**Salary Range Predictive Analysis**

**based on 2018 StackOverflow survey data**

BUSS 256: Intro to Big Data analysis and interpretation

차현권

2018952150

**1장: 서론**

**URL:** <https://www.kaggle.com/stackoverflow/stack-overflow-2018-developer-survey>

**Data size:** 98855 rows X 129 columns

**Data description:**

StackOverflow는 개발자들이 서로 질문하고 답하는 사이트로, 네이버 지식인과 흡사한 형태의 웹사이트이다. 본 데이터는 StackOverflow에서 2018년 실시한 설문조사 데이터이다. 데이타에는 응답자의 나이 및 성별 등 개인 신상정보도 포함되어있지만, 프로그래밍과 관련한 질문들이 집중되어 있다. (예시: Do you contribute to open source projects?). 각 열은 한개의 질문에 해당하고, 각 열은 한 응답자의 답변에 해당하여 총 129개의 행과 98,855개의 열로 이루어져 있다. 질문중에는 여러개의 선택지를 다중선택할 수 있는 질문들이 섞여있고, 그러한 질문들의 답변은 세미콜론(;)으로 이어져 한 문장으로 기록되어 있다.

**Class and attributes:**

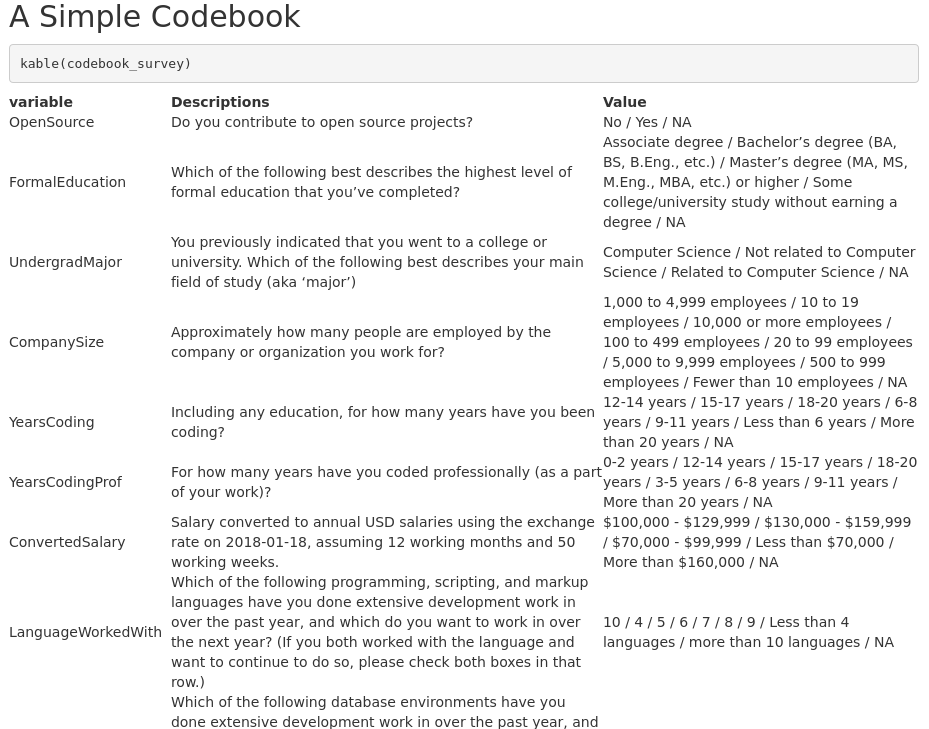
이번 Predictive Analysis를 통해 알아보고 싶은 것은 한 개발자가 받을 만한 연봉의 범위이다. 연봉과 관련된 여러개의 행이 있지만, 그 중에서 가장 규격화된 ConvertedSalary라는 행을 Class로 삼았는데 이 행은 응답자가 1년동안 50주를 일 한다고 가정하고 받는 연봉을 2018년 1월 18일 기준 USD의 환율을 토대로 계산하여 응답한 행이다. 이 Class를 위한 Attributes로는 129개의 행 중 연봉과 명확히 연관이 없어보이는 다수의 행들을 제하고, 나머지는 Descriptive Analysis를 통해 골라내었다.

**2장: Code book**

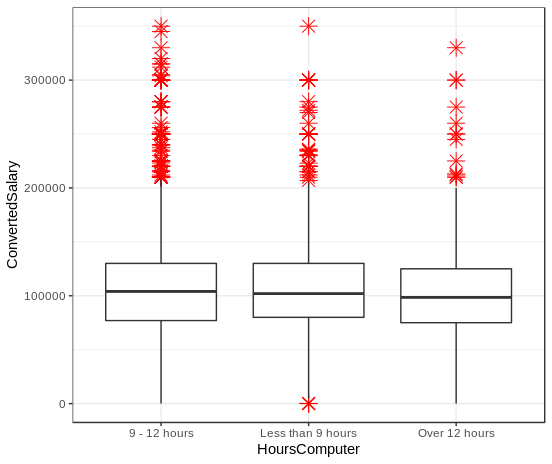
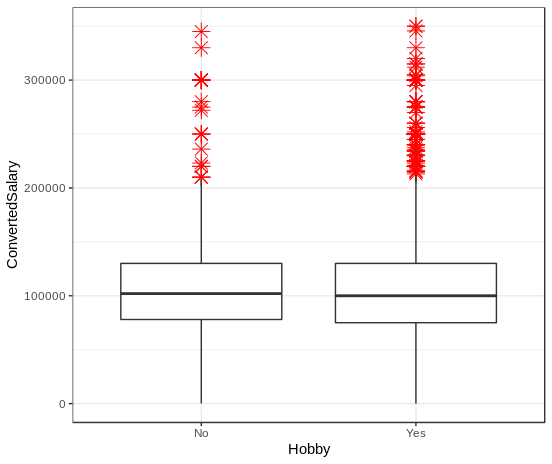
**Cleansing:**

* 일단 Descriptive analysis에 앞서 그냥 보기에도 연봉과 무관하여 보이는 행을 잘라내었다.
* 모델링의 정확도를 높이기 위해 연봉을 예측할 대상을 더욱 구체적으로 한정시켰다.
  + 10만여명중 다수를 차지한 미국 거주인원으로 한정
  + 10만여명중 7만여명인 Full-time employ로 한정
  + 가장 많이 선택된 직업군인 Back-end developer로 자신을 describe한 응답자로 한정
* NA values를 가진 열들을 잘라내었다.
* 모든 variable이 categorical이기 때문에 variable당 각 그룹의 빈도수가 전체 크기의 최소 5% 이상이 되게끔 regroup함
* ConvertedSalary의 경우, 최대 $2,000,000까지 였지만 $350,000이 대부분의 데이터를 커버하기때문에 $350,000이하로 한
* 다중선택이 가능한 질문의 경우, 선택지의 갯수를 세어 그룹별로 나눔

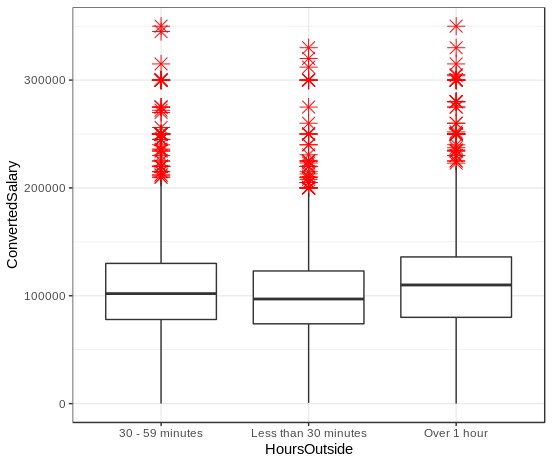
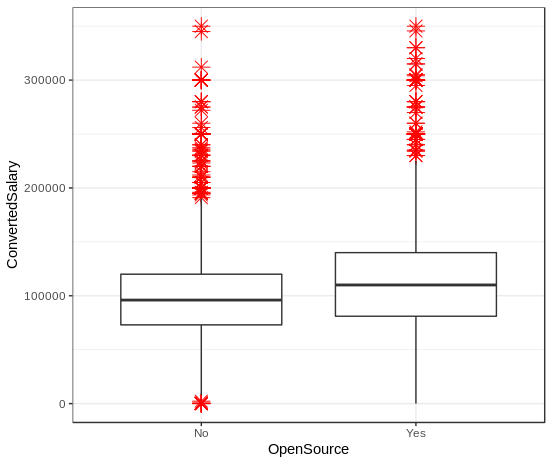
**A snippet of Code book (too long to load the entire code book)**



**3장: Descriptive Analysis**



위 두 그래프들은, 프로그래밍을 취미로 하는지 여부와 컴퓨터 앞에서 보내는 시간은 연봉과 상관관계가 없거나 미미하다는 것을 보여준다.



위 두 그래프는 오픈소스 프로젝트 참여 여부와 바깥에서 보내는 시간은 연봉과 의미있는 상관관계가 있음을 보여준다.

이와 같은 방법으로 연봉과 상관관계가 있어보이는 행들만을 남기고 모델링을 실시하였다.

**4장: Predictive Analysis**

**1. Visualization**

의미있는 Visualization이 없어 스킵한다.

**2. Pre-processing**

모든 variable이 categorical variable 이었기 때문에 “One hot encoding”이라고 알려진 방식에 따라 dummy variables을 만드는pre-processing을 했다.

**3. Data splitting**

트레이닝 데이타와 테스트 데이타는 7:3의 비율로 나눴다.

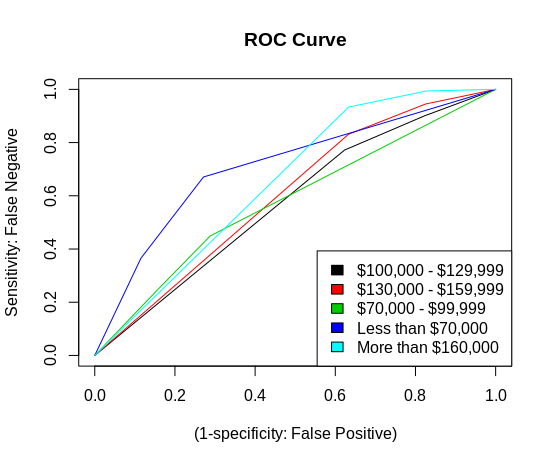
**4. Modeling**

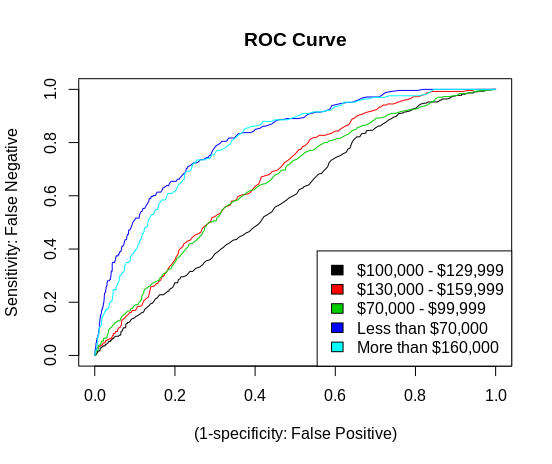
모델링에는 Random Forest, R partition, Sub vector machine, 그리고 Neural Net 알고리즘을 사용하였고, tuning을 위해서 다음과 같은 trControl을 사용하였다.

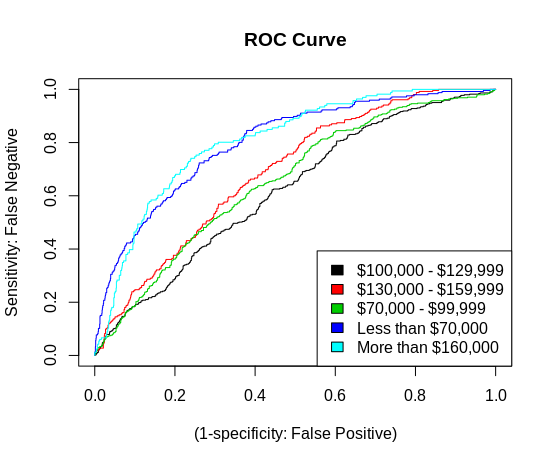
> trControl = trainControl(method="repeatedcv", number=5, repeats=3)

**5. Model Evaluation**

**AUC graph**

  
Illustration 2: R partition AUC

  
Illustration 1: Random Forest AUC

  
Illustration 3: Neural Net AUC

SVM을 이용한 모델의 AUC는 알수없는 오류로 인해 구하지 못했음.

**Confusion Matrices**

Random Forest - Accuracy (average) : 0.3715

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pred\Ref | < $70k | $70k - $100k | $100k - $130k | $130k - $160k | > $160k |
| < $70k | 5.1 | 3.4 | 1.3 | 0.2 | 0.1 |
| $70k - $100k | 7 | 13.5 | 8.6 | 3.2 | 1.4 |
| $100k - $130k | 3.2 | 9.2 | 13.5 | 8 | 5 |
| $130k - $160k | 0.4 | 1.8 | 3.3 | 3.1 | 2.4 |
| > $160k | 0.1 | 0.7 | 1.8 | 1.8 | 2 |

R partition - Accuracy (average) : 0.3389

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pred\Ref | < $70k | $70k - $100k | $100k - $130k | $130k - $160k | > $160k |
| < $70k | 1.9 | 2.1 | 0.7 | 0.2 | 0 |
| $70k - $100k | 8.4 | 11.2 | 6.9 | 2.4 | 0.9 |
| $100k - $130k | 5.6 | 15.3 | 20.8 | 13.8 | 9.8 |
| $130k - $160k | 0 | 0 | 0 | 0 | 0 |
| > $160k | 0 | 0 | 0 | 0 | 0 |

Subvector Machine - Accuracy (average) : 0.3923

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pred\Ref | < $70k | $70k - $100k | $100k - $130k | $130k - $160k | > $160k |
| < $70k | 6.4 | 4 | 1.2 | 0.2 | 0.1 |
| $70k - $100k | 6.4 | 13.7 | 9.1 | 2.9 | 1.1 |
| $100k - $130k | 2.5 | 8.5 | 13.5 | 8.1 | 4.6 |
| $130k - $160k | 0.4 | 4 | 1.2 | 3.1 | 4.6 |
| > $160k | 0.2 | 0.8 | 1.8 | 2.1 | 2.5 |

Neural Net - Accuracy (average) : 0.3708

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Pred\Ref | < $70k | $70k - $100k | $100k - $130k | $130k - $160k | > $160k |
| < $70k | 5.3 | 3.4 | 1.1 | 0.2 | 0.1 |
| $70k - $100k | 7.3 | 13.5 | 2.8 | 9.1 | 1 |
| $100k - $130k | 3 | 10.4 | 15.3 | 10.4 | 6.7 |
| $130k - $160k | 0.2 | 0.9 | 2.2 | 2.2 | 2.1 |
| > $160k | 0.1 | 0.3 | 0.8 | 0.8 | 0.9 |

**Concluion:**

R partition을 사용한 모델은 $100k- $130k구간에서 상대적으로 압도적인 accuracy를 보여주고 있지만 $130k -$160k 과 > $160k구간을 전혀 예측하지 못했다. 나머지 세 알고리즘을 이용한 모델은 전 구간별로 비슷한 accuracy를 보여주고 있고, 그중에서도 SVM를 사용한 모델이 근소한 차이로 가장 성능이 좋게 나왔다.

**6장: 결언**

인적자원에 돈을 얼마나 쓸 것인가는 계산하기 까다로운 문제이다. 정답이 정해진 것도 아니고 가지고 있는 인적자원의 진정한 가치를 판단하는일은 단순한 셈으로 할 수없는 일이다. 하지만 구체적인 설문을 통해 한 개인에게 적당한 연봉이 얼마인지 판단할 수 있다면, 이는 회사와 개인 모두에게 획기적인 일이될 것이다. 안타깝게도 결과가 의미를 가질만큼 좋지는 않았다. 하지만 original data에서 의미있는 variable를 고르는 과정, 다중답변을 수치화하는 과정, 그리고 categorical variable들을 pre-processing하는 과정에서 개선의 여지가 있음을 참작한다면 모델들의 예측 정확도는 더 올라갈 수 있을 것이다.