# 중고 자동차 가격 예측 프로젝트

– 중고 자동차 판매 사이트 Encar 의 아반떼 차량 가격을 중심으로 –

202020140 박문식

201621053 이승민

# 목차

## 1. 주제 선정 이유

COVID-19 팬더믹으로 인해 세계 공급망은 큰 타격을 입었고, 기업들은 물건 생산에 큰 어려움을 입었다. 이는 자동차 업계도 피해갈 수 없었다. 팬더믹 초기 시장 예측치에 맞게 생산을 줄이던 반도체 업계는 회복되는 경제에 맞춰 생산량을 늘렸지만, 아직 완전히 생산량을 회복하지 못했다. 특히 차량용 반도체는 단가가 낮은 점으로 인해 생산 우선순위에서 밀렸다. 설상가상으로, 차량용 반도체가 정상적으로 생산되더라도 반도체를 패키징하는 업체들이 있는 남아시아 국가들에서 확진자가 무더기로 발생하며 강도 높은 봉쇄 조치가 이루어졌다. 이 때문에 완성차 업계에서 필요로 하는 반도체 어셈블리의 공급량이 급감했고, 차량 생산량은 고전을 면치 못했다. 그 결과, 차량을 구입하려는 사람들은 최장 12 개월까지 기다려야 차를 받을 수 있게 되었다. 자연스럽게 중고 자동차 가격은 폭등했다. 중고차가 신차 가격을 뛰어넘는 사례까지 나오고 있다.

이런 배경을 바탕으로 우리 조는 중고차 가격에 관심을 가지게 되었다. 중고차 가격은 연식과 운행 거리, 사고 여부 등에 뚜렷한 상관 관계를 보이는 특성이 있다. 이를 활용하여 우리 조는 여러 데이터를 통해 적정한 중고 자동차 가격을 예측하는 모델을 만들어 진짜 가격과 비교하고자 한다.

## 2. 데이터 수집 및 가공

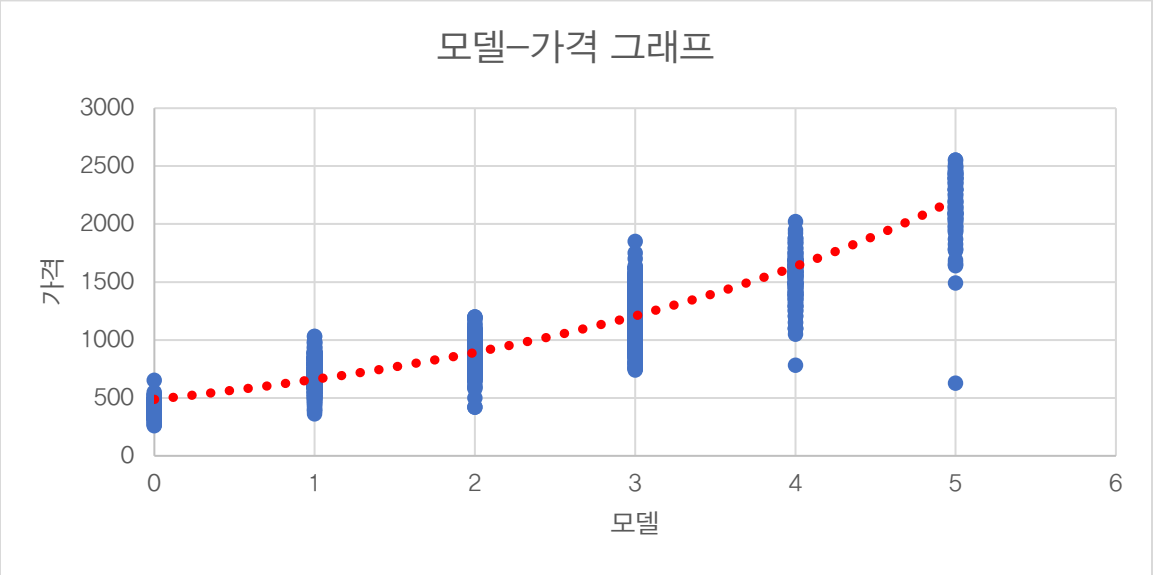
<table>
<tr><th>Gen</th><th>Trim</th><th>Year</th><th>Km</th><th>Color</th><th>Accident</th><th>Price</th></tr>
<tr><td>Nominal</td><td>Nominal</td><td>Nominal</td><td>Numeric</td><td>Nominal</td><td>Nominal</td><td>Numeric</td></tr>
<tr><td>아반떼 HD</td><td rowspan="2">저급</td><td>2007</td><td>1</td><td rowspan="2">흰색</td><td rowspan="3">없음</td><td>1</td></tr>
<tr><td>아반떼 MD</td><td rowspan="4">~</td><td rowspan="4">~</td><td rowspan="4">~</td></tr>
<tr><td>아반떼 AD</td><td rowspan="2">중급</td><td rowspan="2">검정색</td></tr>
<tr><td>더 뉴 아반떼</td><td rowspan="3">있음</td></tr>
<tr><td>더 뉴 아반떼 AD</td><td rowspan="2">고급</td><td rowspan="2">유채색</td></tr>
<tr><td>아반떼 CN7</td><td>2021</td><td>250,000</td><td>2,500</td></tr>
</table>

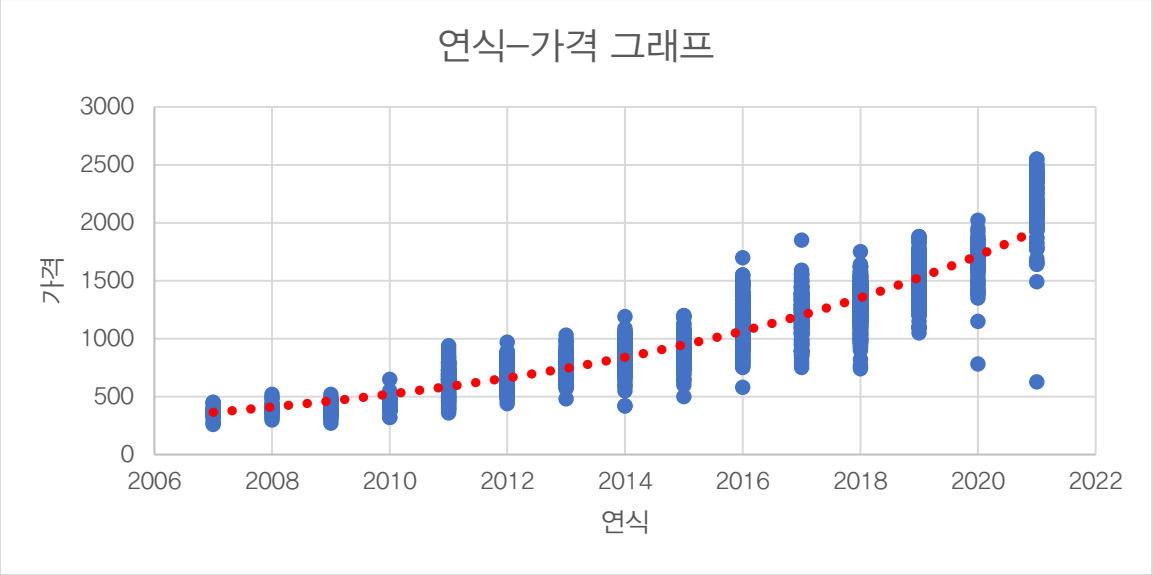
우리 조는 모델링의 편의를 위해 우선 중고차 판매 사이트 Encar 의 “아반떼” 차량으로 도메인을 한정했다. 해당 중고차 판매 사이트의 여러 매물들을 둘러보며 매물 가격에 가장 영향을 많이 끼치는 인자가 차량 운행거리(km) 와 연식(년)임을 확인했다. 하지만 이 두 인자만으로 데이터 마이닝을 하기엔 부족했기에, 중고 자동차 구매에 대한 인터넷 게시글 및 유튜브 영상 등을 보면서 가장 중요한 두 속성 외 어떤 인자를 보는 것이 중요한지 알아보았다. 그렇게 해서 우리는 모델명, 트림, 색상, 사고여부, 색, 주행 거리의 6 개의 속성을 선택했다.

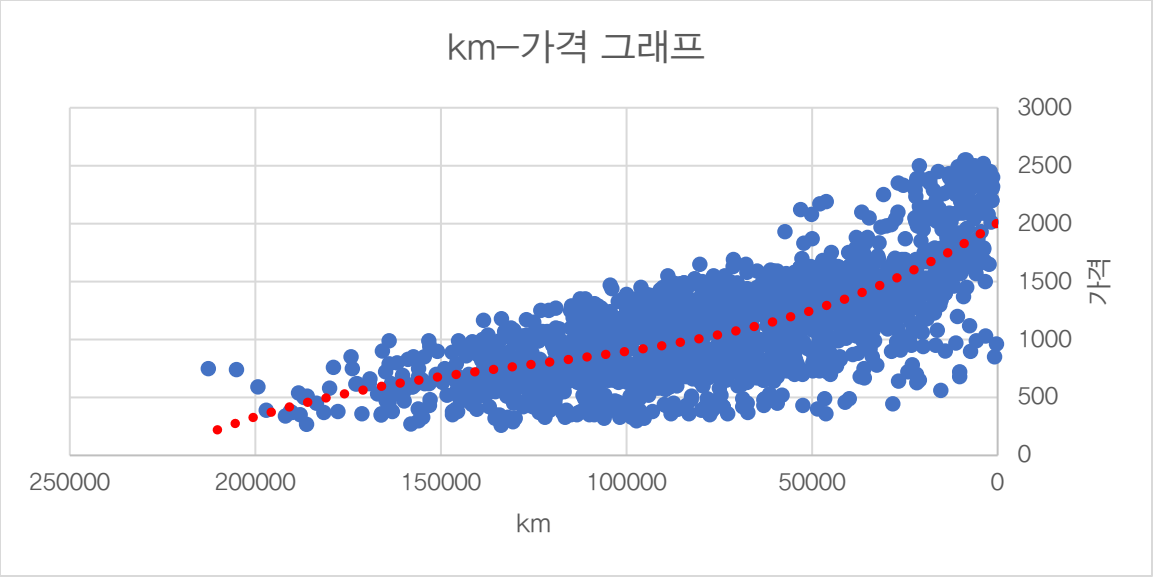
선택된 인자들을 데이터 마이닝에 적합하게 수정하는 것은 많은 반복 작업을 요구했다. 약 80 가지 문자열로 뒤섞여 있던 모델명과 트림을 각각 5 개의 모델명과 3 가지 트림으로 분류하였다. 색상과 사고 여부 속성에 대해서는 고민이 있었다. 자동차의 색상은 흰색과 검정색에 편중되는 경향이 있고, 흰색과 검정색이 아닌 차에 대해서는 중고가가 낮은 경향이 있다. 그래서 색은 흰색/검정색/유채색의 3 가지로 분류하기로 했다. 사고 여부 데이터는 크롤링했을 때 각각 사고 부위별로 어떤 수리를 진행했는지에 대한 데이터가 수집되었는데, 이는 자동차 수리에 대한 전문적인 지식을 가지고 있지 않아 사고가 있었는지 없었는지에 대해서만 분류했다.

그 결과, 2080 개의 데이터를 확보할 수 있었다.

## 3. Modeling


모델–가격 그래프
3000
2500
2000
1500
1000
500
0
가격
0
1
2
3
4
5
6
모델


연식–가격 그래프
3000
2500
2000
1500
1000
500
0
가격
2006
2008
2010
2012
2014
2016
2018
2020
2022
연식




데이터를 시각화한 자료를 보면, 특성들이 가격에 대해 뚜렷한 상관 관계를 보이는 것을 알 수 있다. 이를 확인한 후 본격적인 데이터 마이닝을 진행하였다. 데이터의 상관 관계를 잘 활용할 수 있는 Linear Regression 과 Multilayer Perceptron 알고리즘을 사용하기로 했다.

```
=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9673
Mean absolute error                     80.2791
Root mean squared error                111.7233
Relative absolute error                 22.8557 %
Root relative squared error             25.3759 %
Total Number of Instances             2080
```

*Linear Regression Algorithm w/ 10-fold cross-validation*

우선, Linear Regression 을 사용한 결과, RMSE 값이 111 로 나오는 것을 확인할 수 있었다. 이는 알고리즘이 ±111 의 오차 범위로 가격을 예측할 수 있다는 뜻이다. 또, 피어슨 상관 계수가 0.96 으로 1 에 매우 가깝게 나왔다. 이를 통해 데이터셋이 선형에 가깝다는 것도 수치로 확인할 수 있었다.

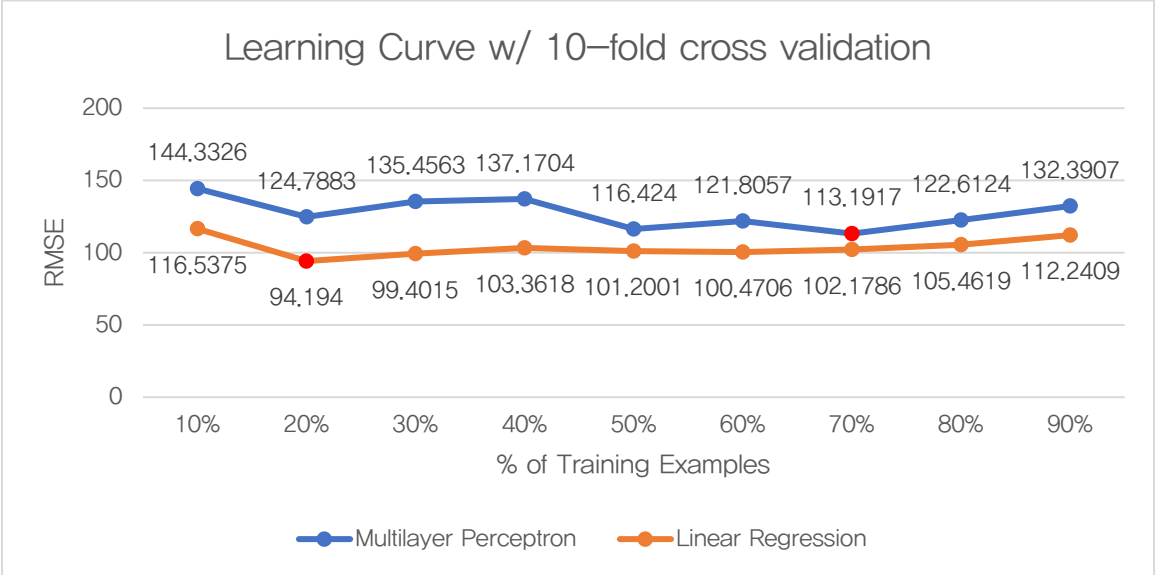
```
=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9648
Mean absolute error                     83.8127
Root mean squared error                116.6878
Relative absolute error                 23.8617 %
Root relative squared error             26.5035 %
Total Number of Instances             2080
```

*Multilayer Perceptron Algorithm w/ 10-fold cross-validation*

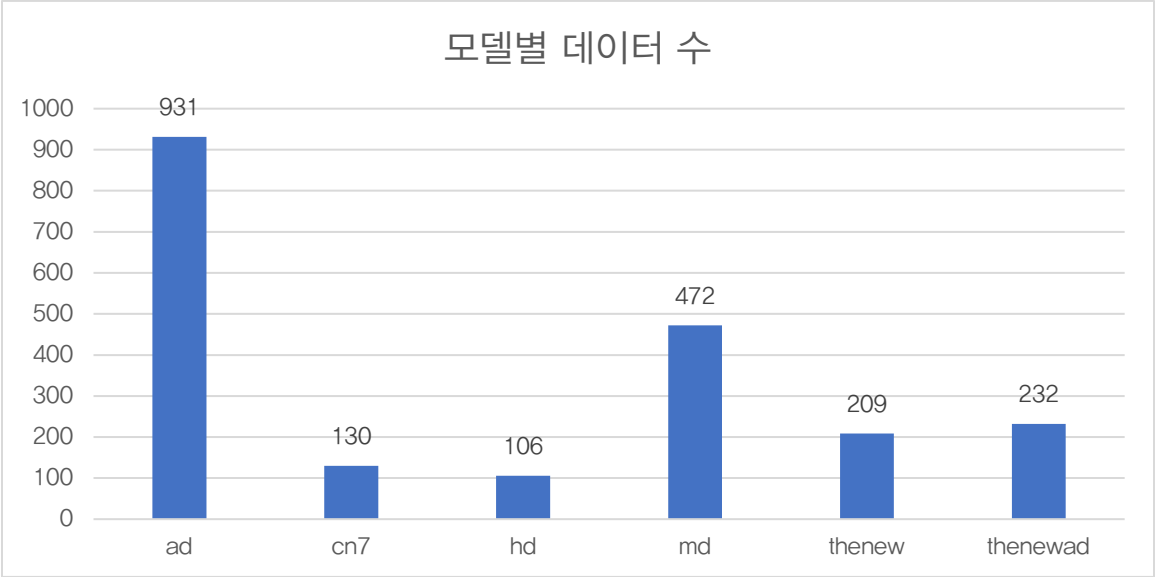
다음으로, Multilayer Perceptron 을 사용한 결과, RMSE 값이 116 으로 나오는 것을 확인할 수 있었다.

이를 통해 전체 데이터를 사용했을 때 Linear Regression 이 Multilayer Perceptron 보다 근소하게 성능이 더 좋다는 점을 알 수 있었다.

## 4. Learning Curve

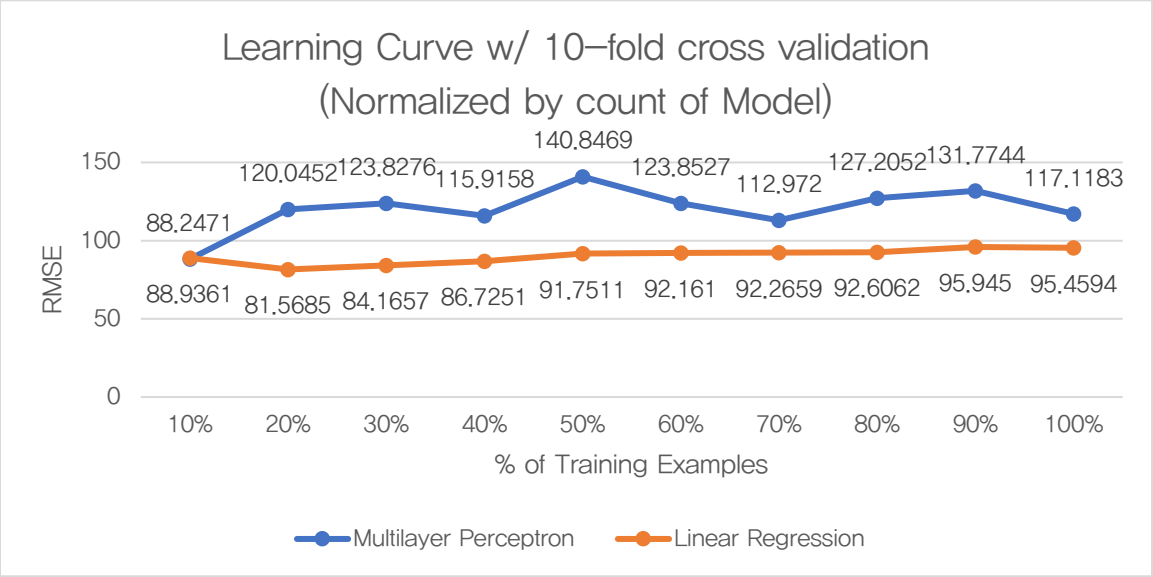



원본 데이터셋에서 각각 N%의 데이터를 추출하여 학습을 진행하였고, 이에 따른 러닝 커브 그래프를 그렸다. 모든 구간에서 Linear Regression 알고리즘을 이용했을 때 Multilayer Perceptron 알고리즘을 이용했을 때보다 더 정확도가 높은 것을 알 수 있었다. 또, Linear Regression 은 20% 구간에서, Multilayer Perceptron 은 70% 구간에서 제일 정확도가 높은 것을 확인할 수 있었다.




테스트를 하던 도중, 특정 모델의 가격은 잘 예측하는 반면 특정 모델의 가격은 잘 예측하지 못하는 문제를 발견했다. 왜 그런지 생각해 본 결과, 잘 예측되는 모델의 개수가 잘 예측되지 못하는 모델의

데이터 수보다 약 9 배가량 차이가 나는 현상을 발견했다. Overfitting 된 것이다. 그렇다면, 만약 가장 적은 모델의 수로 맞춘 후 학습을 진행하면 해결할 수 있지 않을까 하는 생각에 모델별 데이터 수를 106 개로 통일한 후 진행해 보았다.




이 결과, 평균 RMSE 값이 10% 감소하는 것을 확인할 수 있었다.

## 5. ANOVA Test 결과

앞서 러닝 커브를 그리면서 Linear Regression 은 20% 구간에서, Multilayer Perceptron 은 70% 구간에서 제일 정확도가 높은 것을 확인할 수 있었다. 이를 토대로 ANOVA Test 를 진행하였다.

알고리즘은 Linear Regression 과 Multilayer Perceptron, 각 알고리즘마다 5 개를 적용하였다.

| % Of Training Data | RMSE (Linear Regression) | % Of Training Data | RMSE (Multilayer Perceptron) |
|---|---|---|---|
| 68% | 102.3256 | 18% | 136.8779 |
| 69% | 102.5077 | 19% | 113.4935 |
| 70% | 102.1786 | 20% | 124.7883 |
| 71% | 101.7474 | 21% | 142.9429 |
| 72% | 101.71 | 22% | 119.095 |

위 표를 토대로 ANOVA Test 를 90%, 95%, 99% 구간에서 진행하였고, 모두 F 비의 값이 F 기각치보다 크므로 귀무가설을 기각할 수 있었다. 이를 통해 모델의 성능을 확인할 수 있었다.

분산 분석: 일원 배치법 90%신뢰도

요약표

| 인자의 수준 | 관측수 | 합 | 평균 | 분산 |
|---|---|---|---|---|
| linear regression | 5 | 510.4693 | 102.0939 | 0.124883 |
| multilayer perceptr | 5 | 637.1976 | 127.4395 | 150.1473 |

분산 분석

| 변동의 요인 | 제곱합 | 자유도 | 제곱 평균 | F 비 | P-값 | F 기각치 |
|---|---|---|---|---|---|---|
| 처리 | 1606.006202 | 1 | 1606.006 | 21.37463 | 0.001703 | 3.457919 |
| 잔차 | 601.0887947 | 8 | 75.1361 | | | |
| 계 | 2207.094997 | 9 | | | | |

분산 분석: 일원 배치법 95%신뢰도

요약표

| 인자의 수준 | 관측수 | 합 | 평균 | 분산 |
|---|---|---|---|---|
| linear regression | 5 | 510.4693 | 102.0939 | 0.124883 |
| multilayer perceptr | 5 | 637.1976 | 127.4395 | 150.1473 |

분산 분석

| 변동의 요인 | 제곱합 | 자유도 | 제곱 평균 | F 비 | P-값 | F 기각치 |
|---|---|---|---|---|---|---|
| 처리 | 1606.006202 | 1 | 1606.006 | 21.37463 | 0.001703 | 5.317655 |
| 잔차 | 601.0887947 | 8 | 75.1361 | | | |
| 계 | 2207.094997 | 9 | | | | |

분산 분석: 일원 배치법 99%신뢰도

요약표

| 인자의 수준 | 관측수 | 합 | 평균 | 분산 |
|---|---|---|---|---|
| linear regression | 5 | 510.4693 | 102.0939 | 0.124883 |
| multilayer perceptr | 5 | 637.1976 | 127.4395 | 150.1473 |

분산 분석

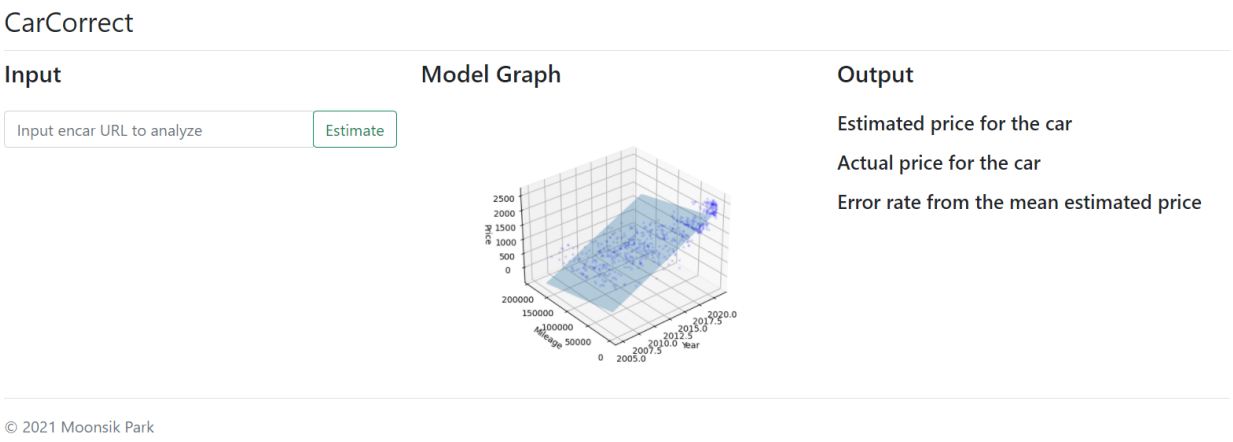
| 변동의 요인 | 제곱합 | 자유도 | 제곱 평균 | F 비 | P-값 | F 기각치 |
|---|---|---|---|---|---|---|
| 처리 | 1606.006202 | 1 | 1606.006 | 21.37463 | 0.001703 | 11.25862 |
| 잔차 | 601.0887947 | 8 | 75.1361 | | | |
| 계 | 2207.094997 | 9 | | | | |

## 6. 결과 시각화

ANOVA 테스트까지 진행하여 모델의 성능을 확인한 후, 결과를 시각화하는 프로그램을 개발했다. 아래와 같은 툴을 활용하였다.

| Backend Language | Python 3.10.1 |
|---|---|
| Backend Server Framework | Flask 2.0.2 |
| Backend Crawling Framework | requests 2.26.0 |
| Backend Graph Visualization Framework | matplotlib 3.5.0 |
| Backend Machine Learning Library | scikit-learn 1.0.1 |
| Frontend Language | HTML, CSS, JavaScript(jQuery) |

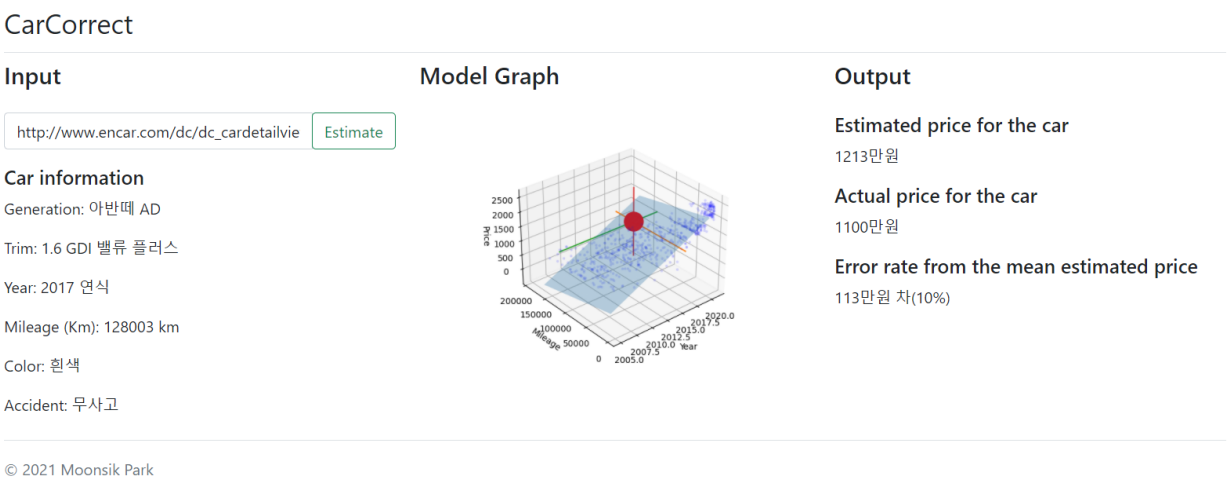
시간상의 제약으로 프로그램의 기능은 처음에 생각했던 것보다 훨씬 간단하게 구현하였다. 우선, 프로그램을 실행하면 프로그램이 구동되어 모든 데이터셋에 대해 Linear Regression 알고리즘으로 학습을 진행한다. 모델은 메모리에 저장되고, 데이터셋은 가공되어 연식, km 수, 가격의 3 축 그래프로 그려진다. 준비 작업이 모두 완료되면 사용자의 입력을 받기 전까지 대기한다.




*입력 대기중인 프론트엔드*

사용자의 입력은 Encar 사이트의 URL 의 형식으로 받는다. URL 을 입력받으면, 파싱하여 차량의 고유 번호를 추출한다. 그 후, 사이트에서 해당 번호의 차량의 정보를 크롤링해 가져온다. 이 과정이

완료되면 학습된 모델에 입력하여 예측값을 산출한다. 예측값은 3 축 그래프에서 빨간색 점으로 표시된다. 프론트엔드는 차량의 정보와 예측값 그리고 그래프를 백엔드로부터 받아 보여준다.




*예측값을 보여주는 모습*

## 7. 결론 및 소회

데이터 마이닝을 처음 접한 초보자의 입장에서, 인터넷에서 쓸모 있는 데이터를 수집하고 가공해 유의미한 결과를 나타낸다는 것은 매우 어려운 일임에 틀림없다. 이는 데이터 마이닝 알고리즘이 모두 공개되어 API 호출로 알고리즘을 한 번에 돌릴 수 있는 지금도 마찬가지다.

초보자가 데이터 마이닝에 어려움을 겪는 이유는 크게 세 가지로 구분할 수 있다. 첫째로, 데이터 선택의 어려움이다. 우리가 현실 세계에서 마주치는 데이터들 중에서 독립시행이고, 상관 관계가 매우 뚜렷하거나 연관 관계를 잘 나타낼 수 있는 데이터를 찾기는 매우 어렵다. 우리 조는 처음에 이승민 팀원이 '리그 오브 레전드'라는 게임에서 게임 통계를 바탕으로 이 사람의 게임 내 계급을 예측하려고 했다. 하지만 해당 데이터는 독립시행이 아닌 종속시행 데이터였다. 즉, 게임 내 계급이 낮은 사람들은 낮은 사람들끼리, 높은 사람들은 높은 사람들끼리 매칭되어 게임을 진행하기 때문에 게임 내 계급이 낮은 어떤 사람이 어떤 게임에서 좋은 성적을 얻었다 하더라도 그 사람이 게임 내 계급이 높은 사람이 낮은 성적을 얻은 것과 비교를 할 수 없는 것이었다. 이를 보정하려면 우리가 모르는 게임 내부 통계를 알아야 했기에, 해당 데이터를 포기해야만 했다. 독립시행이고, 상관 관계가 매우 뚜렷한 데이터를 찾는 중 박문식 팀원이 제안한 중고차 시장의 매물 가격 예측을 하기로 했다.

둘째로, 데이터 마이닝에 데이터를 찾았다 하더라도 학습할 때 제공받는 데이터셋과 달리 가공이 되어 있지 않는 점이다. 데이터 가공은 과정 자체는 매우 간단하다. 하지만 그 과정에서 데이터를 어떻게 가공할지 의사결정을 하는 것은 매우 어렵다. 해당 데이터가 어떻게 산출되었는지, 그리고 어떤 방식으로 가공했을 때 데이터 마이닝 알고리즘이 잘 받아들일 수 있는지에 대한 깊은 이해가 필요하다. 우리 조에서는 차량의 사고 정보에 관한 데이터를 가공할 때, 가공 전 데이터에는 차량의 사고 부위에 대한 자세한 설명이 있었지만 이에 대한 전문적인 지식이 없어 단순히 사고 유무로만 전처리를 할 수밖에 없었다.

마지막으로, 데이터 마이닝 결과에 대한 이해이다. 개인적으로 Weka 의 경우 GUI 의 어려움으로 인해 제대로 데이터 마이닝을 진행해 놓고서도 마이닝이 잘 된 것인지를 알 수가 없었다. 이는 우리가 Weka 에 며칠간 적응한 후 해결되었다. 그럼에도 Weka 를 Java 외 다른 프로그래밍 언어에서 사용하기가 너무 어렵고, 예전부터 익숙했던 프레임워크를 쓰는 것이 좋을 것 같아 박문식 팀원은

scikit-learn을 이용해 데이터 마이닝을 진행하였고, 이승민 팀원은 Weka를 이용해 데이터 마이닝을 진행하였다.

이번 데이터마이닝 프로젝트에서, 우리는 프로젝트가 요구한 대로 직접 데이터를 찾아 수집 및 전처리 과정을 거쳐 RMSE 100 정도의 유의미한 예측을 하는 모델을 만들 수 있었다. 프로젝트를 진행하면서, 이론적으로 완벽한 데이터만 접해왔던 것과는 달리 알고리즘에 잘 맞지도 않고 논리적으로 설명 불가능한 오차도 존재하는 실제 데이터를 접했다. 배운 것과는 다르게, 러닝 커브 그래프가 들쑥날쑥하고 하는 등 잘못된 부분도 있었던 것 같다.

개인적으로 이 프로젝트에 소비할 시간이 더 많았더라면 좋았을 것 같다는 생각이 든다. 예를 들면, 차량 모델에 대해 오버피팅이 된 것을 모델별 RMSE라는 객관적인 수치로 나타내고, 모델의 수가 얼마 없으니 모델별로 학습을 해 RMSE를 구했다면 전체 평균 RMSE가 훨씬 더 낮은 값이 나오지 않았을까 하는 생각이 든다. 또, 지금은 우리가 가격과 연관성이 있으리라 상상한 6가지 속성을 선정했지만, 좀 더 많은 항목을 선택해 전처리해 두고 가격과의 피어슨 상관 관계 계수라는 객관적인 수치를 확인해 항목을 선정했으면 하는 아쉬움도 든다.

프로젝트를 하면서 나온 결과를 보여주는 프론트엔드를 만드는데 시간이 많이 걸렸다. 프론트엔드는 빠른 프로토타이핑이 가능한 Python과 Javascript를 사용했는데도 불구하고 시간이 많이 걸려 원래 계획한 기능보다 훨씬 더 현실적으로 기능을 빼서 개발할 수밖에 없었다. Python과 Weka를 연동하려고 했으나, JVM 연동의 불안전성과 속도 등의 문제로 인해 사용하지 않고 scikit-learn을 도입했다. 이 과정에서 scikit-learn에 Weka와 똑같은 training 환경을 구축하는데 애를 먹었다. matplotlib-3d 라이브러리를 이용해 3축 그래프로 시각화를 시도했는데, 이 또한 매우 어려운 작업이었다. 3축 그래프를 그릴 때 가격, 마일리지 그리고 연식을 2차원 평면으로 나타내고, 상관 관계를 좀 더 뚜렷하게 보이도록 하기 위해 Linear Regression 알고리즘을 2개의 인자를 활용해 연식과 주행 거리에 대한 평면을 그렸다. 그래프가 보여지는 각도를 정하는 작업도 처음 해보는 작업이었다.

결과적으로 이번 프로젝트를 통해 팀은 많은 것을 배웠다. 직접 데이터를 수집해 전처리하고, 모델링하고 그 결과를 분석한 것은 매우 값진 경험이었다. 다른 팀의 프로젝트보다 좀 더 현실적인

데이터로 시각화까지 잘 진행한 것 같아 뿌듯하다. 이는 팀원의 적극적인 협동과 서로의 시간을 양보하여 프로젝트에 투자했기에 가능했다.