컴퓨터정보실무

취향에 따른 차량 추천 시스템

2011270314

컴퓨터정보학과

서인석

목차

개요 (1/2)

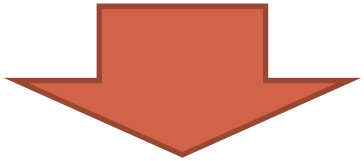
프로젝트의 목적

- 과거 다른 소비자들의 선택을 학습하여, 해당 소비자의 취향에 맞도록 차량의 선택을 추측하여 차량을 추천해 주는 시스템을 구현한다.

개요 (2/2)

21세기인 현재, 차는 선택이 아닌 필수라고 해도 과언이 아닐 만큼 굉장히 중요한 존재가 되었다. 그만큼 개인 자가용의 구입이 늘게 되었으며, 자동차 회사와 선택 할 수 있는 차량의 종류 또한 늘게 되었다.

- **차량 추천 시스템의 필요성**
  - 개인마다 취향이 다르기에 차를 선택하는 기준이 다르다!
  - 기존의 추천 시스템은 인터넷이나 주변 지인들의 경험담, 혹은 소문에 의존하여 차를 추천 받아 선택하는, 수동적인 시스템이다!



그러므로 차량선택에 고민하는 소비자에게 소비자의 개인에 취향에 맞는 차량을 추천하는 시스템이 있다면, 이를 활용하여 소비자는 차량선택에 있어 시간과 수고를 덜 수 있고, 기업은 소비자들의 취향에 따라 자사의 차량을 홍보하는데 도움이 될 것이다.

기계학습단계에 따른 적용 과정(1/2)

데이터 수집

차량선택을 예측, 추천하기 위해 기존 소비자들의 취향 특징과 선택한 차량의 정보의 데이터가 필요

데이터 준비와 탐구

얻은 데이터를 입력하고, 알고리즘에 맞도록 전처리 과정을 거친 뒤 훈련데이터와 테스트데이터로 나눔

데이터에 대한 모델 훈련

전처리 한 테스트 데이터를 알고리즘에 적용하여 모델을 만듬

기계학습단계에 따른 적용 과정(2/2)

모델 성능 평가

테스트데이터를 만든 모델에 적용하여 모델의 성능을 평가 하고, 가장 성능이 좋은, 최적의 알고리즘을 선택

모델 성능 향상

모델의 성능을 향상시키기 위해 선택한 알고리즘에 맞게 데이터를 조정하여 성능을 향상

학습 과정 (데이터 수집)

- 기계학습에 적용하기 위해 기존 소비자들의 차량 취향의 정보와 차량선택의 정보가 있는 데이터가 필요하다.

	choice	college	hsg2	coml5	type1	type2	type3	type4	type5	type6	fuel1	fuel2	fuel3	fuel4	fuel5	fuel6	price1	price2	price3	price
1	choice1	0	0	0	van	regcar	van	stwagon	van	truck	cng	cng	electric	electric	gasoline	gasoline	4.175345	4.175345	4.817706	4.81
2	choice2	1	1	1	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	3.310947	3.310947	3.586859	3.58
3	choice5	0	1	0	regcar	truck	regcar	van	regcar	stwagon	cng	cng	electric	electric	gasoline	gasoline	4.039575	4.039575	2.777208	2.77
4	choice5	0	0	1	regcar	truck	regcar	van	regcar	stwagon	methanol	methanol	cng	cng	electric	electric	7.065968	7.065968	7.387149	7.38
5	choice5	0	1	0	regcar	truck	regcar	van	regcar	stwagon	cng	cng	electric	electric	gasoline	gasoline	5.794157	5.794157	6.345981	6.34
6	choice5	0	0	0	truck	regcar	truck	van	truck	stwagon	cng	cng	electric	electric	gasoline	gasoline	3.532984	3.532984	4.175345	4.17
7	choice2	1	1	1	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	1.927082	1.927082	0.963541	0.96
8	choice5	1	0	1	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	6.070069	6.070069	6.345981	6.34
9	choice5	0	0	0	sportuv	sportcar	sportuv	regcar	sportuv	truck	methanol	methanol	electric	electric	gasoline	gasoline	3.067467	3.067467	3.834333	3.83
10	choice2	1	0	0	regcar	truck	regcar	van	regcar	stwagon	methanol	methanol	cng	cng	electric	electric	3.801738	3.801738	2.459948	2.45
11	choice1	1	0	0	regcar	truck	regcar	van	regcar	stwagon	methanol	methanol	cng	cng	electric	electric	4.138684	4.138684	4.414596	4.41
12	choice2	1	1	0	truck	stwagon	truck	regcar	truck	van	methanol	methanol	electric	electric	gasoline	gasoline	3.323736	3.323736	3.798555	3.79
13	choice5	1	0	0	sportcar	truck	sportcar	sportuv	sportcar	regcar	methanol	methanol	cng	cng	gasoline	gasoline	5.460066	5.460066	6.102427	6.10
14	choice5	0	0	0	regcar	stwagon	regcar	truck	regcar	van	methanol	methanol	electric	electric	gasoline	gasoline	1.284722	1.284722	1.605902	1.60
15	choice3	1	0	0	regcar	stwagon	regcar	truck	regcar	van	methanol	methanol	electric	electric	gasoline	gasoline	4.690508	4.690508	3.586859	3.58
16	choice6	1	0	0	truck	van	truck	stwagon	truck	regcar	methanol	methanol	cng	cng	gasoline	gasoline	1.931386	1.931386	1.931386	1.93
17	choice3	1	1	1	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	8.437131	8.437131	5.45932	5.4
18	choice2	0	0	0	regcar	truck	regcar	van	regcar	stwagon	methanol	methanol	cng	cng	electric	electric	1.022489	1.022489	1.1503	1.
19	choice3	1	1	0	van	regcar	van	stwagon	van	truck	methanol	methanol	cng	cng	electric	electric	3.310947	3.310947	3.586859	3.58
20	choice5	1	0	1	regcar	stwagon	regcar	truck	regcar	van	methanol	methanol	electric	electric	gasoline	gasoline	3.310947	3.310947	4.138684	4.13
21	choice5	0	1	1	regcar	stwagon	regcar	truck	regcar	van	methanol	methanol	electric	electric	gasoline	gasoline	5.955622	5.955622	7.444527	7.44
22	choice6	1	0	0	sportuv	truck	sportuv	sportcar	sportuv	regcar	methanol	methanol	cng	cng	gasoline	gasoline	4.856822	4.856822	5.368067	5.36
23	choice5	0	0	1	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	4.690508	4.690508	3.035035	3.03
24	choice3	1	0	0	regcar	truck	regcar	van	regcar	stwagon	methanol	methanol	cng	cng	electric	electric	4.690508	4.690508	3.035035	3.03
25	choice1	1	0	0	truck	regcar	truck	van	truck	stwagon	methanol	methanol	cng	cng	electric	electric	7.065968	7.065968	7.387149	7.38
26	choice3	1	0	1	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	4.138684	4.138684	4.414596	4.41
27	choice5	1	0	0	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	3.392783	3.392783	2.19533	2.1
28	choice5	0	0	0	truck	regcar	truck	van	truck	stwagon	cng	cng	electric	electric	gasoline	gasoline	0.963541	0.963541	1.124131	1.12
29	choice1	1	0	1	regcar	van	regcar	stwagon	regcar	truck	methanol	methanol	cng	cng	gasoline	gasoline	6.621894	6.621894	4.690508	4.69

(출처 : https://vincentarelbundock.github.io/Rdatasets/datasets.html)

학습 과정 (데이터 전처리)

- 데이터를 학습하기 위해 미리 준비한 데이터인 Car.csv를 읽어 데이터 프레임으로 저장한다.
- 데이터 프레임으로 저장한 데이터를 알고리즘에 적용하기 위해 전처리 과정을 거친다.

```
car <- read.csv( car.csv , stringsAsFactors = FALSE
car_train <- car[1:3000,]
car_test <- car[3001:4654,]
car_int <- car
car_int$choice <- as.integer(car_int$choice)
View(car_int)
View(car_int)
car_int_train <- car_int[1:3000,]
car_int_test <- car_int[3001:4654]
car_int_test <- car_int[3001:4654,]
save.image("~/project.RData")
savehistory("~/.Rhistory")
load("~/project.RData")
normal <- function(x) {
return ((x - min(x)) / (max(x) - min(x)))
}
car_nor <- as.data.frame(lapply(car_int, normal))
```

학습 과정 (데이터에 대한 모델훈련)

- 전처리 과정을 거친 데이터를 여러 알고리즘에 적용한다.

Naivebayes 알고리즘

```
car_Naive_m <- naiveBayes(car_train,car_train$choice)
car_Naive_p <- predict(car_Naive_m,car_test)
```

c5.0 알고리즘 (DST)

```
car_DST_m <- C5.0(car_train[-2],car_train$choice )
car_DST_m
summary(car_DST_m)
car_DST_p <- predict(car_DST_m,car_test)
```

CART 알고리즘(회귀 트리)

```
car_CART_m <- rpart(choice ~ ., data = car_train)
car_CART_m
car_CART_p <- predict(car_CART_m, car_test)
```

학습 과정 (모델 성능 평가)

- Naivebayes 알고리즘 성능 평가

actual	predicted choice1	choice2	choice3	choice4	choice5	choice6	Row Total
choice1	266	0	18	0	2	7	293
	0.908	0.000	0.061	0.000	0.007	0.024	0.177
	0.950	0.000	0.038	0.000	0.004	0.059	
choice2	1	87	6	0	9	3	106
	0.009	0.821	0.057	0.000	0.085	0.028	0.064
	0.004	0.897	0.013	0.000	0.016	0.025	
choice3	2	2	437	0	42	8	491
	0.004	0.004	0.890	0.000	0.086	0.016	0.297
	0.007	0.021	0.926	0.000	0.074	0.067	
choice4	1	0	4	105	11	3	124
	0.008	0.000	0.032	0.847	0.089	0.024	0.075
	0.004	0.000	0.008	0.890	0.019	0.025	
choice5	5	5	5	10	501	0	526
	0.010	0.010	0.010	0.019	0.952	0.000	0.318
	0.018	0.052	0.011	0.085	0.882	0.000	
choice6	5	3	2	3	3	98	114
	0.044	0.026	0.018	0.026	0.026	0.860	0.069
	0.018	0.031	0.004	0.025	0.005	0.824	
Column Total	280	97	472	118	568	119	1654
	0.169	0.059	0.285	0.071	0.343	0.072	

학습 과정 (모델 성능 평가)

- c5.0 알고리즘 (DST) 성능 평가

```
Decision Tree
----------------
Size      Errors

 655  811(27.0%)   <<


 (a)   (b)   (c)   (d)   (e)   (f)    <-classified as
----  ----  ----  ----  ----  ----
 437     2    66     6    80     3    (a): class choice1
  23    86    28     6    18     2    (b): class choice2
  40     8   707     9    87     3    (c): class choice3
  31     5    40   115    33     1    (d): class choice4
  55    12   103    22   774     7    (e): class choice5
  25     8    41     9    38    70    (f): class choice6
```

actual	predicted choice1	choice2	choice3	choice4	choice5	choice6	Row Total
choice1	57	15	86	19	102	14	293
	0.195	0.051	0.294	0.065	0.348	0.048	0.177
	0.208	0.211	0.177	0.136	0.161	0.292	
choice2	23	10	24	8	37	4	106
	0.217	0.094	0.226	0.075	0.349	0.038	0.064
	0.084	0.141	0.049	0.057	0.058	0.083	
choice3	74	9	174	53	170	11	491
	0.151	0.018	0.354	0.108	0.346	0.022	0.297
	0.270	0.127	0.357	0.379	0.268	0.229	
choice4	18	11	42	10	40	3	124
	0.145	0.089	0.339	0.081	0.323	0.024	0.075
	0.066	0.155	0.086	0.071	0.063	0.062	
choice5	88	20	129	40	236	13	526
	0.167	0.038	0.245	0.076	0.449	0.025	0.318
	0.321	0.282	0.265	0.286	0.372	0.271	
choice6	14	6	32	10	49	3	114
	0.123	0.053	0.281	0.088	0.430	0.026	0.069
	0.051	0.085	0.066	0.071	0.077	0.062	
Column Total	274	71	487	140	634	48	1654
	0.166	0.043	0.294	0.085	0.383	0.029	

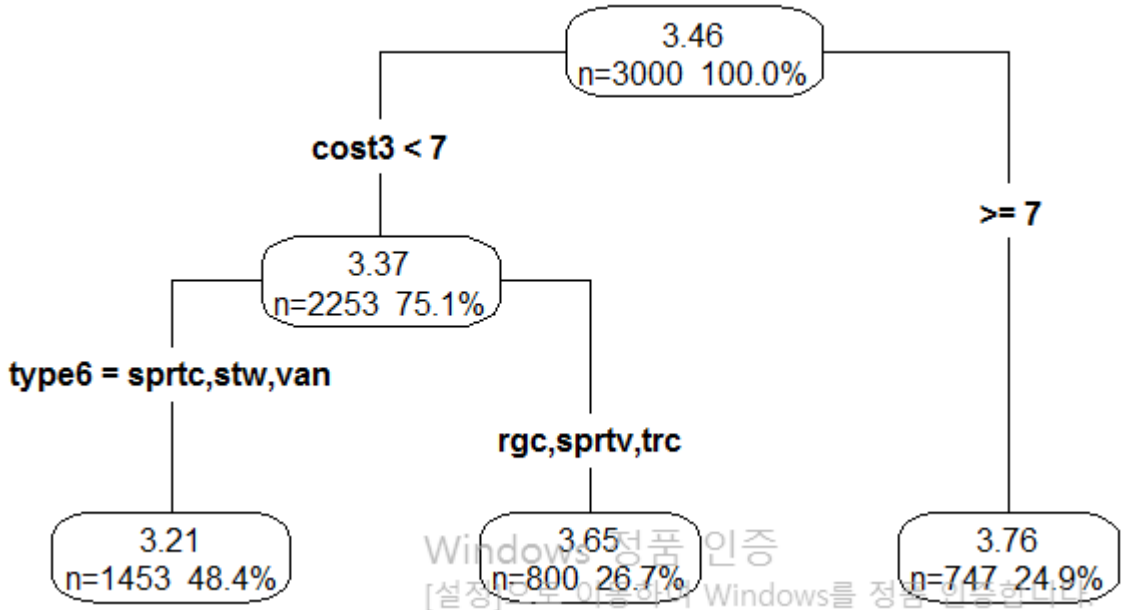
학습 과정 (모델 성능 평가)

- CART 알고리즘(회귀 트리) 성능 평가(1/2)

```
n= 3000

        CP nsplit rel error    xerror       xstd
1 0.012105      0   1.00000 1.0009608 0.01644004
2 0.010000      2   0.97579 0.9886709 0.01697059

Variable importance
type6 cost3 cost4 type4 type2 cost5 cost6 type1 type3 type5
   17    15    15    13    11     8     8     5     5     5

Node number 1: 3000 observations,    complexity param=0.012105
  mean=3.464333, MSE=2.576061
  left son=2 (2253 obs) right son=3 (747 obs)
  Primary splits:
      cost3  < 7        to the left,  improve=0.011279980, (0 missing)
      cost4  < 7        to the left,  improve=0.011279980, (0 missing)
      type6  splits as  RLRLRL,       improve=0.010156270, (0 missing)
      range1 < 225      to the right, improve=0.009807567, (0 missing)
      range2 < 225      to the right, improve=0.009807567, (0 missing)
  Surrogate splits:
      cost4  < 7        to the left,  agree=1.000, adj=1.000, (0 split)
      cost5  < 3        to the right, agree=0.880, adj=0.519, (0 split)
      cost6  < 3        to the right, agree=0.880, adj=0.519, (0 split)
      price3 < 0.666855 to the right, agree=0.752, adj=0.004, (0 split)
      price4 < 0.666855 to the right, agree=0.752, adj=0.004, (0 split)

Node number 2: 2253 observations,    complexity param=0.012105
  mean=3.366178, MSE=2.501066
  left son=4 (1453 obs) right son=5 (800 obs)
  Primary splits:
      type6  splits as  RLRLRL,       improve=0.01773332, (0 missing)
```

학습 과정 (모델 성능 평가)

- CART 알고리즘(회귀 트리) 성능 평가 (2/2)




```
> summary(car_CART_p)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  3.210   3.210   3.650   3.458   3.650   3.760
> summary(car_int_test$choice)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   3.000   3.000   3.499   5.000   6.000
> cor(car_CART_p,car_int_test$choice)
[1] 0.1148936
```

학습 과정 (모델 성능 평가)

Naivebayes 알고리즘

- 약 90%의 정확도를 보인다

c5.0 알고리즘 (DST)

- 학습률은 73%이며 약 28% 정확도를 보인다.

회귀트리 알고리즘(CART)

- cor() 함수를 이용하여 상관관계를 측정한 결과 0.114의 결과를 보인다.

◆ 이중 가장 높은 정확도를 보이는 Naivebayes 알고리즘을 사용한다!

학습 과정 (모델 성능 향상)

- 모델의 성능을 향상시키기 위해 모델을 훈련할 때 라플라스 측정기에 대한 값을 설정한다

```
car_Naive2_m <- naiveBayes(car_train,car_train$choice,laplace = 1)
car_Naive2_p <- predict(car_Naive2_m,car_test)
CrossTable(car_test$choice, car_Naive_p,prop.chisq = FALSE , prop.t = FALSE , dnn = c('actual', 'predicted'))
```

actual	predicted choice1	choice2	choice3	choice4	choice5	choice6	Row Total
choice1	243	4	18	1	2	25	293
	0.829	0.014	0.061	0.003	0.007	0.085	0.177
	0.924	0.035	0.040	0.008	0.004	0.152	
choice2	1	84	5	1	8	7	106
	0.009	0.792	0.047	0.009	0.075	0.066	0.064
	0.004	0.730	0.011	0.008	0.015	0.043	
choice3	6	5	418	7	31	24	491
	0.012	0.010	0.851	0.014	0.063	0.049	0.297
	0.023	0.043	0.935	0.056	0.058	0.146	
choice4	4	1	1	99	7	12	124
	0.032	0.008	0.008	0.798	0.056	0.097	0.075
	0.015	0.009	0.002	0.786	0.013	0.073	
choice5	4	15	4	15	488	0	526
	0.008	0.029	0.008	0.029	0.928	0.000	0.318
	0.015	0.130	0.009	0.119	0.905	0.000	
choice6	5	6	1	3	3	96	114
	0.044	0.053	0.009	0.026	0.026	0.842	0.069
	0.019	0.052	0.002	0.024	0.006	0.585	
Column Total	263	115	447	126	539	164	1654
	0.159	0.070	0.270	0.076	0.326	0.099	

라플라스 측정기의 값을 1로 설정한 후, 예측의 정확도는 약 86%이다. 이후 측정기의 값을 변화 해도 초기 설정 값(0) 보다 높은 정확도는 나오지 않는다.

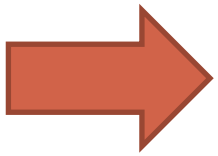
라플라스 측정기의 값이 0인 초기의 모델을 선정한다!

결과

사람들의 선호도와 선택을 학습하여 소비자의 취향에 따라 차를 추천해 주는 시스템

- **기대 효과**

 기업은 소비자의 취향 특징을 알아내, 그 특징을 부각시키거나 개발을 통하여 소비자들에게 어필 할 수 있으며, 소비자는 보다 객관적이고 정확하게 자신의 취향에 맞는 차를 추천 받아, 시간과 수고를 덜 수 있다.



기계 학습을 통하여 약 90%의 정확도를 가지는 모델을 생성하여 그 성능을 확인하였다 !

Q&A