 반드시 문 문제에 변 자격인증 오픈북 허 [데이터 컬럼 RID : 경로 	
 Time_Arri Distance Time_Driv Speed_Pe Address1 Address2 Weekday Hour : Tim Day : Tim 	
• RID : 경토 • Signaltype 1. scikit-le scikit-lear	arn 패키지는 머신러닝 교육을 위한 최고의 파이썬 패키지입니다. 마를 별칭(alias) sk로 임포트하는 코드를 작성하고 실행하세요.
import skles 2. Pandas Pandas	는 데이터 분석을 위해 널리 사용되는 파이썬 라이브러리입니다. 사용할 수 있도록 별칭(alias)을 pd로 해서 불러오세요.
Pandas 함 • A0007IT.ji • signal.csv • df_a와 df	을 위해 분석 및 처리할 데이터 파일을 읽어오려고 합니다. 수로 2개 데이터 파일을 읽고 합쳐서 1개의 데이터프레임 변수명 df에 할당하는 코드를 작성하세요. eon 파일을 읽어 데이터 프레임 변수명 df_a에 할당하세요. 파일을 읽어 데이터 프레임 변수명 df_b에 할당하세요. b 데이터프레임 변수명 df_b에 할당하세요. # 사용하는 키(on): 'RID' # 사용하는 키(on): 'RID'
# 0700 답안 import panda df_a = pd.redf_b = pd.redf = pd.meredf	= 방법(how) : 'inner' 코드를 작성하세요.
2 router-8 3 router-8 9995 router- 9996 router-	84875df7fc-savgi-6-13366366-0 12:23:05 12:36:32.62000 3858.0 796.081970 17.446445 경기도 의정부시 4 12 24 17.0 4875df7fc-Savgs-6-62202063-0 19:10:35 19:29:07.306000 4125.0 1112.468018 13:348698 경기도 광명시 0 19 27 24.0 34875df7fc-8n4sw-5-6914479-0 11:59:58 12:10:55:999000 3748.0 651.973022 20:695335 경기도 안산시 상록구 0 11 20 19.0 4875df7fc-8n4sw-7-75115963-0 15:16:53 15:26:52.222000 3153.0 597.419006 18:999730 경기도 성납시 중원구 2 15 29 14.0
9998 router-8 9999 router- 10000 rows × 2	84875df7fc-52vz8-6-53906748-0 08:25 2 08:40:26.18200 1044.0 890.31298 42.39228 경기도 안산시 단원구 1 8 21 21.0 4875df7fc-52vz8-6-53906748-0 06:49:59 06:55:01.08700 65:20 300.302002 78.26928 경기도 성남시 분당구 4 6 24 10.0 84875df7fc-52vz8-6-53906748-0 23:09:24 23:25:32:43200 7976.0 969.36200 29.621132 서울특별시 마포구 4 23 24 6.0 12.0 12.0 12.0 12.0 12.0 12.0 12.0 12
## seaborn을 한 첫번째, A F번째, X 출력된 그 1. Coun 2. Address	cottis person as pit cottis form, amanger as in
sns.countpledf = df[(d:df.info())] EP04 = 3 class 'panda Int64Index: 9 Data columns # Column 0 RID	교트 작성하세요. tr(data=df, x='Addressl') tr(vaddressl') != '-')] s.core.frame.DataFrame'> 997 entries, 0 to 9999 (total 12 columns):
3 Distance 4 Time_Dri 5 Speed_Pe 6 Address 7 Address 8 Weekday 9 Hour 10 Day 11 Signalty	ving 9997 non-null float64 r_Hour 9997 non-null float64 9997 non-null object 9997 non-null int64 64(4), int64(3), object(5)
5000 - 4000 - 3000 - 2000 -	
5. 실주행	경기도 서울특별시 인천광역시
# 여기에 달인 import seabo	ime_Driving(실주행시간)을 표시하고 Y축에는 Speed_Per_Hour(평균시속)을 표시하세요. 코드를 작성하세요.
300 - 250 - 200 -	
150 - 100 -	
가이드에 • 대상데이	1000 2000 3000 4000 5000 Time_Driving ntplot 그래프에서 시속 300이 넘는 이상치를 발견할수 있습니다. 다라서 전처리를 수행하고 저장하세요. 터프레임: df R대프를 보고 시속 300 이상되는 이상치를 찾아 해당 행(Row)을 삭제하세요.
• 불필요한 • 전처리 반 # 여기에 답인 df_temp.drop df_temp.info <class #="" 'panda="" 0="" 9="" column="" columns="" data="" int64index:="" td="" time_dep<=""><td>RID' 컬럼을 삭제 하세요. 영후에 새로운 데이터프레임 변수명 df_temp 에 저장하세요. 코드를 작성하세요. [dd. Speed_Per_Hour < 300] (YRID', axis=1, inplace=True) ()() s.core.frame.DataFrame'> 996 entries, 0 to 9999 (total 11 columns):</td></class>	RID' 컬럼을 삭제 하세요. 영후에 새로운 데이터프레임 변수명 df_temp 에 저장하세요. 코드를 작성하세요. [dd. Speed_Per_Hour < 300] (YRID', axis=1, inplace=True) ()() s.core.frame.DataFrame'> 996 entries, 0 to 9999 (total 11 columns):
5 Address2 6 Address2 7 Weekday 8 Hour 9 Day 10 Signalty dtypes: float memory usage: C:\Users\1009 A value is tr	9994 non-null float64 ving 9996 non-null float64 vr_Hour 9996 non-null float64 9996 non-null object 9996 non-null int64 9996 non-null it64 9996 non-null it64 9996 non-null it64 9996 non-null it64 9996 non-null float64
7. 모델링 아래 가이	성능을 제대로 얻기 위해서 결축치 처리는 필수입니다. 드를 따라 결축치 처리하세요.
 결측치가 전처리 반 결측치 개 # 여기에 답안 df_temp.ism 	emp.dropna()
아래 가이 • 대상 데이 • 'Time_De • 전처리 반	성능을 제대로 얻기 위해서 불필요한 변수는 삭제해야 합니다. 드를 따라 불필요 데이터를 삭제 처리하세요. 터프레임: df_na parture, 'Time_Arrival' 2개 컬럼을 삭제하세요. 경된 결과를 새로운 데이터크레임 변수명 df_del 에 저장하세요. 코트를 작성하세요. pas.drop(['Time_Departure', 'Time_Arrival') , exis=1)
 0 3246. 1 3858. 2 4125. 3 3748. 4 3153. 	*** Time_Driving Speed_Per_Hour Address Addres
9997 10484. 9998 6529.	1 1511.703003 14.931504 서울특별시 서초구 1 16 28 17.0 1 890.312988 42.392283 경기도 안산시 단원구 1 8 21 21.0 2 300.302002 78.269208 경기도 성납시 분당구 4 6 24 10.0 3 969.362000 29.621132 서울특별시 마포구 4 23 24 6.0 columns
원-핫 인코 대상 데이 원-핫 인코 활용 함수 해당 전처 # 여기에 답안 df_preset =	!코딩(One-hot encoding)은 범주형 변수를 1과 0의 이진형 벡터로 변환하기 위하여 사용하는 방법입니다. 라딩으로 아래 조건에 해당하는 컬럼 데이터를 변환하세요. 터프레임 : df_del I당 대상 : object 타입의 전체 컬럼 : Pandas의 get_dummies 리가 반영된 결과를 데이터프레임 변수 df_preset에 저장해 주세요. 코드를 직성하세요. pd. (get_dummies (data=df_del, columns=['Address1', 'Address2'])
Int64Index: 9 Data columns # Column 0 Distance 1 Time_Dri 2 Speed_Pe 3 Weekday 4 Hour 5 Day 6 Signalty 7 Address 8 Address	Secondary Seco
10 Address2 11 Address2 12 Address2 13 Address2 14 Address2 15 Address2 16 Address2 17 Address2 18 Address2 20 Address2 21 Address2 22 Address2 23 Address2	2강부구 994 non-null vint8 2강부구 9994 non-null vint8 고양시 영양구 9994 non-null vint8 고양시 일산동구 9994 non-null vint8 고양시 일산동구 9994 non-null vint8 고양시 일산자구 9994 non-null vint8 관련시 9994 non-null vint8
24 Address2 25 Address2 26 Address2 27 Address2 28 Address2 30 Address2 31 Address2 32 Address2 33 Address2 34 Address2 35 Address2 36 Address2 37 Address2 38 Address2 39 Address2	고리시 9994 non-null uint8 금본가 9994 non-null uint8 금본가 9994 non-null uint8 나물자 9994 non-null uint8 나물자 9994 non-null uint8 노월구 9994 non-null uint8 도봉자 9994 non-null uint8 통무사 9994 non-null uint8 통무사 9994 non-null uint8 무목자 9994 non-null uint8 무목자 9994 non-null uint8 무목자 9994 non-null uint8 무무자 9994 non-null uint8 무무자 9994 non-null uint8 무무자 9994 non-null uint8
40 Address2 41 Address2 42 Address2 43 Address2 44 Address2 45 Address2 46 Address2 47 Address2 48 Address2 50 Address2 51 Address2 52 Address2 53 Address2	_서대문구 9994 non-null uint8서초구 9994 non-null uint8성남시 분당구 9994 non-null uint8성남시 수정구 9994 non-null uint8성남시 수정구 9994 non-null uint8성남시 중원구 9994 non-null uint8성남시 중원구 9994 non-null uint8성당구 9994 non-null uint8성당구 9994 non-null uint8성당구 9994 non-null uint8
56 Address2 57 Address2 58 Address2 60 Address2 61 Address2 62 Address2 63 Address2 64 Address2 65 Address2 66 Address2 67 Address2 68 Address2 69 Address2	994 non-null uint8 994 non-null uint8 여주시 9994 non-null uint8 여전군 9994 non-null uint8 여전군 9994 non-null uint8 여전군 9994 non-null uint8 의용자 9994 non-null uint8 용사 9994 non-null uint8
70 Address2 71 Address2 72 Address2 73 Address2 74 Address2 75 Address2 76 Address2 77 Address2 78 Address2 79 Address2 80 Address2	의정시 9994 non-null vint8 -이정시 9994 non-null vint8 -이정시 9994 non-null vint8 -중로구 9994 non-null vint8 -중당구 9994 non-null vint8 -중당구 9994 non-null vint8 -파주시 9994 non-null vint8 -파주시 9994 non-null vint8 -파주시 9994 non-null vint8
Time_Driv	H 검증 각각에 사용할 데이터셋을 분리하려고 합니다. ing(실주행시간) 컬럼을 label값 y로, 나머지 컬럼을 feature값 X로 할당한 후 훈련데이터셋과 검증데이터셋으로 분리하세요이드 따라서 훈련데이터셋과 검증데이터셋에 스케일링을 수행하세요. 터프레임: df_preset 중 데이터셋 분리 데이터셋 label: y_train, 훈련 데이터셋 Feature: X_train 데이터넷 label: y_valid, 검증 데이터셋 Feature: X_valid
■ rando ■ Scikit ● RobustSc ■ sklea ■ 훈련대 ■ 검증대 # 여기에 답인 X = df_prese	레이터셋과 검증데이터셋 비율은 80:20 m_state: 42 learnal train_test_split 함수를 활용하세요. lin_test_rest_rest_rest_rest_rest_rest_rest
<pre>from sklear X_train, X_ print(X_tra: from sklear rs = Robust: X_train = r: X_test = rs</pre> X, y shape:	<pre>shape: ', X.shape, y.shape)model_selection import train_test_split raid, y_train, y_valid = train_test_split(X, y, test_size=0.2, random_state=42)shape, X_valid.shape, y_train.shape, y_valid.shape)preprocessing import RobustScaler scaler()fit_transform(X_train) transform(X_train) transform(X_valid) (9994, 80) (9994,) 999, 80) (7995,) (1999,)</pre>
의사결정나 독립 변수 아래 가이 • 의사결정나 • 의사결정나 • 보드를 • rando • 한테포레: • 랜덤포레: • 보드를 • 노드를 • rando	Driving(실주행시간)을 예측하는 머신러닝 모델을 만들려고 합니다. 나무(decision tree)와 랜덤포레스트(RandomForest)는 여러 가지 규칙을 순차적으로 적용하면서 공간을 분할하는 모형으로 분류(classification)와 회귀 분석(regression)에 모두 사용될 수 있습니다. 드에 따라 의사결정나무(decision tree)와 랜덤포레스트(RandomForest) 모델 만들고 학습을 진행하세요. 서무(decision tree) 레제당 강이 : 5로 설정 활 분환하기 위한 최소한의 생물 데이터수(min_samples_spit) : 3로 설정 m_state : 120로 설정 원리무(decision tree) 모델을 데 변수에 저장해 주세요. 트(RandomForest) 레제당 강이 : 5로 설정 용보하기 위한 최소한의 생물 데이터수(min_samples_spit) : 3로 설정 m_state : 120로 설정
# 07/01 달인 from sklears from sklears dt = Decision dt.fit(X_tro	ERJO E(RandomForest) 모델을 f 변수에 저장해 주세요. 네 모델에 대해 ft을 활용해 모델을 학습해 주세요. 학습 시 훈련데이터 셋을 활용해 주세요. 코르를 작성하세요tree import DecisionTreeRegressor .ensemble import RandomForestRegressor intreeRegressor(max_depth=5, min_samples_split=3) intr, y_train) **Corporation of the proof of th
아래 가이• 성능 평가• 11번 문제• 검증 정답• 11번 문제• 검증 정답• 검증 정답	나결정나무(decision tree)와 랜덤포레스트(RandomForest) 모델의 성능을 평가하려고 합니다. 드에 따라 예측 결과의 mae(Mean Absolute Error)를 구하고 평가하세요. 는 검증 데이터셋을 활용하세요. 에서 만든 외사결정나무(decision tree) 모델로 맛값을 예측(predict)하여 y_pred_df0에 저장하세요. (y_valid)과 예측값(y_pred_df)의 mae(Mean Absolute Error)를 구하고 d_mae 변수에 저장하세요. (y_valid)과 예측값(y_pred_f)의 mae(Mean Absolute Error)를 구하고 d_mae 변수에 저장하세요.
• 2개의 모델 # 여기에 답인 y_pred_dt = y_pred_rf = from sklear dt_mae = mea print(dt_mae	Und 대한 mae 성능평가 결과을 확인하여 성능좋은 모델 이름을 답안12 변수에 저장하세요. [안12 = 'decisiontree' 혹은 답안12 = 'randomforest' 코드를 작성하세요. dt.predict(X_valid) rf.predict(X_valid) n.metrics import mean_absolute_error in_absolute_error(y_valid, y_pred_dt) in_absolute_error(y_valid, y_pred_ft) in_absolute_error(y_valid, y_pred_ft) in_atsolute_error(y_valid, y_pred_ft) in_atsolute_error(y_valid, y_pred_ft)
import tensor from tensor from tensor from tensor	#함을 끌기전에 아래 코드를 실행하세요. Porflow as tf flow.keras.models import Sequential, load_model flow.keras.loalyers import Dense, Activation, Dropout, BatchNormalization flow.keras.callbacks import EarlyStopping, ModelCheckpoint
from tensor: tf.random.se 13. Time_ 아래 가이 • Tensorflow • 히든레이야	llow.keras.utila import to_categorical st_seed(1) Driving(실주행시간)을 예측하는 딥러닝 모델을 만들려고 합니다. 드에 따라 모델링하고 학습을 진행하세요. of framework를 사용하여 딥러닝 모델을 만드세요. H(hidden layer) 2개이상으로 모델을 구성하세요.
• dropout 변 • 손실함수는 • 하이퍼파를 • 각 에포크 • 학습정보는 # 여기에 답인 model = Sequence model.add(Demodel	Be 0.2로 Dropout 레이어 1개를 추가해 주세요. MSE(Mean Squared Error)를 사용하세요. H이터 epochs: 30, batch_size: 16 으로 설정해 주세요. 마다 loss와 metrics 평가하기 위한 데이터로 x_valid, y_valid 사용하세요. 마다 loss와 metrics 평가하기 위한 데이터로 x_valid, y_valid 사용하세요. = history 변수에 저진해 주세요. ###################################
model.add(Demodel.add(Demodel.compilhistory = model.compilhistory = model.add(Demodel.add(######################################
Epoch 7/30 Epoch 7/30 Epoch 8/30 Epoch 8/30 Epoch 9/30 Epoch 9/30 Epoch 10/30 Epoch 11/30 Epoch 11/30 Epoch 12/30 Epoch 13/30 Epoch 13/30 Epoch 13/30 Epoch 14/30	
Epoch 14/30 500/500 [==== Epoch 15/30 500/500 [==== Epoch 16/30 500/500 [==== Epoch 17/30 500/500 [==== Epoch 18/30 500/500 [==== Epoch 20/30 500/500 [==== Epoch 21/30 500/500 [=====	
Epoch 22/30 500/500 [==== Epoch 23/30 500/500 [==== Epoch 24/30 500/500 [==== Epoch 25/30 500/500 [==== Epoch 26/30 500/500 [==== Epoch 27/30 500/500 [==== Epoch 28/30 500/500 [====	
500/500 [==== Epoch 30/30 500/500 [==== 14. 위 딥 Matplotlib • 1개의 그리 • 위 2가지 • 그래프의 • X축에는 '	
[1734081091 2986930392 3832141368 4515614110 4944771520 5695370742 6174507899 6515173464 7202061208 7603743830 8211922105 8874132242 8883306024 9700628234 1004300675	2680.0, 5152.0, 9216.0, 9888.0, 5288.0, 9888.0, 9888.0, 9888.0, 9889.0, 9736.0, 98352.0, 1392.0, 1392.0, 1392.0, 1408.0, 1408.0,
1004300675 1046784528 1102574056 1101736370 1029238580 1199890549 1215471550	31776.0, 87552.0, 89555.0, 950568.0, 950568.0, 958480.0, 964480.0, 964400.0, 41088.0, 20960.0, 20660.0, 26640.0, 67712.0,
1211487884 1281244109 1303019677 1330772044 1270134320 1337422868 1373667275 1428035521 1377069778	64384.0, 57120.0, 74080.0,

AICE Associate 자격인증 샘플문항

내비게이션 주행데이터를 이용한 도착시각 예측 문제

