**<Team6 – Upper Class Classification modeling>**

2013130336 박창훈

2013120315 최현호

2014190042 김세준

2015190001 황지수

1. **Introduction**

본 프로젝트는 “adult.csv” 데이터를 통해 어떤 사람이 상류층에 속하는지 판별하는 binary classification 모델을 만드는 것을 목표로 합니다. 기본적인 binary classification이지만 데이터의 몇 가지 특성으로 인해 실용성 있는 모델을 만드는 것이 어렵습니다. categorical data와 numerical data가 혼재되어 있다는 것, 모집단의 비율이 1:2이 아닌 1:3이라는 것, census 특성상 missing data와 artifact가 많다는 것이 모델링을 어렵게 하는 대표적인 특징들입니다. 데이터에서 상류층 비율이 25%밖에 안되기 때문에 zeroR baseline 기준으로 모든 값을 0으로 예측해도 accuracy가 75%가 나옵니다. 그러나 데이터의 출처인 Kaggle에서 만든 가장 성공적인 모델들조차 찍는 수준의 baseline보다 성능이 낮습니다. (logistic regression 기준 0.61, MLP기준 0.71)

데이터에서 요구하는 목표는 간단하지만 이를 달성하기 위해서는 Data science 수업에 등장하는 correlation, missing data, artifact, normalization, outlier 처리들이 모두 등장하며, categorial data와 numerical data를 같이 사용하기 위해서 one-hot encoding이 필요합니다. 본 프로젝트는 Data science pipeline에 따라 적절한 절차에 맞춰 프로젝트를 진행하였습니다. 우선 앞서 발표한 문제 정의 (interest)와 데이터 수집(get data)에 대해 설명하고 데이터의 기본적인 특징들을 살펴볼 것입니다. (explore data) 그 후 baseline에 따라 모델의 목표를 설정하고 data를 모델에 맞는 형태로 처리할 것입니다. (cleaning data) 이를 기반으로 총 5가지의 모델을 train, validate, test하여 성능을 분석할 것입니다. 끝으로 결과를 시각화하고(visualize), 이를 통해 한계가 무엇이고 어떻게 더 발전시킬 수 있는지 정리할 것입니다. (communicate)

1. **Interest(problem)**

대부분의 사람들은 경제적으로 윤택한 삶을 꿈꿉니다. 오래도록 바쁘게 일하거나 열심히 공부하면, 언젠가 자신도 남부럽지 않은 부를 누릴 수 있다는 것이 많은 사람들의 믿음입니다. 그런데 노력과 학력이 아니더라도 우리 삶에 영향을 미치는 것들은 정말 다양합니다. 어떤 사회에서는 특정 인종이나 성별로 태어날 때 삶에 훨씬 많은 기회가 주어진다고 여겨집니다. 저희는 데이터를 통해 세상을 바라보면서, 사람들이 당연하게 믿어온 상식을 되돌아보고자 합니다. 실제로 고소득층인 사람과 그렇지 못한 사람들을 공식적으로 조사하여, 그들이 앞서 말한 근무시간, 노력, 인종, 성별, 학력 뿐만 아니라 어떤 요소를 지니고 있는지 알아본 방대한 자료가 있습니다. 이를 바탕으로 우리는 모델을 만들고, 한 사람의 데이터가 주어졌을 때 그 사람이 과연 정말 고소득층인지를 알아맞혀보려 합니다.

1. **get data**

저희는 1994년 미국 센서스 데이터를 Barry Becker가 추출하여 가공한 데이터를 선택하였습니다. (https://www.kaggle.com/uciml/adult-census-income) 각각 15개의 feature를 가진 32561개의 샘플로 이루어져 있습니다. 저희가 이 데이터를 선택한 이유는 2가지입니다. 우선 저희가 원하는 상류층을 판단하는 목적에 부합합니다. 경제적 상류층이 될 수 있는가를 판단하기 위해서는 다양한 요소들이 필요한데 해당 데이터는 총 15가지의 다양한 feature들을 보유하고 있습니다. 다음으로 데이터의 양과 질입니다. 사회조사 방법론 상 원하는 형태의 정형화된 자료들을 대량으로 확보하기 어렵습니다. 표본의 특성부터 조사 환경, 시대적 상황 및 질문지와 조사방법의 구성 등 구조작으로 다양한 artifact가 개입될 여지가 큽니다. 그렇기에 진부할 수 있는 주제지만 실제로 신뢰성 있는 실험을 할 수 있는 자료를 얻기 어렵습니다. 해당 자료는 3만명 이상의 sample들을 보유하면서 missing data가 10% 미만이며 조사 상황이 분명하게 지시되어 있기 때문에 artifact 발견, missing data 처리, 표준화가 및 outlier 판단이 편리합니다.

저희는 다음15개의 feature 중 14개(Age, WorkClass, fnlwgt, Education, EducationNum, MaritalStatus, Occupation, Relationship, Race, Gender, CapitalGain, CapitalLoss, HoursPerWeek, NativeCountry)를 활용해 target인 income(소득 5만달러 이상 여부)을 예측하려 합니다. table에서 Income은 소득이 5만달러 이상인 표본은 >50K, 5만달러 미만은 <50K로 binary하게 나뉩니다. 개인소득 기준값인 5만 달러는 1994년 당시 미국 가계중위소득 3만 3천달러보다 50%이상 높은 값이므로 경제적 상류층을 판별하기 적합한 기준이 될 수 있다고 판단하였습니다.

1. **explore data**
2. **Type of Data**

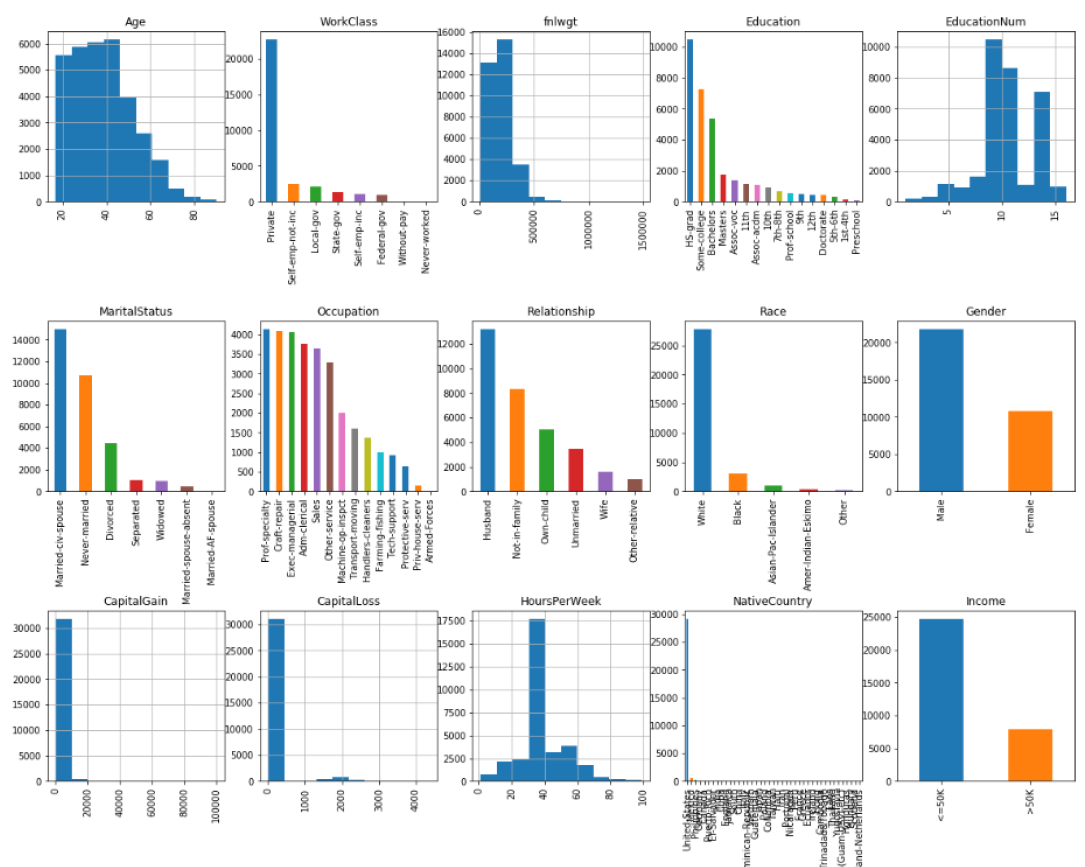
본 데이터에는 크게 3가지 특징이 있습니다. 우선 numerical data와 categorial data가 섞여 있습니다. 14개의 feature들 중 Age, fnlwgt, EducationNum, CapitalGain, CapitalLoss, HoursPerWeek는 quantity data이며, 나머지인 WorkClass, Education, MaritalStatus, Occupation, Relationship, Race, Gender, NativeCountry, 그리고 Income은 값이 class로 나뉘는 categorial data입니다. 이러한 특징 때문에 저희가 예측 모델을 만들기 위해서는 데이터 종류를 통일시켜야 했습니다. 저희는 one-hot encoding을 활용하여 categorial data를 numerical data로 가공하여 사용하였습니다.

1. **Limit of Census**

두 번째 특징은 대형 census의 특징상 다수의 데이터가 수집과정에서 문제를 갖고 있습니다. 우선 work class, occupation, native country는 missing data를 갖고 있습니다. 32561개의 샘플 중 각각 1836, 1943, 583개의 자료가 빠져 있습니다. 비율 상 약 5% 내외의 적은 값이지만 값이 들어 있지 않아 활용을 위해서는 처리가 필요합니다. 또한 native country와 education은 artifact를 갖고 있습니다. Native country의 경의 미국을 제외한 대다수의 데이터가 신뢰성이 떨어집니다. 수집 당시 국가 해당 국가의 환율과 물가 수준, 1인당 GDP가 반영되지 않았습니다. 본 연구의 목적이 상류층을 판별하는 것이고 5만 달러라는 기준이 미국을 기준으로 만든 기준이기 때문에 동일하게 적용하기 어렵습니다. Education의 경우 unification과 comparability artifact 문제가 있습니다. 응답자가 학력을 적게 만들어서 유사한 학력들이 다른 학력으로 집계되었습니다. 예를 들어 community collage, junior collage, state collage가 다르게 집계되었고 경우에 따라서는 spelling이나 띄어쓰기가 달라서 다른 학력으로 집계되기도 하였습니다. 또한 자퇴, 졸업, 재학 등의 구분들이 제대로 되어 있지 않아 각 항목들의 직접 비교가 어렵습니다.

1. **Distribution of Data**

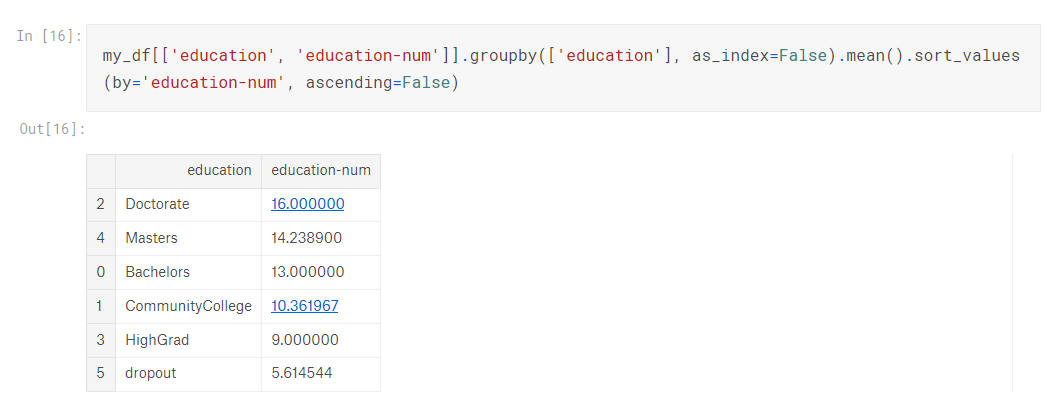
세 번째 특징은 feature들의 분포 중 다수가 모델에 사용하기 어려운 형태를 띄고 있습니다. 특히 numerical data가 큰 문제점을 갖고 있습니다. Numerical data의 가장 큰 문제는 Normalize가 안되어 있다는 점입니다. Tree 계열 machine learning 처럼 feature와 target data의 유사도를 purity를 보는 경우에는 큰 문제가 되지 않으나 regression 계열처럼 각 데이터의 분포가 independent and identical distributed를 가정한 모델에서는 validation과 loss function의 전제를 위반할 위험이 있습니다. 또한 몇몇 numerical 과 categorial data들은 분포가 지나치게 편중되어 있어서 다른 값들이 loss function에서 의미를 갖기 어려울 수 있습니다. 다음은 target을 포함한 15개의 feature의 분포입니다.



이중 age, fnlwgt, education Num, HoursPerWeek 는 어느정도 normalize가 가능하기는 하나 정규분포를 따르지 않고 있습니다. Capital loss와 Capital gain은 어느정도 정규 분포를 취할 것이라 예측되지만 실제로는 99%의 값이 0에 분포하고 있습니다. 조사 결과 두 feature가 사람들이 이해하기 어려운 개념이며, 또 답변하기 민감한 항목이기 때문에 실제 예상 분포와 많이 다르게 나왔습니다. 또한 native country의 경우도 97%가 미국인이며 work class와 race도 실제 다른 통계자료 분포보다 더 자영업과 백인에 편중되어 있습니다.

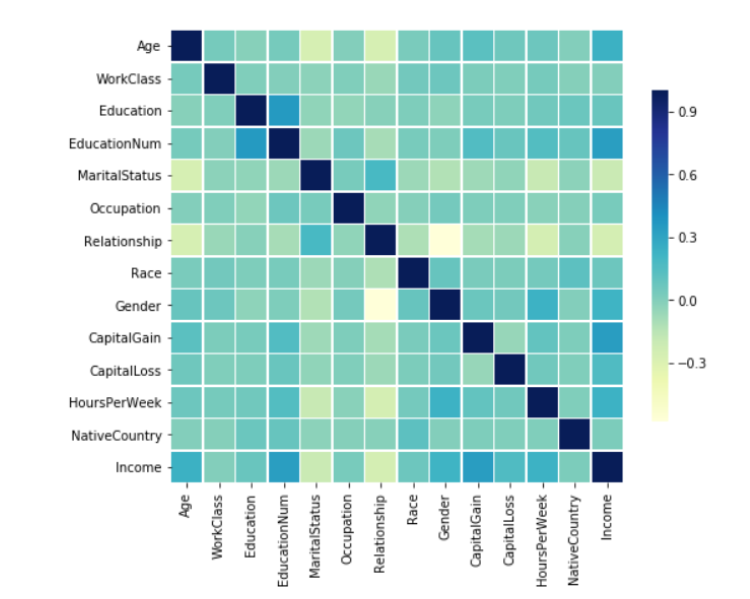
1. **Correlation**

끝으로 kaggle의 선행 연구 중 education과 education number가 중복 데이터이기 때문에 통합해야 한다는 주장이 있었습니다. (<https://www.kaggle.com/jieyima/income-classification-model>) 해당 자료에서 다음과 같이 두 feature의 관계를 분석하였습니다.



각 학력별로 평균 값을 잡아보면 dropout을 제외하고는 해당 학력에 필요로 하는 education number와 거의 같게 나오는 것을 알 수 있습니다.

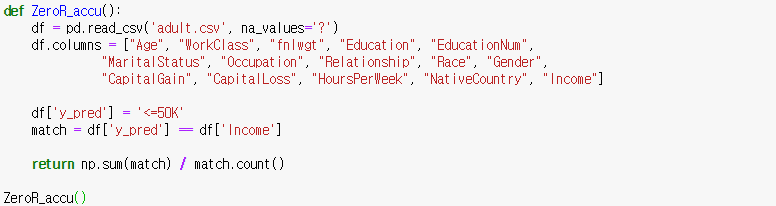
실제로 저희가 correlation을 돌려본 결과 두 변수만 유난히 상관관계가 높았습니다.



1. **Baseline**

저희는 본격적으로 모델을 돌리기 앞서 우선 baseline을 잡아서 적절한 목표 성능을 고민해보았습니다. Baseline은 우선 자체적으로 특정 값으로 결과를 찍는 ZeroR과 OneR을 이용하였습니다. 그리고 Kaggle에 있는 연구 자료 중 가장 치밀하게 설계된 모델 2개를 선택했습니다.

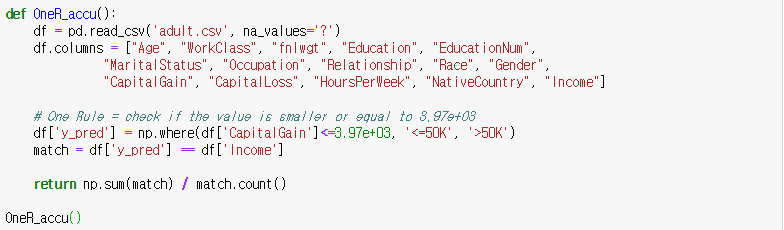
1. **ZeroR**



Result: 0.7607182343065395

직접 구현한 ZeroR 알고리즘입니다. 76%라는 정확도가 나오는 것을 확인할 수 있습니다. 이는 현재 가지고 있는 데이터가 ‘50K 이하’에 편중되어 있어 무조건 ‘50K 이하’로 예측하더라도 대부분 맞기 때문입니다.

**B. OneR**



Result: 0.7989640063879448

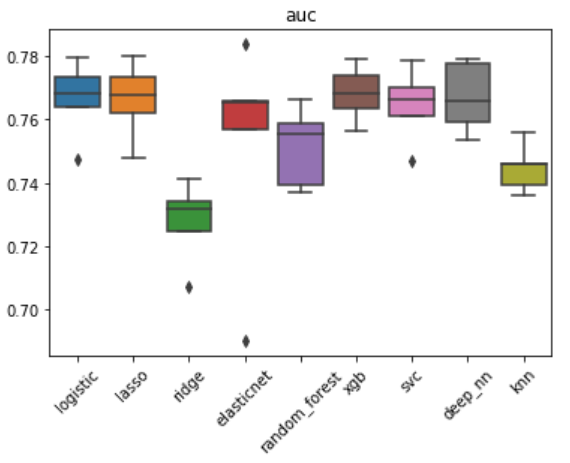
직접 구현한 OneR 알고리즘입니다. 79.9%라는 정확도가 나오는 것을 확인할 수 있습니다. 부여한 조건은 ‘CapitalGain’ 열 값이 3.97e+03 이하라면 ’<=50K’, 초과한다면 ‘>50K’으로 설정하는 것입니다. 비록 간단한 룰이지만, ZeroR보다 높은 accuracy 수치를 보였습니다

**C. Ensemble Modeling with Sklearn Pipelines (BusathBen, 2019)**

해당 저자는 본 보고서와 같은 데이터셋을 사용하여 Ensemble Modeling을 사용하였습니다.

(BusathBen. (2019년 5월 13일). “Income Classification Ensemble Modeling with Sklearn Pipelines”. Kaggle: <https://www.kaggle.com/bbusath5/ensemble-modeling-with-sklearn-pipelines>)

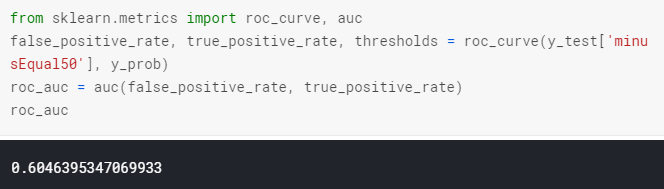
아래 plot은 BisathBen이 사용한 Ensemble 모델들의 Accuracy 결과 값을 Boxplot으로 시각화한 것입니다.



중간값을 기준으로 각 모델들의 area under ROC curve(정확도)를 추정하면 다음과 같습니다. Logistic, lasso, xgb가 대략적으로 비슷한 성능을 보여주고 있으며, 그 값을 대략 0.77입니다. 가장 낮은 값은 중간값을 기준으로 ridge 모델의 대략 0.73입니다.

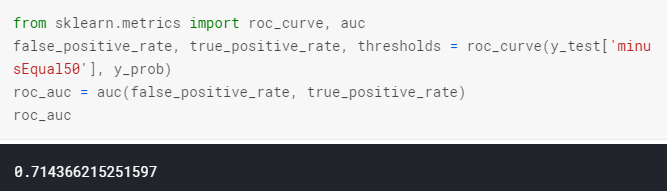
**D. Income Classification + EDA (AZURE DATASET) (BusathBen, 2019)**

해당 저자도 본 보고서와 같은 데이터셋을 사용했습니다 (TomasiDeLorenzo. (2019년 3월 13일). “Income Classification + EDA (AZURE DATASET)”. Kaggle: <https://www.kaggle.com/lodetomasi1995/income-classification-eda-azure-dataset>) . 이 모델의 특징은 로지스틱 회귀와 MLP 모델에 학습을 시켰다는 점입니다. 각 모델의 Area under ROC curve 값들을 알아보면 다음과 같습니다.



(로지스틱 회귀 모델에 대한 AUC 값)

로지스틱 회귀 모델에 대한 AUC 값은 대략 0.605가 나왔으며 60.5%의 정확도를 보인다고 말할 수 있습니다. 모든 예측을 50K 이하로 하는 ZeroR 알고리즘보다 부족한 성능을 볼 수 있습니다.



(MLP 모델에 대한 AUC 값)

MLP 모델에 대한 AUC 값은 대략 0.714가 나왔으며 71.4%의 정확도를 보인다고 말할 수 있습니다. 모든 예측을 50K 이하로 하는 ZeroR 알고리즘보다 약간 미진한 정확도를 보입니다. 데이터 전처리 및 파라미터 튜닝에서의 보완이 필요한 부분입니다.

**E. conclusion**

각 수치들을 비교하였을 때, ZeroR, OneR 알고리즘을 통하여 예측한 정확도를 넘는 것이 쉬운 일이 아니라는 것을 볼 수 있었습니다. 아무런 조건 없이 ‘<=50K’로 예측한 경우가 대략 76%의 정확도, 하나의 조건으로 예측한 경우가 대략 80%의 정확도를 나타난 점에서 최소한 80%를 넘는 모델을 만드는 것이 목표가 되었습니다. 다른 사람들의 보고서를 참고한 결과, 80%의 정확도를 상회하는 것은 까다로울 것이라고 생각했습니다. 첫 번째 참고한 보고서의 가장 높은 값은 ZeroR로 구한 값보다는 높았지만, OneR의 값보다는 낮습니다. 두 번째 보고서에서는 첫 번째 보고서보다 정확도가 떨어지는 것을 확인할 수 있었습니다.

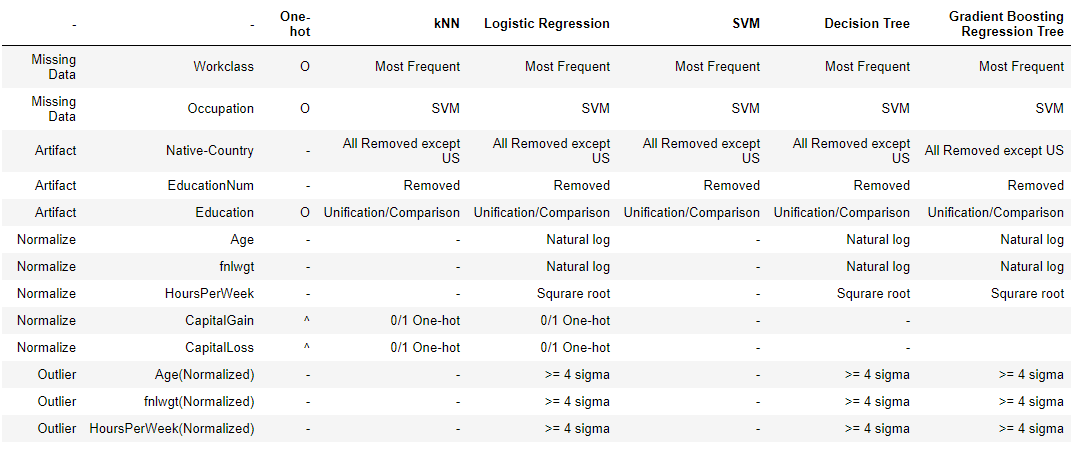
1. **cleaning data & Models**

앞서 data 특징과 baseline을 보고 저희는 유의미한 수준의 AUC 성능을 내는 모델을 만드는 것이 쉽지 않다는 사실을 발견하였습니다. 특정 모델이 baseline보다 높게 나오기 힘들 것이라고 예측하여 총 5가지 서로 다른 형태의 모델을 만들었습니다. 해당 5가지 모델은 KNN, Logistic regression, SVM, decision tree, Gradient boosting decision tree입니다. 이렇게 모델을 나눈 이유는 14개의 feature들이 다양하기 때문에 어떻게 모델을 구성하는가에 대해 팀원들의 의견이 갈렸기 때문입니다.

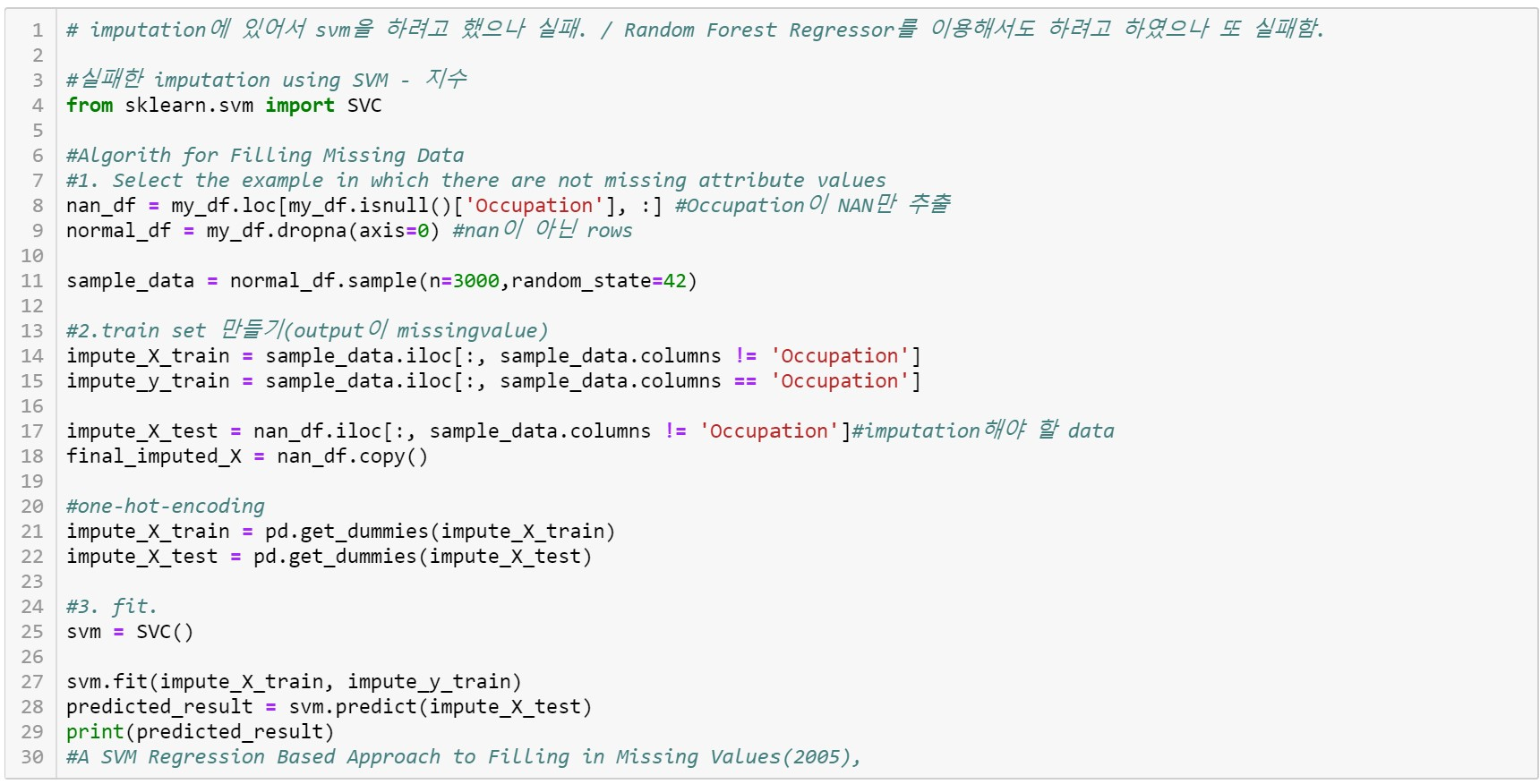
첫 번째로 재기된 주장은 Kaggle에 올라온 모델들이 정교하게 복잡하게 것에 비해 찍기에 가까운 ZeroR과 크게 차이가 없는 이유는 모델이 너무 복잡했기 때문이라는 것이었습니다. 그래서 가급적 간단한 모델인 Decision tree와 다르면서도 좀 더 복잡한 tree 모델인 Gradient Boosting Decision Tree를 사용하였습니다.

다음으로 제기되 주장은 데이터 처리과정에서 artifact와 표준화 처리가 미흡했다는 주장이었습니다. 그래서 같은 방식으로 logistic regression을 하되 전처리를 다르게 해서 결과를 보기 하였습니다.

끝으로 나온 주장은 tree와 regression 계열이 모두 실패했기 때문에 다른 종류의 수업시간에 배운 모델 중 다른 종류의 loss function을 사용하는 모델을 사용해보자는 것이었습니다. 그래서 배운 모델 대표적인 모델인 SVM과 KNN을 선택하였습니다. 이 두가지를 선택한 이유는 변수를 얼마나 많이 반영하는 것이 더 유리한가에 대해 의견이 갈렸기 때문입니다. OneR이 의외로 성공률이 높았기 때문에 소수의 강력한 sample들이 더 잘 반영되어야 한다는 의견에 따라 SVM을 사용하였고 이에 대한 반대 의견으로 가까운 변수들의 관계가 중요하다는 의견에 따라 KNN을 선택하였습니다. 다만 성격이 서로 다른 모델들을 너무 많이 골라서 데이터 처리를 해당 모델에 따라 다 다르게 해야만 했습니다.



다음은 각 모델에 따른 저희의 data cleaning 절차입니다. Missing data의 data의 distribution에 의존하기 때문에 모두 같은 방식으로 처리하였습니다. Workclass의 경우 분포가 ‘private’에 70정도 편중되어 있기 때문에 missing data를 모두 most frequen인 ‘private’으로 하였습니다. occupation는 분포가 균등하여 KNN이나 SVM으로 imputing 하려고 하였으나 전 처리 문제에 실패하여 어쩔 수 없이 most frequent를 사용하였습니다.



Artifact도 data에 의존하기 때문에 모두 일괄적으로 처리하였습니다. Nativecountry는 지나치게 미국에 편중되어 있고 다른 나라 데이터가 의미가 없기 때문에 미국을 제외한 다른 sample들을 모두 제거하였습니다. Education은 앞서 Kaggle 선행자료에서 진행했듯이 항목들 중에서 유사한 학력들은 합치고 비교 가능하게 일괄적으로 ‘진학’으로 통일하였습니다. 예를 들어 university 진학, 자퇴, 졸업은 모두 university로 통일하였습니다.

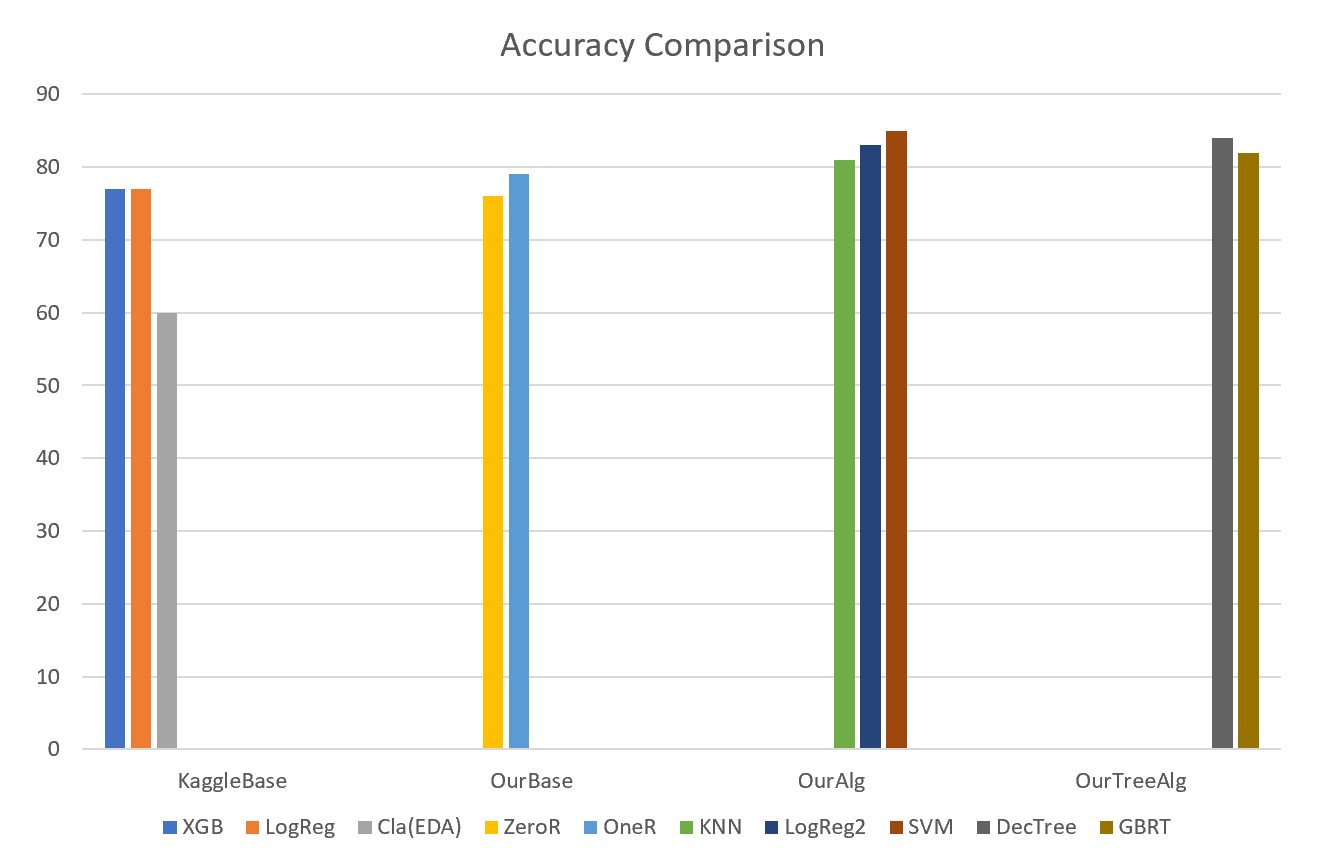
다음으로 Age, fnlwgt, Hours per week는 normalize가 가능한 data set이기 때문에 normalize 하였습니다. Age와 fnlwgt는 결과 값을 log로 transform해서 사용하였고 Hours per week는 결과 값을 제곱근으로 transform해서 사용하였습니다. 그러나 해당 자료들의 요구 조건은 data가 아닌 모델에 의존하기 때문에 모델에 따라 normalize한 자료를 쓰거나 원본자료를 사용하였습니다. Decision tree 계열은 purity로 구분하기 때문에 작은 차이라도 원본 그대로 남겨두는 것이 혹시 모를 희미하지만 중요한 페턴을 찾는데 도움이 될 수도 있다고 판단하여 원본 자료를 사용하였습니다. 한편 distance를 이용하여 loss function을 최소화하는 모델들은 사용된 sample들이 i.i.d 상태에서 선택되었다는 가정하에 평균 loss를 구하는 것이 generality를 유지하는데 좋기 때문에 normalize한 데이터를 사용하였습니다. 결론적으로 tree 계열인 decision tree와 gradient boosting regression tree는 원본 데이터를 사용하였고 이외의 loss function을 쓰는 KNN, logistic regression는 normalized된 데이터를 사용하였습니다.

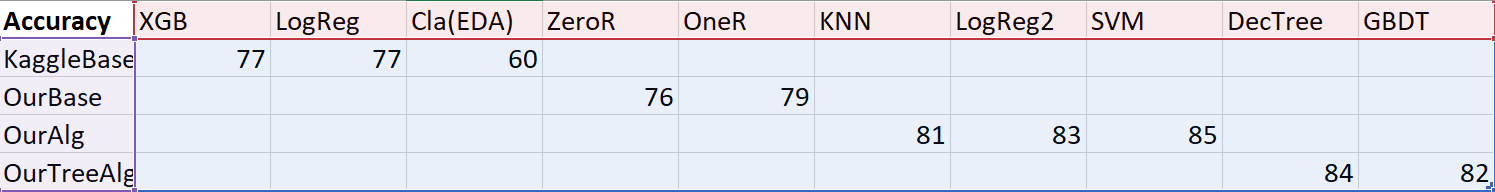
Outlier의 경우 normalize의 연장선 상에서 normalized data를 사용한 모델에 한에서 제거 하였습니다. 여기에는 2가지 이유가 있습니다. 우선 Outlier를 구하기 위해서는 기준이 필요하고 이 기준을 표준편차로 사용해야 하기 때문에 normalize 된 자료들만 사용하였습니다. 다음으로 outlier를 빼는 이유가 norm을 사용하는 loss function에서 나올 확률이 희박하지만 절대값이 큰 outlier가 나오면 실제 population에서 미치는 영향보다 해당 outlier가 더 영향을 미쳐 전체 loss값이 왜곡되기 때문입니다. 즉 앞서 normalize처럼 tree 계열인 decision tree와 gradient boosting regression tree는 원본 데이터를 사용하였고 이외의 loss function을 쓰는 KNN, logistic regression는 outlier를 제거한 자료를 사용하였습니다. Outlier는 공통적으로 two tail 4 시그마 이상을 제외하였습니다. 4시그마가 대략 15,000이기 때문에 전체 크기가 3만인 자료에서 2개 이상 나오기 힘들기 때문입니다.

이외에 education과 correlation이 높은 educationNum은 정보로서 의미가 크지 않으며 Multicollinearity 문제를 야기할 수 있어 제외하였습니다. 또한 capital loss와 capital gain은 자료 수집 과정상에 문제가 제기되었고 99%가 ‘0’값을 갖기 때문에 loss function을 사용하는 모델에서 numerical data임에도 불구하고 ‘0’값인 것과 ‘0’이 아닌 것(1)로 바꿔서 categorial data처럼 사용하였습니다. 그러나 SVM의 경우 대다수의 소수의 멀리 편차가 큰 데이터가 영향을 미치기 힘드므로 원본 데이터를 사용하였습니다.

Data cleaning과 model running, visualize와 이것들의 코드는 원래 같은 쥬피터 노트에서 작성하였으나 data set을 모델마다 다르게 설정하는 과정에서 충돌이 일어났습니다. 그래서 어쩔 수 없이 모두 쥬피터 노트를 나누고 전체에 흐름에 대한 내용을 word파일로 나눴습니다. ds\_term\_project\_fin.zip에 모델들이 있습니다. 1번부터 1. KNN.ipynb, 2. Logistic Regression.ipynb, 3. SVM.ipynb, 4. Decision Tree.ipynb, 5. Gradient Boosting Regression Tree.ipynb 이상 5개 파일이 각각에 해당하는 모델들입니다.

1. **Visualization & Communication**
2. **Conclusion**





다음은 저희가 선택한 baseline들과 직접 실행시킨 모델들의 결과입니다. 성능 비교 기준은 일괄적으로 Accuracy를 사용하였습니다. 왼쪽부터 저희가 baseline으로 참고한 Kaggle의 상위권 모델, 일종의 찍기 baseline(ZeroR, OneR), Loss function을 사용하는 자체 모델들, Tree 계열 자체 모델들 순으로 배열하였습니다. 전체적으로 봤을 때 기본 baseline인 ZeroR, OneR자체가 높기 때문에 성능 차이가 크지 않는 거처럼 느껴집니다. 그러나 자세히 관찰하면 확연하게 Kaggle model<기본 baseline<저희 모델들 순의 확실한 차이를 확인할 수 있습니다. 차이가 크지는 않지만 Kaggle 상위권 모델들이 80을 넘지 못한 반면 저희 모델들은 모두 80을 넘었다는 점에의 의의가 있습니다.

또한 위 결과에서 흥미로운 사실들을 몇 가지 발견할 수 있습니다. 우선 Decision Tree가 더 복잡한 같은 계열 모델인 GBRT보다 더 성능이 좋다는 것이고 Kaggle에서도 가장 간단하게 설계된 모델인 Logistic Regression 더 성능이 높다는 것입니다. 즉 변수가 다양하게 많을 때 모델이 복잡하다는 사실이 반드시 더 좋은 성능을 보장해주지 않음을 보여줍니다. 또한 Logistic regression이 Kaggle과 저희 모두 사용되었는데 저희 모델이 더 유의미하게 성능이 좋았습니다. 두 모델의 차이는 전 처리 과정인데, 같은 모델이라고 해도 데이터의 처리에 따라 성능 차이가 날 수 있음을 보여주고 있습니다. 끝으로 Loss function을 사용하는 모델과 Purity를 사용하는 Tree 계열 모델들 사이에 성능상 차이를 예상하였으나 평균이 같아 차이가 거의 없었습니다. 즉 model 내부에서 모델을 학습하는 목표가 모델의 성능차이와 큰 연관이 없다는 사실을 알 수 있었습니다.

1. **Limit**

저희 팀이 가장 심혈을 기울인 부분은 data cleaning입니다. Census 자료의 경우 그 자체로 많은 문제점을 내포하고 있기 때문에 단순히 주어진 데이터로 좋은 성능을 얻는 것이 큰 의미가 없을 수도 있다고 판단했기 때문입니다. 그런 의미에서 missing data를 다룰 때 occupation을 적절하게 imputing 하지 못한 것이 가장 아쉽습니다. Work class의 경우 가장 약 15종류의 유의미한 category를 갖고 있고 가장 큰 값이 13%밖에 차지하지 않을 정도로 균등한 분포를 갖고 있습니다. 또한 missing data의 비중이 5%정도이기 때문에 KNN이나 SVM과 같은 모델을 돌려 유사도나 연관성을 기준으로 imputing 하였으나 data 처리 문제 때문에 실패하였습니다. 비록 occupation이 유의미한 페턴을 보여줄 것이라는 보장이 없기는 하지만 어쩔 수 없이 적합하지 않은 imputing을 사용하였습니다.

또한 실제 저희가 census논문을 읽기는 하였지만 사회학 전공자가 있었는데도 불구하고 완전히 이해를 하지 못했습니다. 분명 저희가 찾지 못한 artifact들이 더 많을 수밖에 없었지만 domain의 한계로 더 완벽한 전처리를 하지 못한 점이 아쉽습니다. 생물학이나 정보학과 달리 사회학에서 수집되는 자료들이 지나치게 많은 변수들이 존재하는 상황에서 수집되기 때문에 모델의 성능 이전에 제대로 된 데이터 셋이 구성되었는지 확신을 갖기 어려웠습니다. 가령 성비의 경우 남녀의 성비는 거의 절반에 가까우나 수집된 데이터의 성비는 2:1 이었습니다. Pattern을 찾는다면 큰 문제가 없을 수도 있지만 이렇게 모집단을 잘 반영하지 못하는 경우 저희가 예상하지 못한 문제가 있을 수 있을 것이라고 판단됩니다.

1. **Development**

모든 상황을 종합해보면 저희는 크게 3가지 점에서 데이터 과학 영역에서 발전할 부분이 있다고 생각합니다. 우선 전처리 기법들을 더 배워야할 것 같습니다. 앞에서 성비가 모집단과 맞지 않는 경우 적합한 기법을 알고 있었다면 더 좋은 전처리가 가능했을 것 같습니다. 또 normalize경우도 log와 스퀘어 루트를 사용하였는데 더 좋은 transform이 있다면 더 정규분포에 가까운 모양으로 만들 수도 있었을 것 같습니다. 앞서 모델을 돌리는 것만큼이나 전처리와 데이터 정리가 중요했기 때문에 가시적으로 드러나지 않는 부분이라고 해도 더 공부가 필요할 것 같습니다.

다음으로 더 많은 모델들을 학습해야 할 것 같습니다. Kaggle model들을 보고 저희가 같은 몇 가지 의문들을 해결하기 위해 KNN, Logistic regression, SVM, Decision tree, GBRT를 사용하였는데 만약 저희가 더 많은 모델들을 알고 있었다면 더 적합한 비교 모델들을 선택할 수 있었을 것 같습니다. 특히 SVM같은 경우 적용 가능한 응용 모델이 많았지만 저희가 아직 공부가 부족하여 사용하기 어려운 모델들이 많았습니다.

끝으로 모델에 적합한 visualization을 공부해야 할 것 같습니다. 저희가 project를 진행하면서 visualization이 주로 data cleaning과 결과값 비교에 사용되었는데 개별 모델들 결과를 좀 더 효과적으로 visualize할 수 있으면 좋을 것 같습니다. 특히 변수가 3개 이상 넘어가는 경우와 numerical 과 category가 섞이는 경우, Tree 계열 모델인 경우 결과를 쉽게 전달하도록 visualize하는데 큰 어려움을 겪었습니다. 이와 같은 상황에서 전달력을 높일 수 있는 visualize tool들을 공부할 것입니다.