DATA5322 Written Homework 1
Ting-Yu, Lin

Analysis of Youth Drug Use: A Decision Tree Approach

Abstract

This project utilizes decision trees to analyze youth drug use using data from the 2020 National Survey on Drug Use and Health (NSDUH), focusing on youth experiences, demographics, and substance use. Our classification models achieve accuracies above 0.9, indicating effective data analysis. We discover patterns showing how the use of one substance may relate to another and how peer perceptions influence drug use behaviors. By employing decision trees and ensemble methods, we aim to uncover hidden patterns and drive evidence-based interventions to mitigate youth drug use, enhancing our understanding of the complex factors influencing these behaviors.

Introduction

Youth drug use is a multifaceted issue that encompasses a range of substances, including tobacco, alcohol, marijuana, and other illegal drugs.The patterns of use among youths can be influenced by various factors such as economic status, peer pressure, family dynamics, health conditions, and the availability of substances. Our project aims to delve into the underlying factors associated with youth drug use.

We leverage decision trees model —a form of machine learning— to analysis data from the 2020 National Survey on Drug Use and Health (NSDUH). The dataset provides a comprehensive of data and categories available. Our analysis is structured into three distinct domains: youth experiences, demographic factors, and substance use.Through careful selection and processing of the data, we aim to identify underlying patterns and also to provide evidence-based recommendations that can inform strategies to address youth drug use issues.

Background

Decision Tree

In our project, we use decision trees for data analysis. These are non-parametric, supervised learning methods ideal for classification and regression tasks. The goal is to build models that predict target values by learning decision rules from data features.

Decision trees initiate from a root node and branch out based on decision rules applied to input features. This branching continues until it meets certain criteria such as

minimum node size or maximum tree depth, which are adjustable parameters during model training. The leaves of the tree denote outcomes—class labels for classification tasks and continuous values for regression.

The construction of a decision tree involves selecting the best attributes to split the data, guided by metrics like classification error, the Gini index, or entropy, aiming to create homogenous subsets for accurate predictions. And these indicators have been calculated with the following formula:

$$E = 1 - \text{largest proportion in class} = 1 - \max_k (\hat{p}_{mk})$$

Classification error

$$G = \sum_{\text{classes}} \text{proportion in class}\,(1 - \text{proportion in class}) = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk})$$

Gini index

$$D = -\sum_{\text{classes}} (\text{prop in class}) \log (\text{prop in class}) = -\sum_{k=1}^{K} \hat{p}_{mk} \log (\hat{p}_{mk})$$

Entropy

Decision Tree Ensemble Methods

Decision tree ensembles use multiple decision trees to enhance prediction accuracy and robustness. Bagging involves running multiple models on different random subsets of the training data, where each subset is created with replacement. By aggregating results from several trees, these ensembles help reducing the overfitting which commonly seen with single decision trees. The final output is either averaged (for regression) or determined by majority vote (for classification).

Random Forests enhance the bagging approach by introducing feature randomness. Instead of selecting the best feature from all available features for each split, the algorithm picks a random subset and chooses the best feature from it. This method, which involves building each tree by sampling "m' predictors from the total 'p' available predictors, allows for a broader exploration of potential trees. To run the Random Forest algorithm, you must provide the predictor variables, the response variables, and specify 'm', the number of predictors to sample for each tree.

Boosting is another ensemble technique we'll use in our project, where each new tree is developed to correct the errors from the previous trees. These models focus on improving accuracy by specifically addressing challenging cases misclassified in earlier iterations. During training, we can adjust the shrinkage (α) to regulate the learning rate, thereby impacting training speed and overall model performance.

Methodology

Data Preparation

To prepare the data for our models, we first selected the variables of interest and categorize them into youth experiences, demographics, and substance use. This categorization will be helpful for model building later on.

Next, we converted some categorical data to factors, including binary and ordered where appropriate. In some variable columns, certain values like 991, 993, 91, 93, 5, and 6 lack numerical significance; they likely represent "Never used" or "Not used in the past period." Therefore, we convert them to 0. Additionally, values such as 94, 97, 98, and 99 in the "eduskpcom" column do not have analytical meaning, so we converted them to NA.

Furthermore, to ensure interpretability, we removed data with missing values before proceeding with the analysis.

Models

For each model, the data was split into 70% for training and the remaining 30% was reserved for testing model accuracy or MSE. In binary classification, "mrjflag" (whether marijuana had been used or not) was chosen as the response variable, and "alcflag" (whether alcohol had been used or not), "tobflag" (whether tobacco had been used or not), along with variables in demographic_cols and youth_experience_cols were used as predictors, to predict marijuana use. A single decision tree was first used for prediction, followed by a bagging approach by adjusting the number of trees to reduce variance and achieve optimal accuracy.

In multi-class classification, "mrjmdays" (frequency of marijuana use in the past month) was selected as the response variable, and variables in demographic_cols and youth_experience_cols were used as predictors, to predict the frequency of marijuana use in the past month. A single decision tree was initially used for prediction, followed by random forest by adjusting the number of variables sampled at each split (mtry) to improve accuracy.

For regression, "irmjfy" (frequency of marijuana use in the past year) was chosen as the response variable, and variables in demographic_cols and youth_experience_cols were used as predictors, to predict the number of days of marijuana use in the past year. A single decision tree was first used for prediction, followed by a boosting approach by adjusting the learning rate to obtain the optimal model.

Results

Binary classification

This classification tree model, shown in Figure 1, was built using training data to predict the "mrjflag" variable. It incorporated variables such as "alcflag," "yflmjmo," "stndsmj," "frdmjmon," and "tobflag" in its construction. With 8 terminal nodes, the model achieved a residual mean deviance of 0.4593, indicative of a good fit. It has a misclassification error rate of 0.09796, meaning that around 9.8% of the samples were incorrectly classified. The model's predictions on the test data, presented in Table 1(a), result in an accuracy of approximately 0.9174.

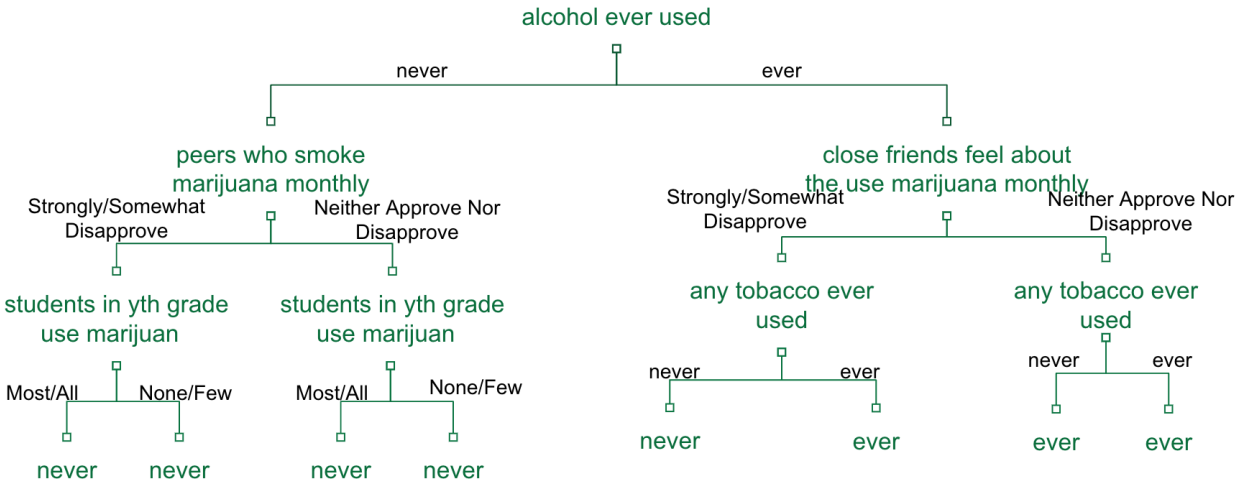
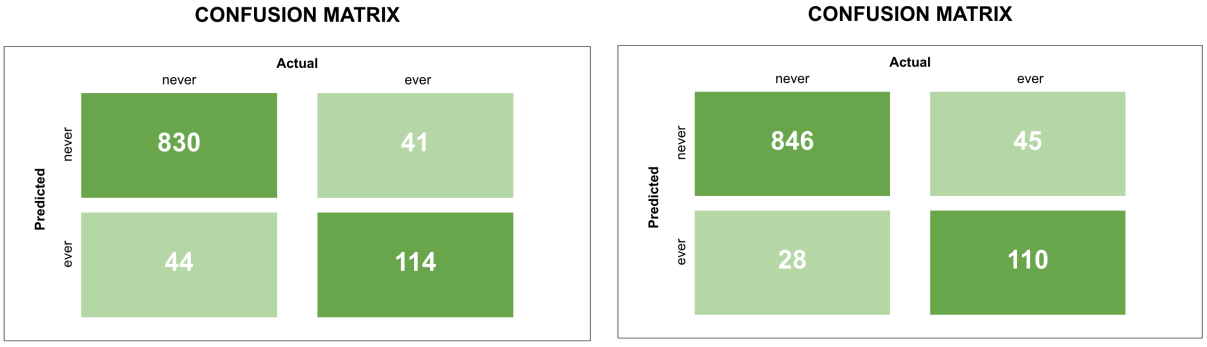


Figure 1. The decision tree for determining whether marijuana had been used or not.



(a) (b)

Figure 2. The confusion matrices for binary classification tasks.(a) Confusion Matrix with a Single Decision Tree (b) Confusion Matrix with Bagging Approach.

To obtain the most suitable bagging approach, we tuned the models for the values of ntrees (number of trees). The results for the tuning can be seen in Figure 3. We observed that with different values of ntrees, there is not much difference. We were still able to identify the tree number with the lowest error rate, enabling us to proceed with the random forest model using bagging approach and obtain the optimal model. We got the result with each decision tree considering 62 variables at every split. The model yields an out-of-bag (OOB) error estimate of 10.21%, indicating an expected prediction error rate of approximately 10.21% on unseen data. The model's predictions on the test data, presented in Figure 2(b), result in an accuracy of approximately 0.9291, indicating a slight improvement compared to the single decision tree model. Through Figure 4, the Gini Index reflects how each variable contributes to the homogeneity of the nodes and splits in the tree. Variables "alcflag" and "tobflag" show high values, indicating their importance in creating pure nodes in the tree that effectively distinguish between users and non-users of marijuana. This result is similar to that of the previous single decision tree model.

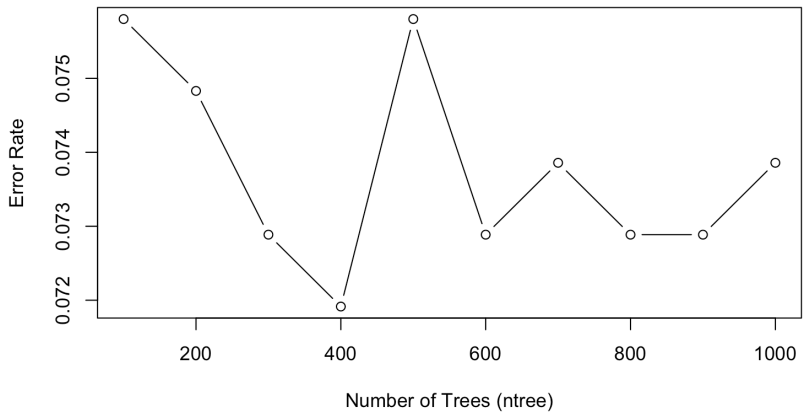


Figure 3. Impact of tree number on bagging approach

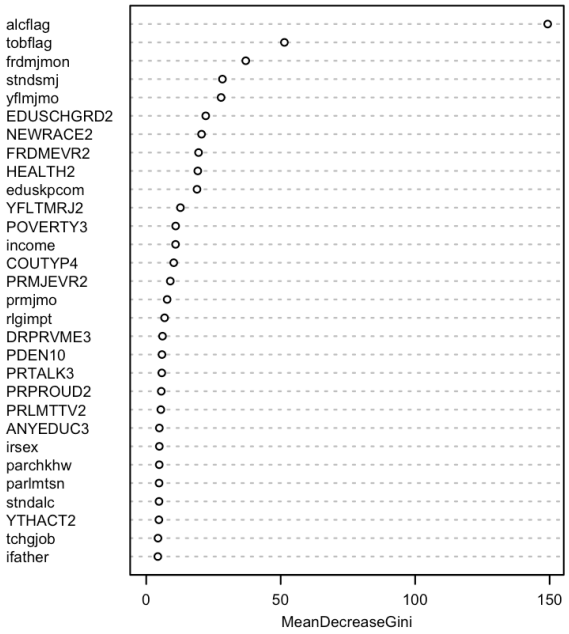


Figure 4. Variable importance in predicting youth marijuana use

Multi-class classification

The single tree model for multi-class classification was constructed using the training data to predict the "mrjflag" variable. It utilized 8 variables and comprised 8 terminal nodes. The model attained a residual mean deviance of 0.4474 and a misclassification error rate of 0.06503. Upon evaluating the model's predictions on the test data, it achieved an accuracy of approximately 0.9291.

To improve the model, we utilized the random forest approach and adjusted the models for various values of mtry. The tuning outcomes are illustrated in Figure 5. The best-performing model was determined with an mtry value of 21. This model yields an out-of-bag (OOB) error estimate of 7%. Furthermore, it achieved an accuracy of 0.9300, slightly exceeding that of the single decision tree model. Through Figure 6, it's evident that variables "EDUSCHGRD2" (current or expected grade level) and "frdmjmon" exhibit high importance values. This highlights their significance in creating pure nodes within the tree, thus aiding in effectively distinguishing different levels of monthly marijuana usage frequency.

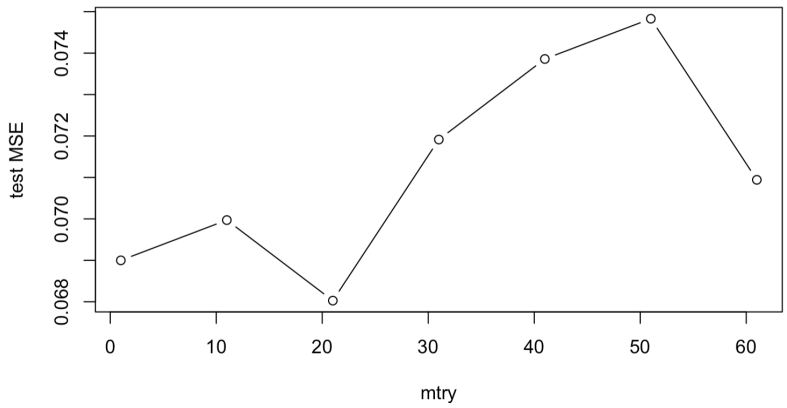


Figure 5. Impact of mtry on random forest approach

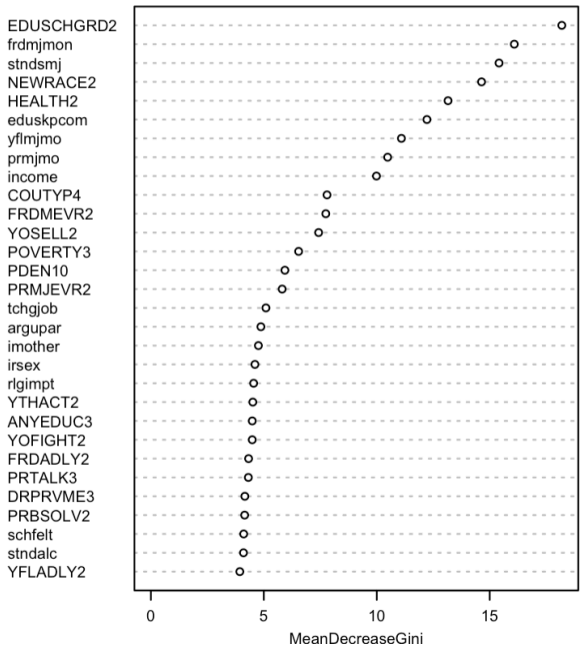


Figure 6. Variable importance for predicting monthly marijuana usage frequency.

Regression

The single tree model for regression was built using the training data to predict the "irmjfy" variable. It employed 6 variables and consisted of 7 terminal nodes. Upon evaluating the model's predictions on the test data, it resulted in a test MSE of approximately 1358.83.

To enhance the model, we applied the boosting approach and adjusted the models for various learning rate values. The optimal model was identified with a learning rate of 0.03 and a maximum depth of 3 for tree growth. This model achieved a test MSE of approximately 1243.07, lower than the single decision tree model.

Through Figure 7, it's evident that variables "prmjmo" (parents' feelings about youth marijuana use) and "stndsmj" exhibit high importance values. This underscores their significance in creating pure nodes within the tree, thus aiding in effectively predicting the number of days of marijuana use in the past year.

	var <chr>	rel.inf <dbl>
prmjmo	prmjmo	11.98490856
stndsmj	stndsmj	10.88768862
frdmjmon	frdmjmon	9.36283722
YOSELL2	YOSELL2	8.24646278
yflmjmo	yflmjmo	7.02409954
YOSTOLE2	YOSTOLE2	3.75306732
EDUSCHGRD2	EDUSCHGRD2	3.11484117
eduskpcom	eduskpcom	3.03197286
FRDADLY2	FRDADLY2	3.03032351
NEWRACE2	NEWRACE2	2.72563708

Figure 7. Top 10 variables in importance for predicting the number of days of marijuana use in the past year.

Discussion

Flow and Interpretation of a Tree Model

In the flow of our tree model as depicted in Figure 1, begins at the root node with an assessment of "alcohol ever used." As the tree branches, a notable split occurs at "any tobacco ever used" , which strongly suggests the influence of alcohol and tobacco use behaviors. The diagram shows that on the left side—where no alcohol use is reported—the analysis tends to predict a lower likelihood of marijuana use. Conversely, on the right side of the tree, where alcohol use is confirmed, and when factoring in friends who neither approve nor disapprove of monthly marijuana use, the likelihood of marijuana use markedly increases.

Data Types and Predictive Changes

The model incorporates variables in different formats: binary, ordinal, and numerical. Our analysis shows that binary variables reveal distinct behavior patterns, while ordinal variables offer insights into drug use frequency and severity. Numerical variables, such as the number of days marijuana was used, provide direct quantification and enable precise regression modeling. The appropriate use of each data type depends on the

specific analysis goal—binary and ordinal variables are beneficial for classification tasks, while numerical variables are essential for regression.

Importance of Variables

Variables such as "alcflag" and "tobflag" play a crucial role in predicting marijuana use, indicating a pattern where the use of one substance may be related to the use of another. "prmjmo" and "frdmjmon" are also significant factors, suggesting that perceptions of drug use by people around can have a certain degree of influence.

Ethical Consideration in Communication

As data scientists, it is our responsibility to communicate findings ethically. We must present data objectively, without inferring causation from correlation. Our communication should aim to inform, support public health efforts, and contribute to a comprehensive understanding of youth drug use.

Conclusions

Our study provided valuable insights into the predictors of youth drug use through decision trees and ensemble methods. The variables related to alcohol and tobacco use, along with perceptions of drug use among peers, emerged as significant predictors. Our findings underscore the interrelated nature of substance use and the social factors surrounding youth. It is important to approach these findings with caution and ethical consideration, recognizing the difference between correlation and causation. The knowledge gained from our models can inform public health strategies aimed at understanding and reducing drug use among youth.

References

National Survey on Drug Use and Health (NSDUH**),** https://www.datafiles.samhsa.gov/dataset/national-survey-drug-use-and-health-2020-nsduh-2020-ds0001

Appendix

Data Preparation

```{r}
data_clean <- df%>%
  mutate(across(everything(), ~ replace(.x, .x %in% c(993, 991), 0)))
```

```
data_clean <- data_clean%>%
  mutate(across(c(ircigfm, IRSMKLSS30N, iralcfm, irmjfm), ~ replace(.x, .x %in% c(93, 91), 0)))

data_clean <- data_clean %>%
  mutate(across(c(alcmdays, mrjmdays, smklsmdays), ~ replace(.x, .x == 5, 0)))

data_clean <- data_clean %>%
  mutate(across(c(alcydays, mrjydays, cigmdays), ~ replace(.x, .x == 6, 0)))

data_clean <- data_clean %>%
  mutate(eduskpcom = replace(eduskpcom, eduskpcom %in% c(94, 97, 98, 99), NA))

data_clean  <- na.omit(data_clean)
```

Split data to train dataset and test dataset

```{r}
indices <- sample(1:nrow(data_clean), size = 0.7*nrow(data_clean))

# Split data based on indices
train_data <- data_clean[indices, ]
test_data <-data_clean[-indices, ]
```

Binary classification

```{r}
data_for_model1 <- data_clean[, c("mrjflag","alcflag", "tobflag", demographic_cols,youth_experience_cols)]
```

Single tree

```{r}
tree.mrjflag <- tree(mrjflag ~ ., train_data)

plot(tree.mrjflag)
text(tree.mrjflag, pretty = 0)

summary(tree.mrjflag)

tree.pred <- predict(tree.mrjflag, test_data,
```

```
    type = "class")
confusion_matrix <- table(tree.pred, test_data$mrjflag)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
```
Bagging approach
```{r}
all_predictors<- ncol(data_for_model1)-1
ntree_number <- seq(100, 1000, by = 100)

error_rates <- numeric(length(ntree_number))

for (i in 1:length(ntree_number)) {
  mrjflag_bag <- randomForest(mrjflag ~ ., data = train_data, mtry = all_predictors, ntree
= ntree_number[i])
  yhat <- predict(mrjflag_bag, newdata = test_data, type="class")
  error_rates[i] <- sum(yhat != test_data$mrjflag) / nrow(test_data)
}

# Plot the error rates against the number of trees
plot(ntree_number, error_rates, type = "b", xlab = "Number of Trees (ntree)", ylab =
"Error Rate")
best_ntree_number <- ntree_number[which.min(error_rates)]
```

```{r}
mrjflag_bag <- randomForest(mrjflag ~ ., data = train_data, mtry = all_predictors,
importance = TRUE, ntree = best_ntree_number)
```

```{r}
mrjflag.pred <- predict(mrjflag_bag, test_data, type='class')
accuracy <- 1 - sum(mrjflag.pred != test_data$mrjflag) / nrow(test_data)
paste("Accuracy:", accuracy)
```

```{r}
confusion_matrix <- table(mrjflag.pred, test_data$mrjflag)
accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
confusion_matrix
print(accuracy)
```

```

```{r}
varImpPlot(mrjflag_bag, cex = 0.55)
```

Multi-class classification

```{r}
data_for_model2 <- data_clean[, c("mrjmdays",
demographic_cols,youth_experience_cols)]
data_for_model2$mrjmdays <- factor(data_for_model2$mrjmdays)
```
```{r}
mtry_values <- seq(1, ncol(train_data), by = 10)
error_rates <- numeric(length(mtry_values))
for (i in 1:length(mtry_values)) {
  mrjmdays_rf <- randomForest(mrjmdays ~ ., data = train_data, mtry = mtry_values[i])
  yhat <- predict(mrjmdays_rf, newdata = test_data, type="class")
  error_rates[i] <- sum(yhat != test_data$mrjmdays) / nrow(test_data)
}

plot(mtry_values, error_rates, type = "b", xlab = "mtry", ylab = "test MSE",)
best_mtry <- mtry_values[which.min(error_rates)]
```

```{r}
mrjmdays_rf <- randomForest(mrjmdays ~ ., data = train_data, mtry = best_mtry,
importance = TRUE)
```

Regression
```{r}
data_for_model3 <- data_clean[, c("irmjfy", demographic_cols,youth_experience_cols)]
```
```{r}
lambdas <- seq(0.01, 0.2, by = 0.01)
test_mse <- numeric(length(lambdas))

for (i in 1:length(lambdas)) {
  set.seed(1)
```
```

```
  irmjfy_boost <- gbm(irmjfy~ ., data = train_data,
                      distribution = "gaussian", n.trees = 500,
                      interaction.depth = 3, shrinkage = lambdas[i], verbose = F)
  yhat_boost <- predict(irmjfy_boost, newdata = test_data, n.trees = 500)
  test_mse[i] <- mean((yhat_boost - test_data$irmjfy)^2)
}

plot(lambdas, test_mse, type = "b", xlab = "Shrinkage", ylab = "Test Set MSE")
```

```{r}
irmjfy_boost <- gbm(irmjfy ~ ., data = train_data,
                    distribution = "gaussian", n.trees = 500,
                    interaction.depth = 3, shrinkage = best_lambdas, verbose = FALSE)
yhat_boost <- predict(irmjfy_boost, newdata = test_data, n.trees = 500)
mean((yhat_boost - test_data$irmjfy)^2)
```