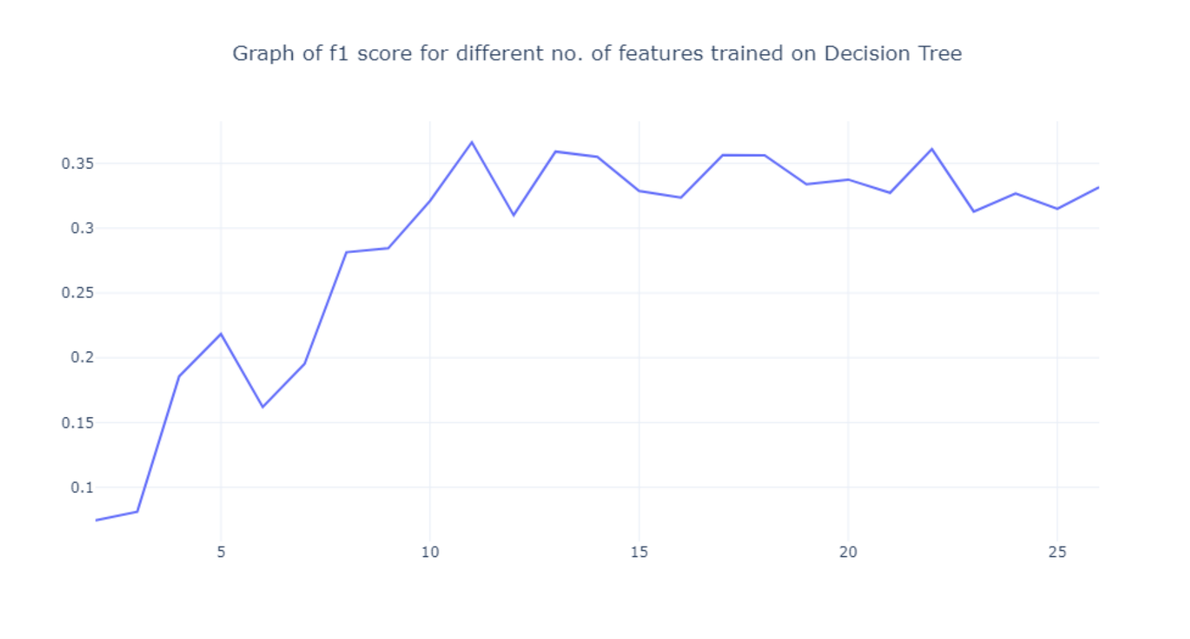
Get no. of feature to subset

- Using recursive feature elimination (RFE) fit on a decision tree
- Scoring using F1-score due to imbalanced dataset

```
estimator = DecisionTreeClassifier(random_state = 42)
cv = StratifiedKFold(n_splits = 5)
results = []

for i in range(2, 27):
    rfe = RFE(estimator = estimator, n_features_to_select = i)
    rfe.fit(X_train, y_train)
    X = X_train.iloc[:, rfe.support_]
    model = DecisionTreeClassifier(random_state = 42)
    f1 = cross_val_score(model, X, y_train, scoring = 'f1', cv = cv)
    results.append(round(f1.mean(), 4))
```

Results



Data Preprocessing

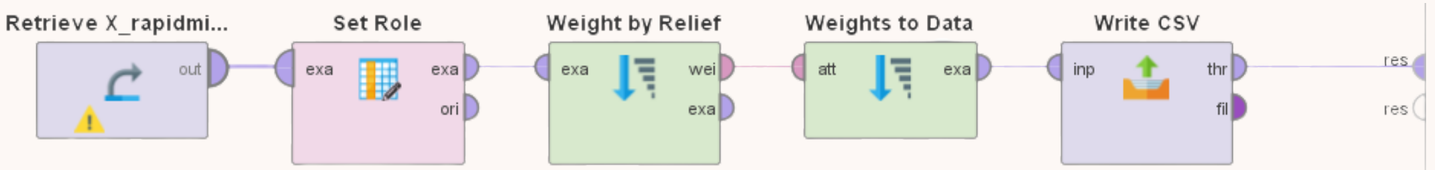
Feature Selection

- sklearn_relief in Python for Relief and RReliefF
- RapidMiner for ReliefF

Relief

```
relief = sr.Relief(n_features = 11, categorical = cat_index,
                   n_iterations = 200, random_state = 42)
relief_arr = relief.fit_transform(X_train.to_numpy(), y_train.to_numpy())
```

ReliefF



RReliefF

```
RReliefF = sr.RReliefF(n_features = 11, categorical = cat_index,
                       n_iterations = 200, random_state = 42)
RReliefF_arr = RReliefF.fit_transform(X_train.to_numpy(), y_train.to_numpy())
```

Data Preprocessing

Relief		ReliefF		RReliefF	
G3	1.065	sex	0.096	failures	0.389
age	0.78	higher	0.075	traveltime	0.337
absences	0.69	guardian	0.072	famsup	0.32
goout	0.635	address	0.065	age	0.291
health	0.585	failures	0.062	address	0.286
Fedu	0.58	famsup	0.036	Medu	0.28
Medu	0.575	paid	0.035	freetime	0.263
freetime	0.56	famsize	0.031	higher	0.257
Mjob	0.52	romantic	0.027	famrel	0.251
famrel	0.475	Pstatus	0.026	Fedu	0.194
Fjob	0.465	famrel	0.025	sex	0.149
studytime	0.46	Fjob	0.025	studytime	0.143
sex	0.43	traveltime	0.02	goout	0.091
famsup	0.43	activities	0.018	guardian	0.074
traveltime	0.405	goout	0.018	internet	0.074
activities	0.31	Medu	0.015	Mjob	0.074
guardian	0.295	Fedu	0.014	health	0.034
romantic	0.27	health	0.014	famsize	0.023
famsize	0.26	age	0.013	schoolsup	0.006
address	0.255	freetime	0.003	romantic	-0.029
paid	0.255	studytime	0.003	paid	-0.04
failures	0.215	G3	0.001	Pstatus	-0.04
internet	0.195	absences	0	activities	-0.057
schoolsup	0.16	internet	-0.011	Fjob	-0.097
Pstatus	0.16	Mjob	-0.019	G3	-0.297
higher	0.08	schoolsup	-0.044	absences	-1.143

Note: Highlighted attributes were selected for model training

Data Preprocessing

Splitting and Upsampling

Baseline set (All attributes)

```
smote = SMOTE(random_state = 42)
X_train_up, y_train_up = smote.fit_resample(X_train, y_train)
```

Relief set

```
# define X_train and X_test for RELIEF based on selected features
X_train_relief = X_train[relief_mapping['original_name']]
X_test_relief = X_test[relief_mapping['original_name']]
```

```
smote = SMOTE(random_state = 42)
X_train_relief, y_train_relief = smote.fit_resample(X_train_relief, y_train)
```

ReliefF set

```
# define X_train and X_test for RELIEFF based on selected features
X_train_reliefF = X_train[reliefF_mapping['Attribute']]
X_test_reliefF = X_test[reliefF_mapping['Attribute']]
```

```
smote = SMOTE(random_state = 42)
X_train_reliefF, y_train_reliefF = smote.fit_resample(X_train_reliefF, y_train)
```

RReliefF set

```
# define X_train and X_test for RRELIEFF based on selected feature
X_train_RReliefF = X_train[RReliefF_mapping['original_name']]
X_test_RReliefF = X_test[RReliefF_mapping['original_name']]
```

```
smote = SMOTE(random_state = 42)
X_train_RReliefF, y_train_RReliefF = smote.fit_resample(X_train_RReliefF, y_train)
```

Modelling

Phase One

Train four machine learning models

- Logistic Regression
- Random Forest
- Support Vector Machine (SVM)
- K-nearest-neighbor (KNN)

on all four sets using default hyperparameters

Phase Two

- Tune hyperparameters
- Run GridSearchCV for all machine learning models using Relief, ReliefF and RReliefF sets

Modelling

Hyperparameters List

Default

Algorithms	Parameters
LR	penalty='l2', C=1, solver='lbfgs'
SVM	C=1, kernel='rbf', gamma='scale'
RF	n_estimator=100, criterion='gini', max_depth='None', bootstrap=True, max_features='sqrt'
KNN	n_neighbors=5, weights='uniform', algorithm= 'auto', p=2

Tuned Relief

Algorithms	Parameters
LR	penalty='l1', C=0.1, solver='liblinear'
SVM	C=10, kernel='rbf', gamma=0.1
RF	n_estimator=100, criterion='entropy', max_depth='None', bootstrap=False, max_features='sqrt'
KNN	n_neighbors=2, weights='uniform', algorithm= 'brute', p=1

Tuned ReliefF

Algorithms	Parameters
LR	penalty='l2', C=1, solver='newton-cg'
SVM	C=10, kernel='rbf', gamma='1'
RF	n_estimator=100, criterion='entropy', max_depth='None', bootstrap=True, max_features='sqrt'
KNN	n_neighbors=3, weights='distance', algorithm= 'brute', p=1

Tuned RReliefF

Algorithms	Parameters
LR	penalty='l2', C=0.1, solver='saga'
SVM	C=10, kernel='rbf', gamma='1'
RF	n_estimator=300, criterion='gini', max_depth='None', bootstrap=True, max_features='sqrt'
KNN	n_neighbors=3, weights='distance', algorithm= 'brute', p=1

Evaluation

Table of Evaluation Metrics

F1-Score

Harmonic mean between precision and recall

Algorithm / Type	LR	SVM	RF	KNN
Baseline (D)	0.419	0.364	0.600	0.292
Relief (D)	0.293	0.253	0.588	0.250
ReliefF (D)	0.409	0.380	0.500	0.369
RReliefF (D)	0.386	0.388	0.647	0.358
Relief (T)	0.304	0.222	0.514	0.327
ReliefF (T)	0.409	0.512	0.452	0.457
RReliefF (T)	0.395	0.706	0.706	0.650

Note: D - Default hyperparameters, T - Tuned hyperparameters

Balanced Accuracy

Mean between specificity and recall

Algorithm / Type	LR	SVM	RF	KNN
Baseline (D)	0.741	0.725	0.722	0.662
Relief (D)	0.655	0.612	0.738	0.609
ReliefF (D)	0.818	0.798	0.688	0.704
RReliefF (D)	0.777	0.817	0.766	0.698
Relief (T)	0.663	0.565	0.708	0.639
ReliefF (T)	0.818	0.740	0.663	0.680
RReliefF (T)	0.783	0.794	0.794	0.805

Note: D - Default hyperparameters, T - Tuned hyperparameters

Evaluation

Table of Evaluation Metrics

ROC AUC

Summarizes ROC with a single number

Algorithm / Type	LR	SVM	RF	KNN
Baseline (D)	0.805	0.790	0.944	0.674
Relief (D)	0.653	0.692	0.787	0.654
ReliefF (D)	0.828	0.866	0.855	0.784
RReliefF (D)	0.839	0.849	0.848	0.738
Relief (T)	0.658	0.600	0.794	0.695
ReliefF (T)	0.828	0.788	0.845	0.753
RReliefF (T)	0.836	0.834	0.850	0.824

Note: D - Default hyperparameters, T - Tuned hyperparameters

Evaluation

Discussion

For tuned hyperparameters:

- Best machine learning model is tuned random forest, trained using features selected by RReliefF, with F1-score = 0.706 and balanced accuracy = 0.794, ROC AUC = 0.850

For default hyperparameters:

- RReliefF performed the best with the highest F1-score for 3/4 models, highest balanced accuracy and ROC AUC for half of the models
- ReliefF performed better than better than baseline set, having higher F1-score for half the models and higher balanced accuracy and ROC AUC for 3/4 models
- Relief performed the worst, even when compared to baseline set

Evaluation

Discussion

Why ReliefF and RReliefF performed better than baseline set?

- Feature selection subsets the most influential features
- Less noisy data present
- Improves prediction quality of machine learning models

Why did Relief performed the worst?

- Relief updates weights based on nearest hit and nearest miss
- ReliefF and RReliefF updates weights based on k-nearest-neighbors
- Thus, Relief is more susceptible to noisy data which makes its estimations unreliable (Kononenko, 1994)

Deployment

Web application

- Best performing machine learning model is deployed in a web application named '**BoozeBreak**'
- Developed using Streamlit
- Low-code solution which requires no prior front-end and back-end experience

Stakeholders & Target Users

Education Industry

- Teachers
- Schools

General Public

- Parents / Guardians
- Adolescents

Conclusion

- RELIEF-based feature selection techniques successfully extracted the influential factors
- Trained four different machine learning algorithms to predict alcohol addiction among adolescents
- The best performing model was random forest with tuned hyperparameters that was trained on features selected by RReliefF

Conclusion

Limitations

- Dataset used only consisted of students who are attending secondary schools
- Imbalanced dataset with ratio of 9:1, SMOTE only creates synthetic examples based on k-nearest-neighbors (Chawla et al., 2002)
- Limited generalizability as dataset consists of only students from Portuguese secondary schools.

Future Works

- Include data for adolescents who are not attending school
- Use a more balanced dataset with more data from adolescents who have alcohol addiction
- Include data of adolescents from multiple countries and continents, to further improve generalization.

References

Arshad, M., Omar, M., & Shahdan, N. A. (2015). Alcoholism among youth: A case study in Kuala Lumpur, Malaysia. International Journal of Culture and History, 1(1), 21-28.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16, 321-357.

Chhoa, K. H., Zakaria, H., & Abd Rahman, F. N. (2019). Problematic alcohol use and depression in secondary school students in Miri, Malaysia. Pediatrics international, 61(3), 284-292.

Cortez, P., & Silva, A. M. G. (2008). Using data mining to predict secondary school student performance.

Gowin, J. L., Sloan, M. E., Stangl, B. L., Vatsalya, V., & Ramchandani, V. A. (2017). Vulnerability for alcohol use disorder and rate of alcohol consumption. American Journal of Psychiatry, 174(11), 1094-1101.

Institute for Public Health Malaysia. (2019). National Health and Morbidity Survey (NHMS) 2019: Non-Communicable Diseases, Healthcare Demand and Health Literacy. (Publication No. NMRR-18-3085- 44207).

Ismail, S., Azlan, N. I. A. N., & Mustapha, A. (2018). Prediction of alcohol consumption among Portuguese secondary school students: A data mining approach. In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 383-387). IEEE.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. In European conference on machine learning (pp. 171-182). Springer, Berlin, Heidelberg.

NHS. (2022, October 4). Alcohol misuse - Risks. Retrieved January 9, 2023, from https://www.nhs.uk/conditions/alcohol-misuse/risks/

Palaniappan, S., Hameed, N. A., Mustapha, A., & Samsudin, N. A. (2017). Classification of alcohol consumption among secondary school students. JOIV: International Journal on Informatics Visualization, 1(4-2), 224-226.

World Health Organization. (2022, May 9). Alcohol. Retrieved January 9, 2023, from https://www.who.int/news-room/fact-sheets/detail/alcohol