프로젝트 수행사항

[지원분야 : 빅데이터 분석가 / 신입**]**

프로젝트

1. Ferrari's Nightmare 2010 ~ 2019	2020. 10. ~ 2020. 11.
2. Neural Network Image Classification	2021. 02 ~ 2021. 03
3. Cryptocurrency Market Manipulation	2021. 03 ~ 2021. 04
4. Fairness in Machine Learning	2021. 04 ~ 2021. 05
5. Bitcoin Price Prediction (진행 중)	2021. 07 ~ 2021. 08

프로젝트 기술서 (1)

2020.10~2020.11

▶ 프로젝트명 : Ferrari's Nightmare 2010 ~ 2019

▶ 개발환경 : Jupyter Notebook

▶ 인 원 : 3

▶ 사용언어 : Python, SQL

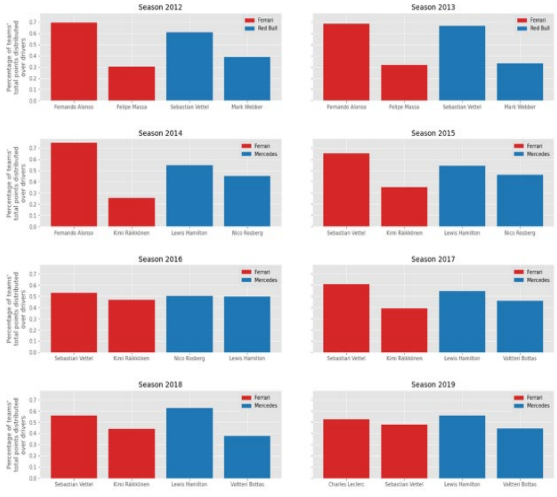
▶ 프로젝트 깃헙 : https://github.com/jasonkwak190/works/tree/main/python_project_1

▶ 프로젝트 소개 : F1 레이싱은 1950 년부터 시작되어 대략 60년 간의 긴 역사를 가지고 있습니다. 수많은 팀 중 페라리 팀의 압도적인 성적으로 1등을 가장 많이 수상하였지만, 최근 2010 ~ 2019 년도 성적을 보면 1등을 한번도 하지 못하였습니다. 그 이유를 알아보기 위해서 프로젝트를 시행하였습니다.

▶ 본인 역할 :

1. 팀원이 크롤링 한 csv 파일을 SQL Database 에 연동하는 역할을 맡았습니다
   1) SQLite Library 를 이용하여 파이썬 환경에서 SQL query를 원활하게 이용할 수 있도록 연결 하였습니다.
   2) 6개의 csv 파일 모두 연동 완료 하였습니다.
2. SQL Query를 이용하여 지난 2010 ~ 2019의 운전자 성적을 분석하였습니다.
   1) select, where, group by, having 등 SQL query를 이용하여 운전자 이름, 성적 팀 이름을 가져와서 비교 분석 하였습니다.
   2) 분석 결과 2016년 까지 페라리 팀의 운전자 성적은 저조 하였지만, 그 이후는 경쟁자 팀과 비슷한 성적을 낸 것으로 분석 되었습니다.
   3) 2016년 이후에는 페라리 팀이 비슷한 성적을 냈지만, 1등을 하지 못한 이유를 알기 위해 다른 데이터를 분석 하였습니다.

```
In [2]: def create_connection(db_file, delete_db=False):
            import os
            if delete_db and os.path.exists(db_file):
                os.remove(db_file)

            conn = None
            try:
                conn = sqlite3.connect(db_file)
                conn.execute("PRAGMA foreign_keys = 1")
            except Error as e:
                print(e)

            return conn


        def create_table(conn, create_table_sql, drop_table_name=None):

            if drop_table_name: # You can optionally pass drop_table_name to drop the table.
                try:
                    c = conn.cursor()
                    c.execute("""DROP TABLE IF EXISTS %s""" % (drop_table_name))
                except Error as e:
                    print(e)

            try:
                c = conn.cursor()
                c.execute(create_table_sql)
            except Error as e:
                print(e)

        def insert_sql_statement(insert_data, conn, table_name):
            with conn:
                cur = conn.cursor()

                empty_value_container = "?,"*len(insert_data[0])
                empty_value_container = "".join(["(",empty_value_container.strip(","),")"])

                cur.executemany(f"INSERT INTO {table_name} VALUES {empty_value_container}", insert_dat
        a)

        def read_sql_query(query, conn):
            result = pd.read_sql_query(query, conn)
            return result
```



프로젝트 기술서 (2)

2021. 02 ~ 2021. 03

▶ 프로젝트명 : Neural Network Image Classification

▶ 개발환경 : Pycharm

▶ 인 원 : 3

▶ 사용언어 : Python

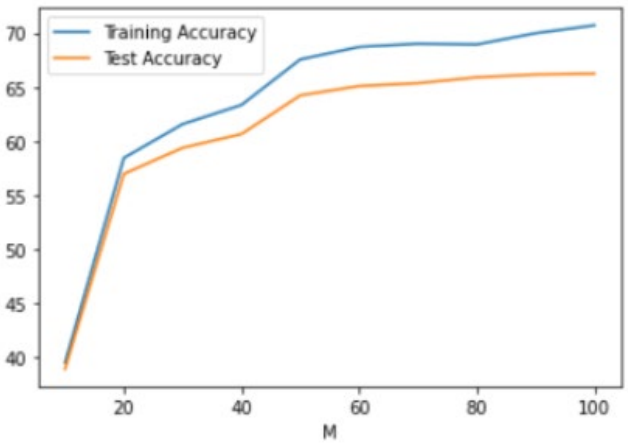
▶ 프로젝트 깃헙 : https://github.com/jasonkwak190/works/tree/main/Deep_learning

▶ 프로젝트 소개 :

Neural Network 를 이용하여 이미지를 분류하는 프로젝트 입니다. Keras, Tensorflow를 이용하지 않고 그 안에 있는 공식을 살펴보기 위해 직접 수학적인 공식을 대입 하였습니다. 최종 목적은 Hidden Layer (M), 람다, Layer (L) 의 최적의 개수를 도출하고 Activation Function 중 Sigmoid, Relu, Tanh 를 대입하여 각각의 Accuracy rate를 구하는 것입니다.

▶ 본인 역할 :

1. Neural Network에 대한 전체적인 이해와 공부를 하였습니다.
2. 데이터셋을 Train, Test data 로 나누어서 관련 공식들을 대입하였습니다.
   1) Train data : 67.79% / Test data : 64.52% 의 Accuracy Rate 를 도출하였습니다.
   2) M 이 100 에 가까워 질수록 Accuracy 가 더욱더 올라가는 것을 확인 하였습니다.
   3) Sigmoid, Relu, Tanh 대입하여 Train, Test data 의 Accuracy를 확인하고 그중 Relu가 제일 높은 결과가 나왔음을 확인 했습니다.



- sigmoid:
  Training completed in 91.59 seconds.
  Training set Accuracy: 80.20%
  Testing set Accuracy: 76.34%
- relu:
  Training completed in 92.46 seconds.
  Training set Accuracy: 88.27%
  Testing set Accuracy: 73.63%
- tanh:
  Training completed in 94.70 seconds.
  Training set Accuracy: 79.73%
  Testing set Accuracy: 75.61%

프로젝트 기술서 (3)

2021. 03 ~ 2021. 04

▶ 프로젝트명 : Cryptocurrency Market Manipulation

▶ 개발환경 : Pycharm, Jupyter Notebook

▶ 인 원 : 3

▶ 사용언어 : Python

▶ 프로젝트 깃헙 : https://github.com/jasonkwak190/works/tree/main/python_project_2

▶ 프로젝트 소개 :

가상화폐는 시가조작이 법으로 규제되어 있지 않아 주식에서는 불법적인 것이 흔하게 일어나고 있습니다. 주로 축적 과정을 사전에 하고 가격을 급격하게 펌핑하고 그 후 전량 매도하는 덤핑 방법을 사용 합니다. 이러한 과정을 Apriori & Anomaly Detection으로 미리 탐지 할 수 있는지 알아보기 위해 프로젝트를 수행하였습니다.

▶ 본인 역할 :

1. Kaggle 에서 가져온 비트코인 데이터에 Apriori 방법을 도입하여 변형 하였습니다.
   1) 펌핑과 덤핑 할 때 하려면 봇으로 빠른 속도로 여러 번 거래 하기 때문에 총 거래량의 평균보다 많은 수를 거래 한 계정을 골라 냈습니다. Apriori 방법은 평균을 초과한 계정들만 1로 표시하고 나머지를 0으로 하는 것 입니다.
   2) 그 결과 특정 계정들이 펌핑과 덤핑을 주도 하고 있다는 것을 알 수 있었습니다.
   3) 단점은 거래소가 특정 계정들의 아이디에 대한 정보를 제공 하지 않으면 알 수가 없습니다. 그리고 지속적으로 모니터링을 해야 하기 때문에 데이터셋이 너무 크면 시간과 비용적으로 손해 입니다.

Apriori Algorithm

	Source	Target	Trade_Id	Bitcoins	Money	Money_Rate	Date
0	895	3931	35372	23.020	18.061	0.784579	2011-04-01 00:28:54
1	895	722	35373	10.000	7.800	0.780000	2011-04-01 00:28:54
2	895	3605	35374	35.000	27.300	0.780000	2011-04-01 00:28:54
3	895	3966	35375	10.600	8.246	0.777925	2011-04-01 00:28:54

	Source	Date_2011-04-01 00:28:54	Date_2011-04-01 06:42:47	Date_2011-04-01 07:21:41
10	2387	0	1	0
11	895	0	0	1
12	895	0	0	1
13	895	0	0	0

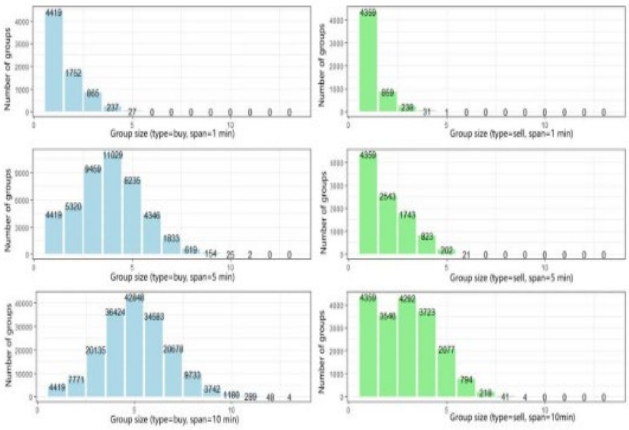


Fig. 2: The Results of Improved Apriori Algorithm.

프로젝트 기술서 (4)

2021. 04 ~ 2021. 05

▶ 프로젝트명 : Fairness in Machine Learning

▶ 개발환경 : Pycharm

▶ 인 원 : 3

▶ 사용언어 : Python

▶ 프로젝트 깃헙 : https://github.com/jasonkwak190/works/tree/main/Deep_learning

▶ 프로젝트 소개 :

COMPAS는 지난 범죄 기록을 바탕으로 피고가 전과가 있을 확률을 구하는 모델 입니다. 하지만 이 모델의 문제점은 인종이 흑인이면, 백인보다 두배 더 높은 수치를 나타냅니다. 이는 심각한 윤리적인 문제가 있기에 이러한 불공평한 차별을 없애기 위해 프로젝트를 수행 하였습니다. Naïve Bayes, Neural Network, SVM 을 이용하여 Accuracy 와 비용을 계산 하고 비교분석 하였습니다.

▶ 본인 역할 :

1. 3가지 방법 중 Neural Network 방법을 맡아 하였습니다.
   1) 각 인종당 FPR, FNR, TPR, TNR 을 계산 하였습니다.
   2) Accuracy와 비용을 비교 분석 하였지만 다른 method보다 accuracy가 낮았습니다.