Used cars project

Dataset Overview: The dataset contains information about used cars, including their brand, model, year, age, kilometers driven, transmission type, owner type, fuel type, posted date, additional information, and asking price.

Key Processing Steps:

- Data exploration
- Duplicate removal
- Currency conversion
- Feature engineering
- Model development for both classification and regression
- Data cleaning including duplicate removal
- Currency conversion from Ruby to USD
- Classification Section
- Features separation and target variable identification
- Model preparation and training
- Regression Section
- Data preparation for regression analysis

Used algorithms:

- Random forest
- Decision tree
- Logistic regression
- Linear regression
- Neural network

Algorithm accuracy :

```
log_reg_model <- glm(y_train ~ ., data = X_train, family = binomial)
predictions <- predict(log_reg_model, X_test, type = "response")
pred_classes <- ifelse(predictions > 0.5, "High", "Low")
log_reg_accuracy <- mean(pred_classes == y_test)
print(paste("Logistic Regression Accuracy:", log_reg_accuracy))
```

```
[1] "Logistic Regression Accuracy: 0.817717206132879"
```

```
linear_model <- lm(mpg ~ ., data = cbind(X_train, mpg = y_train))
linear_predictions <- predict(linear_model, newdata = X_test)

mae <- mae(y_test, linear_predictions)
rmse <- rmse(y_test, linear_predictions)
rss <- sum((y_test - linear_predictions)^2)
tss <- sum((y_test - mean(y_test))^2)
r2 <- 1 - (rss / tss)

print(paste("MAE : " , mae))
print(paste("RMSE : " , rmse))
print(paste("R2 : " , r2))
```

```
Warning message in predict.lm(linear_model, newdata = X_test):
“prediction from rank-deficient fit; attr(*, "non-estim") has doubtful cases”
[1] "MAE :  1631.27905581527"
[1] "RMSE :  2323.66307039889"
[1] "R2 :  0.899920996945397"
```

```
randomforest_model <- randomForest(x = X_train, y = y_train, ntree = 100,
random_state = 42)
randomforest_predictions <- predict(randomforest_model, X_test)
randomforest_accuracy <- mean(randomforest_predictions == y_test)

print(paste("Random Forest Classifier Accuracy:", randomforest_accuracy))
```

```
[1] "Random Forest Classifier Accuracy: 0.938671209540034"
```

```
decision_tree_model <- rpart(y_train ~ ., data = X_train, method = "class")
decision_tree_predictions <- predict(decision_tree_model, X_test, type =
"class")
decision_tree_accuracy <- mean(decision_tree_predictions == y_test)

print(paste("Decision Tree Accuracy:", decision_tree_accuracy))
```

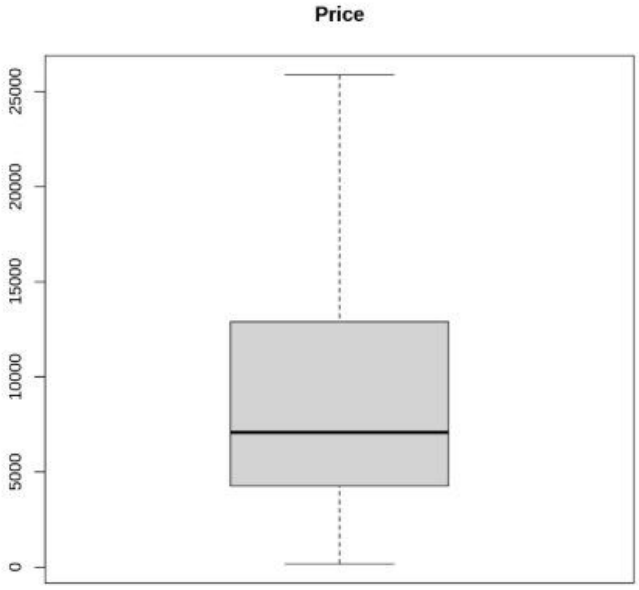
```
[1] "Decision Tree Accuracy: 0.880749574105622"
```

```
nn_model <- nnet(y_train ~ ., data = X_train, size = 5, decay = 0.01, maxit =
200)
nn_predictions <- predict(nn_model, newdata = X_test, type = "class")
nn_accuracy <- mean(nn_predictions == y_test)
print(paste("Neural Network Accuracy:", nn_accuracy))
```

```
# weights:  41
initial  value 4141.579883
iter  10 value 4033.660039
iter  20 value 3969.233695
iter  30 value 3827.812940
iter  40 value 3427.913611
iter  50 value 2907.172490
iter  60 value 2761.798243
iter  70 value 2670.399123
iter  80 value 2597.088030
iter  90 value 2528.419643
iter 100 value 2459.334659
iter 110 value 2367.248844
iter 120 value 2314.528306
iter 130 value 2289.703676
iter 140 value 2260.450194
iter 150 value 2234.222746
iter 160 value 2210.975618
iter 170 value 2194.717749
iter 180 value 2192.607260
iter 190 value 2192.129109
iter 200 value 2190.906175
final  value 2190.906175
stopped after 200 iterations
[1] "Neural Network Accuracy: 0.860874503123225"
```

Visualization :

```
boxplot(data$AskPrice , main = "Price")
boxplot(data$kmDriven , main = "km Driven")
boxplot(data$Age , main = "Age")
boxplot(data$kmperage , main = "km per age")
```




```
boxplot(data$AskPrice , main = "Price")
boxplot(data$kmDriven , main = "km Driven")
boxplot(data$Age , main = "Age")
boxplot(data$kmperage , main = "km per age")
```