

파이널 프로젝트

-마키아벨리즘 심리테스트를 활용한 투표 여부 분석

Team Wangbau



Contents

01. 프로젝트 설계

문제 상황분석
시스템 아키텍처

02. 데이터

데이터 탐색
데이터 수집
데이터 분석 및 전처리

03. 모델링 기법 선택

-기준 : 타입 및 분석목적

04. 설계 및 성능 평가

-성능평가, 모델 검증

05. 프로그램화

-서버 구축, 깃 활용

06. 결과 분석 & 소감

07. Q&A

문제상황 분석

현대 민주주의 사회에서는 선거가 아주 중요한 행사입니다. 각 나라에서 국가의 지도자를 뽑기 위해 선거를 실시하고 있습니다. 선거에 대한 관심도는 전 세계적으로 모든 나라에서 높아지고 있는 추세입니다. 이런 중요한 행사를 진행하기에 앞서 어느 지역에서 어느 정도의 투표율이 나와 선거구를 어느 정도 설치하고 선거에 투입되어야 하는 인력 배치를 어느 정도 해야 되는지에 대한 예측이 굉장히 중요하다고 생각합니다. 그리고 투표율을 예측하는 것은 어느 정당에서나 많은 관심을 가지고 있는 지표라고 생각합니다.



Problem

투표율의 예측을 위하여 최근 MBTI 등으로 많은 사람들의 관심을 받고 있는 심리테스트의 데이터들과 인적 사항들의 데이터들을 모았습니다. <https://openpsychometrics.org/>에서 사람들의 마키아벨리즘 정도를 파악할 수 있는 심리 테스트 질문들을 통해서 해당 사람이 투표를 할 것인지 아닌지에 대해 예측을 하였습니다.

MACHIAVELLISM TEST?



마키아벨리아니즘, 마키아벨리즘이라고도 부르는 마키아벨리 주의 (machievellianism or machiavellism)는 일반적으로 국가의 발전과 인민의 복리증진을 위해서는 어떠한 수단이나 방법도 허용된다는 국가 지상주의적인 정치 이념을 뜻한다.

최근 MBTI를 대표적으로 여러 심리테스트들을 매우 많은 사람들이 이용을 하고 있고 그에 따른 결과를 통해서 자기 자신을 소개하거나 또 다른 사람의 심리테스트 결과를 듣고 그 사람인지 어떤 사람인지에 대해서 파악을 하는 사람들이 늘고 있다. 이런 심리테스트의 데이터들이 많이 쌓이고 있는 시대에서 심리테스트 결과에 따른 사람들의 행동에 대해서 예측을 할 수 있게 되는 것은 굉장히 비즈니스적으로 뿐만 아니라 여러 분야에서 의미 있는 분석이 될 것이다.

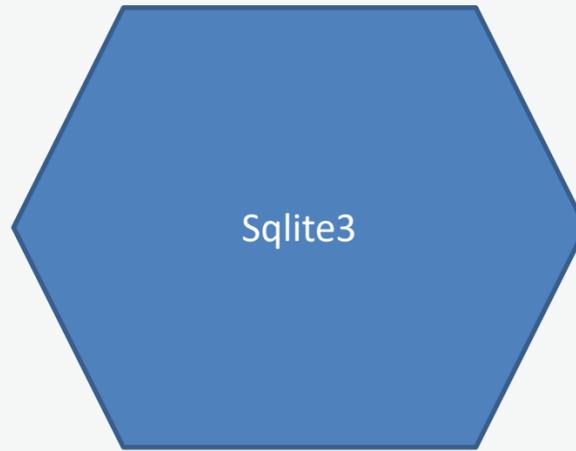
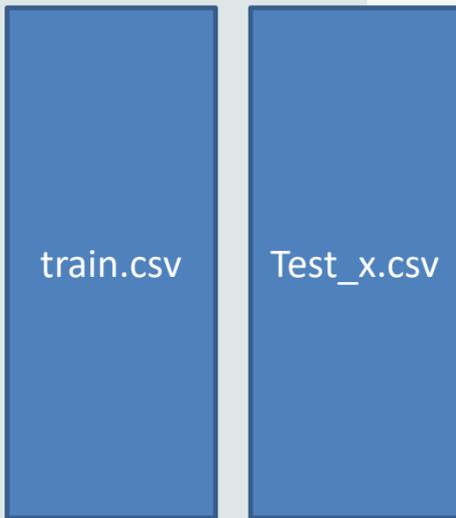
시스템 아키텍처

데이터 수집 & 전처리

머신러닝 모델링

데이터베이스 구축

웹 서버 구현



웹페이지에서의 설문조사



조사값을 데이터베이스에 저장과 동시에 csv로 만듦



csv를파일을 읽어 머신러닝모델이 작동한다



모델의 결과값으로 데이터베이스에 저장, 화면에 출력

데이터 탐색과 수집

심리 성향 예측 AI 경진대회

월간 데이콘 8 | 심리 테스트 분석 | AUC | 분류

상금 : 100만원+애플워치

2020.09.28 ~ 2020.11.16 17:59 [+ Google Calendar](#)

1,399팀 마감



[참여](#)

대회안내 **데이터** 코드 공유 토론 대회문의 리더보드 제출

설명

데이터 출처: openpsychometrics.org
train.csv : 훈련용 데이터
shape : (45532, 78)

test_x.csv : 정답용 데이터
shape : (11383, 77)
voted 없음

데이터 변수 설명 : <https://www.dacon.io/competitions/official/235647/talkboard/401534?page=1&dtype=recent&ptype=pub>

```
In [5]: 1 train.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45532 entries, 0 to 45531
Data columns (total 78 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   index      45532 non-null  int64
1   QaA        45532 non-null  float64
2   QaE        45532 non-null  int64
3   QbA        45532 non-null  float64
4   QbE        45532 non-null  int64
5   QcA        45532 non-null  float64
6   QcE        45532 non-null  int64
7   QdA        45532 non-null  float64
8   QdE        45532 non-null  int64
9   QeA        45532 non-null  float64
10  QeE        45532 non-null  int64
11  QfA        45532 non-null  float64
12  QfE        45532 non-null  int64
13  QgA        45532 non-null  float64
14  QgE        45532 non-null  int64

In [6]: 1 test.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11383 entries, 0 to 11382
Data columns (total 77 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   index      11383 non-null  int64
1   QaA        11383 non-null  float64
2   QaE        11383 non-null  int64
3   QbA        11383 non-null  float64
4   QbE        11383 non-null  int64
5   QcA        11383 non-null  float64
6   QcE        11383 non-null  int64
7   QdA        11383 non-null  float64
8   QdE        11383 non-null  int64
9   QeA        11383 non-null  float64
10  QeE        11383 non-null  int64
11  QfA        11383 non-null  float64
12  QfE        11383 non-null  int64
13  QgA        11383 non-null  float64
14  QgE        11383 non-null  int64
```

질문지

1. 세상에서 앞서가는 대부분의 사람들은 깨끗한 도덕적 삶을 영위합니다.
 2. 대부분의 범죄자들과 다른 사람들의 가장 큰 차이점은 범죄자들이 잡힐만큼 어리 석다는 것입니다.
 3. 다른 사람을 완전히 신뢰하는 사람은 문제를 일으킵니다.
 4. 일반적으로 사람들은 강요하지 않으면 열심히 일하지 않습니다.
 5. P.T. Barnum은 매분마다 잘 속는 사람이 태어 났다고 말했다 때 틀렸습니다. (*)
 6. 다른 사람에게 거짓말을 할 이유가 없습니다. (*)
 7. 불치병으로 고통받는 사람들은 고통없이 죽음을 당할 수있는 선택권을 가져야합니다.
 8. 대부분의 사람들은 재산을 잃는 것보다 부모의 죽음을 더 쉽게 잊습니다.
 9. 대부분의 사람들은 용감합니다.
 10. 모든 사람들이 악의적인 행동을 가지고 있으며 기회가 주어지면 나올 것이라고 가정하는 것이 가장 안전합니다.
 11. 대체로 중요하고 정직하지 않은 것보다 겸손하고 정직한 것이 낫습니다. (*)
 12. 정직은 모든 경우에 최선의 정책입니다.
 13. 여기 저기 모퉁이를 자르지 않고는 앞서 나가기가 어렵습니다.
 14. 누군가에게 당신을 위해 무언가를 해달라고 요청할 때, 더 많은 무게를 지닐 수있는 이유를 제시하는 것보다 그것을 원하는 진정한 이유를 제시하는 것이 가장 좋습니다.
 15. 사람들을 다루는 가장 좋은 방법은 그들이 듣고 싶은 것을 말하는 것입니다.
 16. 모든 면에서 선해지는 것이 가능합니다.
 17. 대부분의 사람들은 기본적으로 착하고 친절합니다. (*)
 18. 도덕적으로 옳다고 확신 할 때만 행동을 취해야합니다. (*)
 19. 중요한 사람들을 아첨하는 것이 현명합니다.
 20. 그렇게하는 것이 유용하지 않다면 당신이 무언가를 한 진짜 이유를 누구에게도 말하지 말아야 합니다.
- 1=매우 그렇지 않다, 2=약간 그렇지 않다, 3=보통이다, 4=약간 그렇다, 5=매우 그렇다.
다른 질문들은 5에 가까울수록 마키아벨리즘 성향이 강한 것이지만 (*) 된 질문들은 1에 가까울 수록 마키아벨리즘 성향이 강한 질문 내용입니다.

질문지

age_group : **연령**
education : **교육 수준**
1=**고졸 미만**, 2=**고졸**, 3=**학사 학위**, 4=**석사 학위**, 0=**무응답**
engnat : **모국어가 영어**
1=**그렇다**, 2=**그렇지 않다**, 0=**무응답**
familysize : **형제자매 수**
gender : **성별**
Male, Female
hand : **필기하는 손**
1=**오른손**, 2=**왼손**, 3=**양손**, 0=**무응답**
married : **혼인 상태**
1=**미혼**, 2=**최근 결혼**, 3=**과거에 결혼**, 0=**그 외**
race : **인종**
Asian, Arab, Black, Indigenous Australian, Native American, White, Other
religion : **종교**
Agnostic, Atheist, Buddhist, Christian_Catholic, Christian_Mormon, Christian_Protestant, Christian,_Other, Hindu, Jewish, Muslim, Sikh, Other
tp__ 항목은 "내 자신을 다음과 같이 본다"로 평가되었습니다.
tp01 : **외향적이고 열정적입니다.**
tp02 : **비판적이고 다툼이 있습니다.**
tp03 : **신뢰할 수 있고 자제력이 있습니다.**
tp04 : **불안하고 쉽게 화를 냅니다.**
tp05 : **새로운 경험에 개방적이며 복잡합니다.**
tp06 : **조용한 편입니다.**
tp07 : **동정적이고 따뜻합니다.**
tp08 : **무질서하고 부주의합니다.**
tp09 : **침착하고 정서적으로 안정되어 있습니다.**
tp10 : **비창조적입니다.**
순서형 변수 (0~7중 자신이 어느 정도인지 고름)
urban : **유년기의 거주 구역**
1=**시골쪽**, 2=**교외 지역**, 3=**도시**, 0=**무응답**
voted (타겟): **지난 해 국가 선거 투표 여부**
1=Yes, 2=No

데이터 분석 및 전처리

```
1 # train 데이터의 누락값 확인  
2 print(np.count_nonzero(train.isnull()))
```

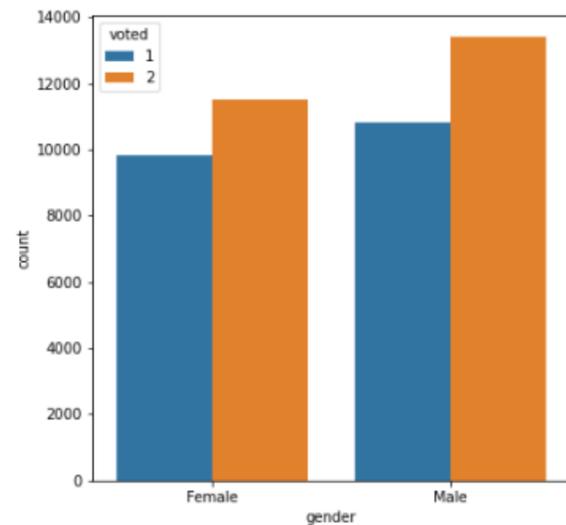
0

```
1 # test 데이터의 누락값 확인  
2 print(np.count_nonzero(test.isnull()))
```

0

```
1 plt.figure(figsize=(6,6))  
2 sns.countplot(data=eda_train,x='gender',hue=eda_train['voted'])
```

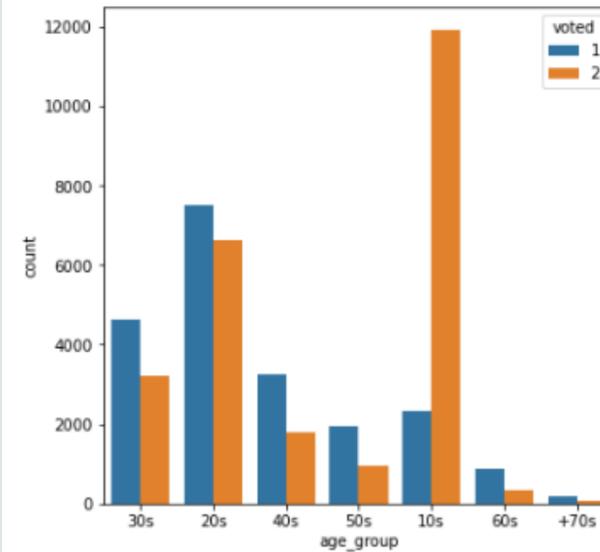
<AxesSubplot: xlabel='gender', ylabel='count'>



• 성별에 따른 투표여부

```
1 plt.figure(figsize=(6,6))  
2 sns.countplot(data=eda_train,x='age_group',hue=eda_train['voted'])
```

<AxesSubplot: xlabel='age_group', ylabel='count'>



• 연령대에 따른 투표여부

```
1 eda_train.loc[eda_train['gender']=='Female','gender']=0
2 eda_train.loc[eda_train['gender']=='Male','gender']=1
3 eda_train['gender']
```

```
0    0
1    0
2    1
3    0
4    1
...
45527 0
45528 1
45529 0
45530 1
45531 1
Name: gender, Length: 45532, dtype: object
```

- 성별 속성의 값을 문자열에서 여자는0 남자는 1로 변경

```
1 eda_train.loc[eda_train['race']=='Asian','race']=0
2 eda_train.loc[eda_train['race']=='Arab','race']=1
3 eda_train.loc[eda_train['race']=='Black','race']=2
4 eda_train.loc[eda_train['race']=='Indigenous Australian','race']=3
5 eda_train.loc[eda_train['race']=='Native American','race']=4
6 eda_train.loc[eda_train['race']=='White','race']=5
7 eda_train.loc[eda_train['race']=='Other','race']=6
8 eda_train['race']
```

```
0    5
1    0
2    5
3    0
4    5
...
45527 5
45528 5
45529 5
45530 5
45531 5
Name: race, Length: 45532, dtype: object
```

- 인종 속성의 데이터를 숫자화

```
1 del eda_train['wf_01']
2 del eda_train['wf_02']
3 del eda_train['wf_03']
4 del eda_train['wr_01']
5 del eda_train['wr_02']
6 del eda_train['wr_03']
7 del eda_train['wr_04']
8 del eda_train['wr_05']
9 del eda_train['wr_06']
10 del eda_train['wr_07']
11 del eda_train['wr_08']
12 del eda_train['wr_09']
13 del eda_train['wr_10']
14 del eda_train['wr_11']
15 del eda_train['wr_12']
16 del eda_train['wr_13']
```

모델링 소개 & 기법 선택

-엑스트라트리

```
min_max_scaler=preprocessing.MinMaxScaler()
X_scale=min_max_scaler.fit_transform(X)
X_train,X_test,y_train,y_test=train_test_split(X_scale,Y,test_size=0.3,random_state=123)
xtree=ExtraTreesClassifier(bootstrap=False, max_features=0.7000000000000001,criterion='gini', min_samples_leaf=14, min_samples_split=7, n_estimators=100)
analy_model=xtree.fit(X_train, y_train)

return analy_model
```

모델링 성능 평가

```
1] 1 model = Sequential([
2     Dense(32, activation='sigmoid', input_shape=(40,)),
3     Dense(32, activation='sigmoid'),
4     Dense(2, activation='sigmoid'),
5 ])

2] 1 model.compile(optimizer=Adam(lr=0.001), loss='binary_crossentropy', metrics=['accuracy'])

4] 1 hist=model.fit(X_train,y_train,batch_size=100, epochs=200)

Epoch 192/200
31872/31872 [=====] - 1s 28us/step - loss: 0.5561 - accuracy: 0.6991
Epoch 193/200
31872/31872 [=====] - 1s 29us/step - loss: 0.5558 - accuracy: 0.6978
Epoch 194/200
31872/31872 [=====] - 1s 30us/step - loss: 0.5560 - accuracy: 0.6982
Epoch 195/200
31872/31872 [=====] - 1s 28us/step - loss: 0.5553 - accuracy: 0.6993 0s - loss: 0.5543 - ac
Epoch 196/200
31872/31872 [=====] - 1s 28us/step - loss: 0.5553 - accuracy: 0.6974
Epoch 197/200
31872/31872 [=====] - 1s 27us/step - loss: 0.5560 - accuracy: 0.6989
Epoch 198/200
31872/31872 [=====] - 1s 26us/step - loss: 0.5554 - accuracy: 0.6987
Epoch 199/200
31872/31872 [=====] - 1s 29us/step - loss: 0.5551 - accuracy: 0.6990
Epoch 200/200
31872/31872 [=====] - 1s 31us/step - loss: 0.5549 - accuracy: 0.6996
```

```
1 from sklearn.ensemble import ExtraTreesClassifier
2

1 xtree=ExtraTreesClassifier(bootstrap=False, max_features=0.7000000000000001, criterion='gini', min_samples_
< >

1 xtree.fit(X_train,y_train)
2 eval=(X_test,y_test)
3 print(xtree.score(X_train,y_train))
4

0.8380396596345381
```

12프로 정도의 정확도 향상

데이터베이스 구축 - SQLite3

Name:	<input type="text" value="왕바우"/>
Question1:	<input type="text" value="매우아니다"/>
Question2:	<input type="text" value="매우아니다"/>
Question3:	<input type="text" value="아니다"/>
Question4:	<input type="text" value="매우그렇다"/>
Question5:	<input type="text" value="그렇다"/>
Question6:	<input type="text" value="매우그렇다"/>
Question7:	<input type="text" value="보통"/>
Question8:	<input type="text" value="그렇다"/>
Question9:	<input type="text" value="그렇다"/>
Question10:	<input type="text" value="매우그렇다"/>
Question11:	<input type="text" value="그렇다"/>

Django의 데이터베이스 테이블을 만들어주는 파일

서버구현 - Django



analy.py
forms.py
ML_model.py
models.py
urls.py
views.py

Django는 총 6개의 파이썬 파일로 구성되어 있습니다.
이중 서버구현과 관련이 있는
forms.py , models.py url.py, views.py 파일을 살펴보겠습니다.

서버구현 – Django

forms.py

```
class AnswerForm(forms.Form):
    Answer_choice = ((1, '매우아니다'), (2, '아니다'), (3, '보통'), (4, '그렇다'), (5, '매우그렇다'))
    age_choice = ((1, '10대'), (2, '20대'), (3, '30대'), (4, '40대'), (5, '50대'), (6, '60대'))
    edu_choice = ((1, '고졸미만'), (2, '고졸'), (3, '대졸'), (4, '석사'))
    engant_choice = ((1, '네'), (2, '아니오'))
    family_choice = ((1, '1명'), (2, '2명'), (3, '3명 이상'))
    gender_choice = ((1, '남자'), (2, '여자'))
    hand_choice = ((1, '오른손'), (2, '왼손'), (3, '양손'))
    married_choice = ((1, '미혼'), (2, '최근 결혼'), (3, '결혼 10년차 이상'), (4, '돌싱'))
    race_choice = ((1, '아시아인'), (2, '아랍'), (3, '흑인'), (4, '백인'), (5, '그 외'))
    religion_choice = ((2, '무교'), (3, '불교'), (4, '크리스찬'), (5, '무슬림'), (6, '그 외'))
    tp_choice = ((0, '매우 강한 부정'), (1, '강한 부정'), (2, '약한 부정'), (3, '매우 약한 부정'), (4, '매우 약한 긍정'), (5, '약한 긍정'), (6, '강한 긍정'), (7, '매우 강한 긍정'))
    urban_choice = ((1, '시골'), (2, '교외 지역'), (3, '도시'))
    Name = forms.CharField(label='이름', required=False)

class AnswerForm(forms.ModelForm):
    class Meta:
        Answer_choice = ((1, '매우아니다'), (2, '아니다'), (3, '보통'), (4, '그렇다'), (5, '매우그렇다'))
        age_choice = ((1, '10대'), (2, '20대'), (3, '30대'), (4, '40대'), (5, '50대'), (6, '60대'))
        edu_choice = ((1, '고졸미만'), (2, '고졸'), (3, '대졸'), (4, '석사'))
        engant_choice = ((1, '네'), (2, '아니오'))
        family_choice = ((1, '1명'), (2, '2명'), (3, '3명 이상'))
        gender_choice = ((1, '남자'), (2, '여자'))
        hand_choice = ((1, '오른손'), (2, '왼손'), (3, '양손'))
        married_choice = ((1, '미혼'), (2, '최근 결혼'), (3, '결혼 10년차 이상'), (4, '돌싱'))
        race_choice = ((1, '아시아인'), (2, '아랍'), (3, '흑인'), (4, '백인'), (5, '그 외'))
        religion_choice = ((2, '무교'), (3, '불교'), (4, '크리스찬'), (5, '무슬림'), (6, '그 외'))
        tp_choice = ((1, '예'), (2, '아니오'))
        urban_choice = ((1, '시골'), (2, '교외 지역'), (3, '도시'))
        model = Result
        fields = ['Name', 'Question1', 'Question2', 'Question3', 'Question4', 'Question5', 'Question6', 'Question7', 'Question8', 'Question9', 'Question10',
                 'Question11', 'Question12', 'Question13', 'Question14', 'Question15', 'Question16', 'Question17', 'Question18', 'Question19', 'Question20',
                 'Question21', 'Question22', 'Question23', 'Question24', 'Question25', 'Question26', 'Question27', 'Question28', 'Question29', 'Question30',
                 'Question31', 'Question32', 'Question33', 'Question34', 'Question35', 'Question36', 'Question37', 'Question38', 'Question39', 'Question40']
```

Django에서 지원하는 파일
Html에서 따로 form 코딩을 하지 않아도
자동으로 세팅이 되어있음

서버구현 – Django Models.py

```
class Result(models.Model):
    Answer_choice = ((1, '매우아니다'), (2, '아니다'), (3, '보통'), (4, '그렇다'), (5, '매우그렇다'))
    age_choice = ((1, '10대'), (2, '20대'), (3, '30대'), (4, '40대'), (5, '50대'), (6, '60대'))
    edu_choice = ((1, '고졸미만'), (2, '고졸'), (3, '대졸'), (4, '석사'))
    engant_choice = ((1, '네'), (2, '아니오'))
    family_choice = ((1, '1명'), (2, '2명'), (3, '3명 이상'))
    gender_choice = ((1, '남자'), (2, '여자'))
    hand_choice = ((1, '오른손'), (2, '왼손'), (3, '양손'))
    married_choice = ((1, '미혼'), (2, '최근 결혼'), (3, '결혼 10년차 이상'), (4, '돌싱'))
    race_choice = ((1, '아시아인'), (2, '아랍'), (3, '흑인'), (4, '백인'), (5, '그 외'))
    religion_choice = ((2, '무교'), (3, '불교'), (4, '크리스찬'), (5, '무슬림'), (6, '그 외'))
    tp_choice = ((0, '매우 강한 부정'), (1, '강한 부정'), (2, '약한 부정'), (3, '매우 약한 부정'), (4, '매우 약한 긍정'), (5, '약한 긍정'), (6, '강한 긍정'), (7, '매우 강한 긍정'))
    urban_choice = ((1, '시골'), (2, '교외 지역'), (3, '도시'))
```

```
Question31 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question32 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question33 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question34 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question35 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question36 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question37 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question38 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question39 = models.IntegerField(null=True, blank=True, choices=tp_choice)
Question40 = models.IntegerField(null=True, blank=True, choices=urban_choice)
voted=models.CharField(max_length=200, blank=True, null=True)
```

Django의 데이터 테이블을 만들
별도의 쿼리 작성 없이 자동으로
생성, 수정 삭제 가능.

서버구현 – Django url.py

```
import ...
app_name='review'
urlpatterns=[
    path('', views.index, name='index'),
    path('after_test/', views.after_test, name='after_test'),
    # path('export/', views.export, name='export'),
]
```

Django의 파일과 웹의 url을 연결해주는 파일
이 파일에서 views.py의 함수이름을 url과 연결해야
만 웹에서 해당 사이트를 볼 수 있습니다.

서버구현 – Django

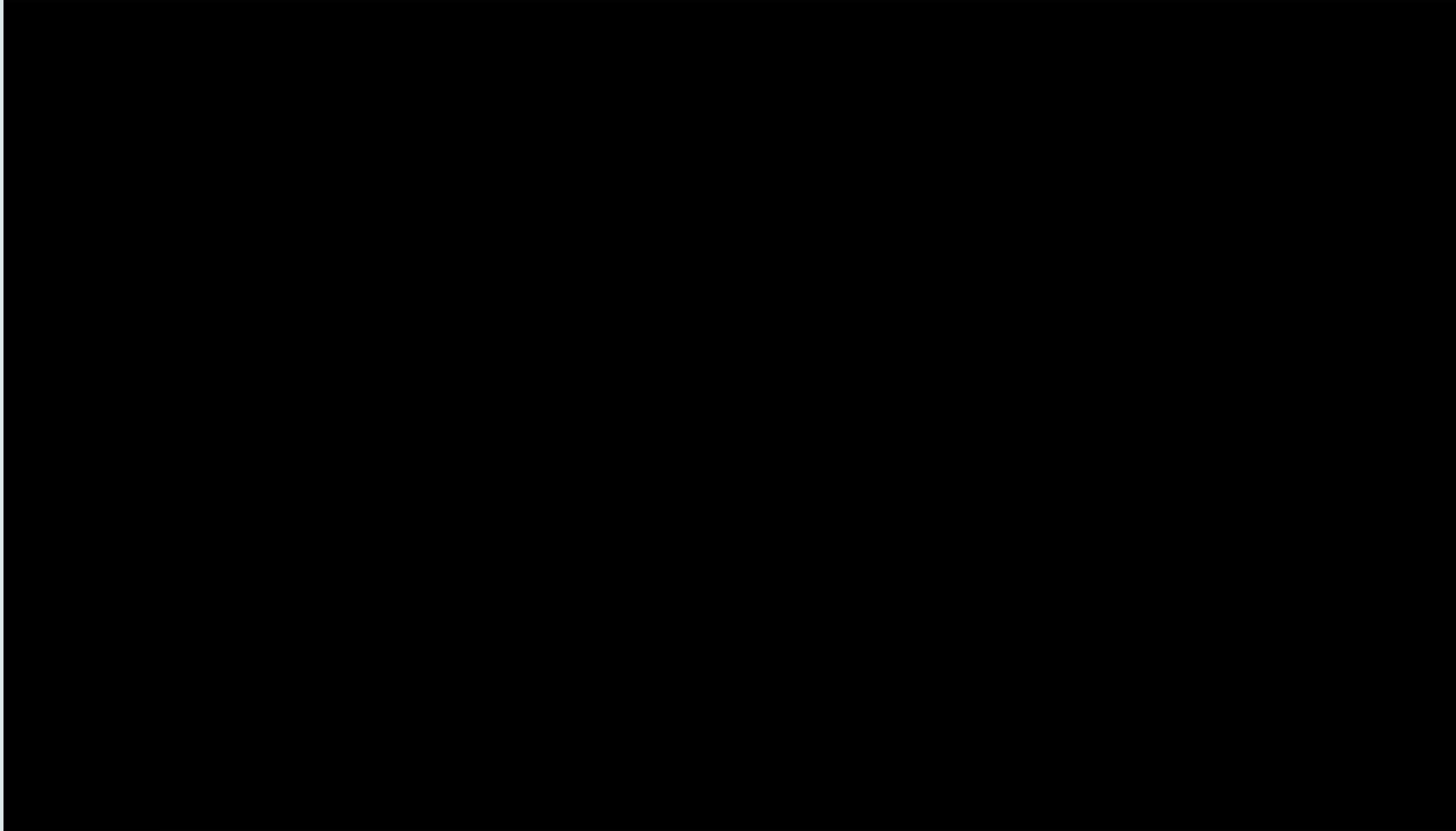
view.py

```
def index(request):  
  
    form=AnswerForm(request.POST or None)  
    if form.is_valid():  
        print("form good")  
        obj=Result()  
        obj.Name=form.cleaned_data['Name']  
        obj.Question=form.cleaned_data['Question']  
        obj.save()  
  
        return HttpResponseRedirect('after_test/')  
    return render(request, 'index.html', {'form': form})  
  
def after_test(request):  
    template=loader.get_template("after_test.html")  
    result= Result.objects.order_by('-pk')[0]  
    context={  
        'result':result  
    }  
  
    return HttpResponse(template.render(context, request))
```

View.py 파일은 index 파트와 after_test 파트로 구성
가장 처음에 나오는 페이지 설정 함수

사용자가 입력한 데이터를 머신러닝 모델에 넣고
결과값을 csv파일로 만듦

구현 영상



결과분석& 프로젝트 후 소감

처음에는 공통으로 원하는 주제를 찾고 문제를 해결하기 위한 목적을 가지고 프로젝트를 진행하는 과정에서 서로 많은 이야기들을 나누면서 진행을 하였습니다. 마키아벨리즘 성향이 투표에 영향을 끼치는 정도에 대해 의문점이 들기도 하였지만 모델을 계속해서 발전시키는 과정에서 서로가 잘 하는 것을 분업을 잘 나누어서 진행을 하게 되었고 최종적으로는 모델의 정확도를 80%까지 끌어올리면서 처음 저희의 목적에 맞는 프로그램을 만들어 낼 수 있는 결과물까지 도달하게 되었습니다. 중간중간 어려운 부분이 많았지만 계속해서 고쳐 나아가면서 원하는 결과물을 뽑아내게 되어서 굉장히 만족스러운 프로젝트가 되었다고 생각합니다. 이후에 좀 더 알맞은 환경과 정서에 맞게 변경한 뒤 적용하게 된다면 더욱 더 유권자들의 투표여부를 예측할 수 있는 발전된 모델이 될 것이라고 자신합니다.

Our TEAM



팀을 짜서 진행한 프로젝트는 처음이었는데 배운 내용을 실제로 활용하면서 결과물을 만들었을 때의 쾌감이 너무 좋았습니다.

데이터 탐색 및 전 처리



프로젝트에 직접 참여해보니 데이터 전 처리부터 서버구현까지 전반적인 흐름을 알 수 있어 좋은 기회가 되었습니다.

데이터 탐색 및 전 처리



짧은 시간동안 많이 배울 수 있는 시간이였고 좋은 팀원분들 프로젝트로 멋진 결과를 낸 것 같아서 좋습니다.

머신러닝 모델링



Django를 처음 사용해보는거라 시간이 부족해서 만족스러울만큼의 결과물을 얻지 못해 아쉽지만 팀원들과 소통하고 프로젝트를 만들어내는 과정에서 배운것도 많고 재밌었습니다

모델링 및 서버 구현

Q&A

Thank you

