



어디 갈래?

앱 개발 아이디어 발표

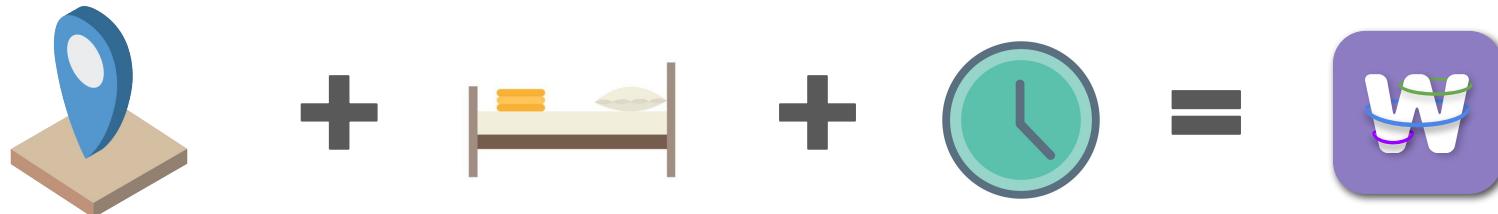
코드스테이츠 AI 18기 한진희

순서

1. 구상 배경
2. 앱 구성
3. 앱 주요기능1, 2
4. 사용자 피드백 수용 및 보완

1. 어디 갈래? 구상 배경

- 여행을 떠나기 전 구체적인 계획을 세우는 사람이 있고 대략적인 계획을 세우는 사람이 있다. 대략적인 계획을 세우는 사람이라도 어디를 보고, 어디서 묵을지 정도는 생각한다.
- 요즘 ‘에어비앤비’, ‘야 놀자’ 같은 여러 가지 앱로 쉽게 예약하고 찾아볼 수 있어서 이런 앱들이 등장하기 전보다는 훨씬 수월해졌다.
- 하지만 플랫폼이 다양한 만큼 여기저기서 찾고 예약하다 보면 혼란스럽고 계획 세우는 게 어려운 사람들에게는 어렵기만 하다.
- 그래서 관광지 검색, 숙소 예약, 일정 세우기 이 세 가지를 한 번에 할 수 있으면 좋겠다는 생각이 들어 ‘어디 갈래?’ 앱을 구상하게 됐다.



2. 어디 갈래? 구성



1. 메뉴바 : 관심 여행지, 마이페이지, 여행 계획표, 최근 본 게시물, 설정, 고객센터를 볼 수 있다.
2. 지도 : 지도의 지역을 선택하여 가려는 여행지를 고른다.
3. : 여행지, 숙소 등 정보를 보여 준다.



3. 어디 갈래? 주요기능1 : 관광지 둘러보기

전체 강릉시 고성군 동해시 삼척시 양구군 영월군
원주시 인제군 정선군 철원군 춘천시 태백시 평창군
홍천군 화천군 횡성군

강원도 전체 방문자 추이

연도	증감률(%)
2023	~10%
2024	~15%
2025	~10%
2026	~12%
2027	~10%
2028	~10%
2029	~10%
2030	~10%
2031	~10%
2032	~15%

자연 속초해변 대포항 목호항 경

속초해변 대포항 목호항 경

맛집

속초해변

강원 속초시 조양동
매일 06:00 - 24:00 수영 가능시간(09:00~18:00)
033-639-2027
<http://www.sokchotour.com>

관심

▶ 주변 맛집

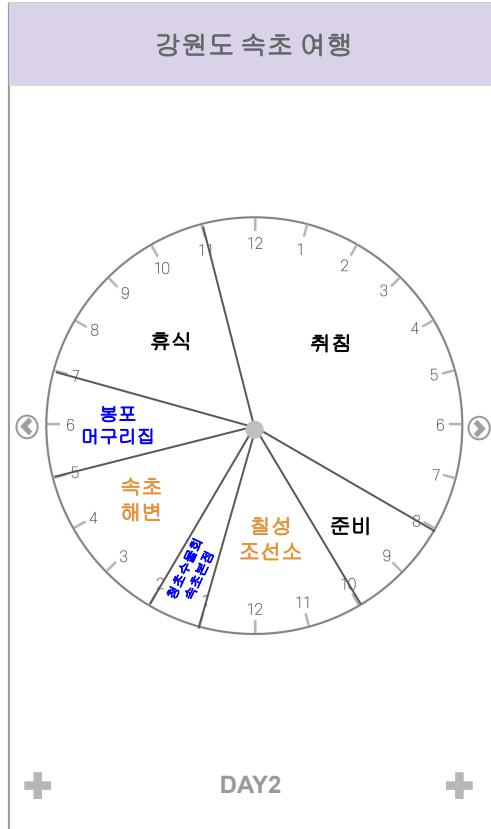
청초수물회 속초본점 봉포머구리집 만석닭강정 본점 88

▶ 주변 숙소

- 지도에서 지역을 누르면 전체 단위로 자연 관광지, 쇼핑 관광지, 문화 관광지, 역사 관광지, 맛집, 숙소를 살펴 볼 수 있고 시/군/구를 선택하면 선택한 곳에 맞는 정보를 볼 수 있다.
- 특정 관광지를 선택하면 관광지의 주소, 영업시간, 전화번호, 홈페이지 등의 정보를 볼 수 있다.
- 사진 아래 하트♥를 눌러 관심 여행지에 저장 ♥ 할 수 있다.
- 특정 관광지의 주변 맛집과 주변 숙소의 정보를 볼 수 있다.



3. 어디 갈래? 주요 기능2 : 일정 계획하기



- 메뉴바 의 여행 계획표에서 일정을 작성한다.
- 시간을 누르면 글자를 입력할 수 있다.
- 관심 여행지에 있는 여행지를 적용시킬 수 있다.
- 를 눌러 일자를 추가할 수 있다.



4.



어디 갈래? 사용자 피드백 수용 방법

- 사용자의 리뷰를 분석해서 긍정적인(**Positive**) 리뷰인지 부정적인(**Negative**) 리뷰인지 판단하는 머신러닝을 도입한다.
- Negative로 판단된 리뷰를 수용하여 앱을 개선해 나간다.
- (현재는 데이터가 없는 관계로 유산한 데이터를 이용하였다.)
- 이용 데이터 : 사용자 호텔 리뷰 데이터
- 특성 : Review(리뷰), Date of stay(작성일자), Rating(평점)

	Review	Date of stay	Rating
0	Most beautiful stay at Mussoorie...such a gorg...	Date of stay: August 2021	
1	NaN	NaN	NaN
2	The Savoy is one of those rare hotels that not...	Date of stay: October 2020	
3	NaN	NaN	NaN
4	Def Best stay in Uttarakand ! Amazing nature. ...	Date of stay: November 2021	

4. W 어디 갈래? 사용자 피드백 수용 방법 예시

1. 전체적인 전처리 과정

- 1) 결측치 및 중복치 제거
- 2) Rating에서 평점(10, 20, 30, 40, 50) 이외의 문자열 삭제
- 3) Sentiment 특성 생성 : 50과 40은 Positive, 30은 Neutral, 20과 10은 Negative로 가정
- 4) Date fo stay에서 year과 month를 분리하여 각 특성으로 생성
- 5) Sentiment가 Neutral인 행과 2018년 이전에 작성된 행은 삭제
- 6) 불필요한 특성(Date of stay) 삭제 및 인덱스 리셋

	Review	Rating	Sentiment	year	month
0	Most beautiful stay at Mussoorie...such a gorg...	5	Positive	2021	August
1	The Savoy is one of those rare hotels that not...	5	Positive	2020	October
2	Def Best stay in Uttarakand ! Amazing nature. ...	5	Positive	2021	November
3	Very very friendly and efficient staff ... loved...	5	Positive	2021	November
4	Exceptional property. Just great experience. V...	5	Positive	2021	October

4. W 어디 갈래? 사용자 피드백 수용 방법 예시

2. Review 특성 전처리를 위한 함수 생성

- 1) 분석에 불필요한 단어들 삭제 및 소문자로 변환한다.
- 2) 불필요한 단어 : 숫자, 구두점, urls 관련 단어, html 관련 단어, 이모지

```
# 11. 리뷰 전처리 함수 생성
def preprocess_data(text):

    text = re.sub(r'[0-9]+', '', str(text))      # 숫자 삭제
    text = re.sub(r'[\^\\w\\s]', '', str(text))    # 구두점 삭제
    text = " ".join(x.lower() for x in text.split()) # 소문자로 변환
    text = re.sub(r'https://\S+|www.\S+', '', text) # urls 삭제
    text = re.sub(r'<.*?>', '', text) # html 삭제
    emoji_pattern = re.compile("["
                                u"\U0001F600-\U0001F64F" # 이모지 삭제
                                u"\U0001F300-\U0001F5FF"
                                u"\U0001F680-\U0001F6FF"
                                u"\U0001F1E0-\U0001F1FF"
                                u"\U00002702-\U000027B0"
                                u"\U000024C2-\U0001F251"
                                "]+", flags=re.UNICODE)
    text = emoji_pattern.sub(r'', text)
    text = " ".join(x for x in text.split() if x not in StopWords.words('english'))
    return text
```

4. W 어디 갈래? 사용자 피드백 수용 방법 예시

3. 타겟(Sentiment) 인코딩

- Positive, Negative로 지정되어 있으므로 인코딩

```
# 17. Sentiment 인코더(1: pos, 0: neg)
le = LabelEncoder()
df['Sentiment'] = le.fit_transform(df['Sentiment'])
```

4. 사용할 특성(Review) 벡터화

- TF-IDF 사용 : TF(단어가 자주 등장하는지 계산), IDF(단어가 얼마나 많이 등장하는지 계산)
- 2번에서 생성한 함수(preprocess_data) 사용

```
# 19. TF-IDF 으로 벡터로 바꿈
tfv = TfidfVectorizer(min_df=3, max_features=None, decode_error = "replace", preprocessor = preprocess_data,
                      strip_accents='unicode', analyzer='word',token_pattern=r'\w{1,}', 
                      ngram_range=(1, 3), use_idf=1,smooth_idf=1,sublinear_tf=1,
                      stop_words = 'english')

tfv.fit(list(xtrain))

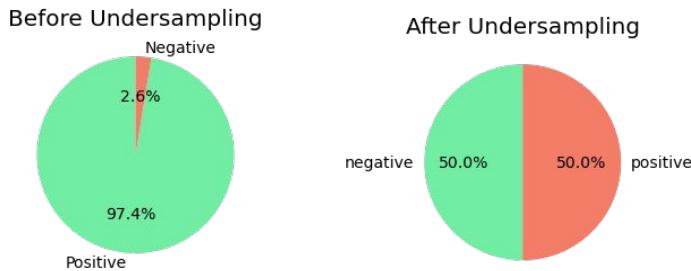
xtrain_tfv = tfv.transform(xtrain)
xval_tfv = tfv.transform(xval)
```

4. 어디 갈래? 사용자 피드백 수용 방법 예시

5. 타겟 데이터 불균형 해소

- 1) RandomUndersampling 사용 : 높은 비율 클래스를 낮은 비율 클래스에 맞게 줄인다.

```
# 20. RandomUndersampling  
Xtrain_samp, ytrain_samp = RandomUnderSampler(random_state=42).fit_resample(xtrain_tfv, ytrain)
```



6. 성능 평가 지표 선정

- 1) 정확도(accuracy)와 로그손실(logloss) 사용
- 2) 정확도 accuracy : 전체 데이터 중 모델이 정확하게 예측한 데이터 수의 비율(1에 가까울수록 성능이 좋음)
- 3) 로그손실 logloss : 실제 값을 예측하는 확률에 로그를 취해 부호를 반전시킨 값(0에 가까울수록 좋음)

4. W 어디 갈래? 사용자 피드백 수용 방법 예시

7. 기준 모델 선정

- 1) MultimnoialNB모델 : 각 특성의 확률을 계산하여 새로운 샘플이 주어졌을 때 해당 샘플이 각 클래스에 속할 확률을 계산하고 가장 높은 확률을 가진 클래스로 분류한다.
- 2) 성능 확인
 - 정확도 accuracy : 0.884
 - 로그손실 logloss : 0.479

```
# 23. MNB 모델
MNB = MultinomialNB()
MNB.fit(Xtrain_samp, ytrain_samp)

ypred_xg = MNB.predict(xval_tfv)
print(classification_report(yval, ypred_xg))
print("accuracy:", accuracy_score(yval , ypred_xg).round(3))

predictions = MNB.predict_proba(xval_tfv)
print ("logloss: %0.3f " % log_loss(yval, predictions))
```

4. W 어디 갈래? 사용자 피드백 수용 방법 예시

8. 모델 1

- 1) xgb 모델 : 이전 트리의 오차를 다음 트리에 반영하는 방식
- 2) 성능 확인
 - 정확도 accuracy : 0.857 (기준모델 : 0.884)
 - 로그손실 logloss : 0.338 (기준모델 : 0.479)

```
# 22. xgb 모델
xgb = XGBClassifier()
xgb.fit(Xtrain_samp, ytrain_samp)

ypred_xg = xgb.predict(xval_tfv)
print(classification_report(yval, ypred_xg))
print("accuracy:", accuracy_score(yval , ypred_xg).round(3))

predictions = xgb.predict_proba(xval_tfv)
print("logloss: %0.3f " % log_loss(yval, predictions))
```

4. W 어디 갈래? 사용자 피드백 수용 방법 예시

9. 모델 2

- 1) svc 모델 : 데이터를 분류하는 경계선을 찾는다.
- 2) 성능 확인
 - 정확도 accuracy : 0.967 (기준모델 : 0.884)
 - 로그손실 logloss : 0.123 (기준모델 : 0.479)

```
# 24. svc 모델
svc = SVC(kernel='linear', probability=True)
svc.fit(Xtrain_samp, ytrain_samp)

ypred_xg = svc.predict(xval_tfv)
print(classification_report(yval, ypred_xg))
print("accuracy:", accuracy_score(yval , ypred_xg).round(3))

predictions = svc.predict_proba(xval_tfv)
print ("logloss: %0.3f " % log_loss(yval, predictions))
```

4.



어디 갈래? 사용자 피드백 수용 방법 예시

10. 최종 모델 설정 : svc 모델

1) 성능 확인

- 정확도 accuracy : 0.966 (훈련성능 : 0.967 / 기준모델 : 0.884)
- 로그손실 logloss : 0.126 (훈련성능: 0.123 / 기준모델 : 0.479)

```
# 28. 모델 티닝 학습
from sklearn.model_selection import GridSearchCV
params = {'C': [1,4,8,16,32], 'kernel' : ['linear','rbf']}
svc = SVC()
svc_grid = GridSearchCV(svc,params, cv = 5)
svc_grid.fit(Xtrain_samp,ytrain_samp)
```

11. 보안 사항

- 데이터들을 더 추가하여 모델의 성능을 높인다.
- 현재는 모델이 분류한 리뷰들을 개발자들이 직접 읽어야 하는 번거로움이 있다.
- 리뷰에서 중요한 핵심만 간추려 개발자들의 번거로움과 시간을 줄일 수 있도록 한다.
- 앱의 지역 범위를 국내에서 국외로 확장 시킨다.



감사합니다

앱 개발 아이디어 발표

코드스테이츠 AI 18기 한진희