Veri setinin ilk 5 satırı movie_id movie_title genre o tt1745960 Top Gun: ['Action Mayerick 'Drame	Well, this film and surpasses length featu rw8168012 The so	its predecessor on every lefilm. All the stunts, visual undtrack and score was ar	level.The directing by no als and V/SFX were brea mazing. The 131 min rur	oumping edge or your seat ac ovice Joseph Kosinski was out othtaking, and the camera wo ntime literally flew by with its	ction, this story also had standing, especially cons rk perfection, that you'll spot-on pacing. I actual	sidering this was his four feel you're in the cockp lly wanted to see more. (othing and	736.0	0.88
Maverick 'Drama' 1 tt1745960 Top Gun: ['Actio Maverick 'Drama'	performances critique or w being a vis If you were a We had cars thrilling cinema t perfectly re own childred missir In one of the body can'd further tha	by all were also perfection ish was better, as this genter and masterpiece. And the magnificent. I'll late teen or in your early though, and bikes, and we ride like we had never see to see it again and again, a epresent that magical point who have moved on them g bookend to that whole genter memorable lines in the cash." Sometimes I wonder Cruise these days. Ridicular	n, and once again Tom Com was perfect in every we fact the dog fights were I be definitely seeing this twenties in the mid 1980 and spent the following don't in our lives for so many generation. The original was the original Top Gun, Market if Tom Cruise took the allously entertaining Top or the mass of the control of the original was the original top of the original Top Gun, Market origi	ruise reminds us why he is stray. Absolutely stunning, action real and not CGI (actors had a again and adding the DVD to D's the world was very different ed films too. The original Topiche games, the bikes, the airous decades quoting the movie. Any of us.Now, 36 years later, where the end of our own can be start of our magnificent.I'm 55, but you werick gets chewed out by a fat putdown as a personal changun: Maverick. Cruise was in	ill one of the top actors in packed with a compel to train to fly the jets), just on my collection. It's a raint. No computers, no more Gun captured this mome raft and my word, those is time went on, it remains are a generation that incers and our young self young adult lives, and the esterday, just for one later superior who tells him, "superior who tells him," the searly 20s when he fill to train the superior who tells him, "superior who tells him,"	in the industry. There's no lling and emotional story just makes this film that are perfect 10/10 must see obile phones, no interned tent in time perfectly, and effying scenes. We went intend like a static snapshown has lost our parents, we wes are gone forever. This new film now marks the intended in the state of t	othing I can , as well as much more e from me. , no DVD's. d gave us a back to the t in time to ve had our s film is the he end. It's Thank you. hecks your ush himself k" Mitchell,	330.0	0.95
2 tt1745960 Top Gun: ['Actio Maverick 'Drama	the insubordinate where he pust and bounds. I', rw8173679 Impossible movie soundtrack tugs at your hook cockpite respect, Top Gong You start to the heads, and advancin advancin this is one other issues. I	e cocky young Navy pilot was ever: Now a Navy test hes a new plane beyond its ightest for an impossibly owriter-director, Christopher of truly epic proportions. The features a Lady Gaga song teartstrings nonetheless. The with these pilots, and that inderstand why Cruise, the state-of-the-art F-18s duke ga case for the enduring a sequel that looked like it was almost as if we were not	with the aviator sunglass pilot in his late 50s, Maves limits. Partly as punish dangerous new mission. If McQuarrie - have taker hey're tapping into nosting, "Hold My Hand." It's not he action sequences are at's because you are: The chrowback to a lost era one creative force behind the it out with rusty old F-1 appeal of the movies and would never get made. It the meant to see a sequel to the movies are the second of the movies and the meant to see a sequel to the movies and the meant to see a sequel to the meant to see a sequel to the movies and the meant to see a sequel to the movies and the meant to see a sequel to the meant	ses, the Kawasaki motorcycle verick still knows how to tick of ment, he's ordered to return And so the three screenwrite in the threads of the original attalgia for the original, while a nowhere near as iconic a charge much more thrilling and im a eactors underwent intense for practical moviemaking, before he movie, was so driven to make their power to transport us the two to the total movie to the total m	and the need for speed. off his superiors, as we so to TOPGUN, the elite pile of the spun them into an infining for new levels of extra topper as the original representation of the original state of the second and flew according to the second and flew according to the second of the original state of the second of the original state of the second for the original state of the second for the original for years and years. In the second of the second	In the sequel, he's as an see in an exciting opening ot-training school, and to including Cruise's regulatergenerational male we emotional grandeur. To the movie's "Take My Breath inal. You feel like you're retual planes during shoot in visual effects took over where older and younge old and the new to coexist action and big, sweeping it worth the wait? The	rogant and g sequence ain its best lar Mission: epie, a dad at end, the Away," but eally in the ing. In that Hollywood. pilots butt t. He's also emotions. COVID and answer is a	223.0	0.85
Top Gun: ['Actio Maverick 'Drama' tt1745960 Top Gun: ['Actio Maverick 'Drama' Veri setinin istatistiki verileri	year but he You see I You see I performance beginning edge o "Iceman" who I', rw8177954 end.I could	can still do the job! He is a Maverick's innate cockinesse talks about his dead wings as well playing Maverick it is wonderful nostalgia. Of your seat the whole time is now an Admiral and dying scene is so sactantly dragged into the the	awesome as Maverick cases (something all fighter pagman's son and how he k's love interest and Goo Oh and I should also mere. To wrap it all up I wanteing of throat cancer. Val had but they have a joke a heater, thinking that the	apturing emotions and giving pilots have I'm sure) but you can't bear the thought of hir ose's son. Yes you 1980's must not the aieral sequences ared to mention the touching s Kilmer looks so bad because at the end where he asks Max by didn't need to make a Top or ges after walking out of the the	a performance that he halso see his vulnerability in dying as well. Jennifer sic lovers we get to hear the best and most integene where Maverick go he has had throat cance terick who the best pilot.	hasn't given since A Few y and his pain especially Connally and Miles Telle Kenny Loggin's Danger Sense in movie history you ses to see his old Top Gurer and can no longer species? Go see this movie by one was where that story	Good Men. the scenes give great Zone at the are on the classmate k well. The all means!	983.0 885.0	0.85
mean 35.699819 std 94.279334 min 0.000000 25% 4.000000 50% 11.000000 75% 29.000000	0.000000 93445.000000 0.702189 0.626121 0.201091 0.274538 0.000000 0.100000 0.580000 0.400000 0.710000 0.700000 0.840000 0.800000								
<pre>sid = SentimentIntensityAnalyzer(sentiments = [] for review in df['review']: sentiment_score = sid.polarit sentiments.append(sentiment_s sentiments_df = pd.DataFrame(sent df = pd.concat([df, sentiments_df</pre>	y_scores(review) core) iments)								
movie_title genres 0 0.801308 ['Action', 'Drama'] 1 0.801308 ['Action', 'Drama'] 2 0.801308 ['Action', 'Drama'] 3 0.801308 ['Action', 'Drama'] 4 0.801308 ['Action', 'Drama'] 5 0.801308 ['Action', 'Drama'] 6 0.801308 ['Action', 'Drama']	Wow. The first Top Gun is a If you were a late teen or In one of the more memora This is one sequel that log I was reluctantly dragged If there's any movie that de	classic, and as we in your early twent ole lines in the orig oked like it would n into the theater, th eserves to be seen	2736.0 5330.0 1223.0 983.0 1885.0 1071.0 658.0	0.881.00.0360.951.00.0510.851.00.0540.851.00.1270.901.00.1630.861.00.071	0.812 0.024 -0.72 0.742 0.187 0.59	944 211 829 922 227			
8 0.801308 ['Action', 'Drama']		erick is an awesom Cruise Post-Mid-19	641.0 399.0 406.0	0.75 1.0 0.022	0.709 0.269 0.99	708 976 970			
<pre>print(len(uni)) 4804 for movie_id in grouped_df['movie compound_value = grouped_df[g df.loc[df['movie_id'] == movi df.drop(columns="movie_id", inpla df.drop(columns="review_id", inpl df.dropna(subset=['genres'], inpl</pre>	_id']: rouped_df['movie_id'] == movi e_id, 'movie_title'] = compou ce=True) ace=True)		[0]						
 0.801308 ['Action', 'Drama'] 0.801308 ['Action', 'Drama'] 0.801308 ['Action', 'Drama'] 	Wow. The first Top Gun is a cl If you were a late teen or in In one of the more memorable This is one sequel that looke I was reluctantly dragged int	assic, and as we your early twent lines in the orig ed like it would n	d_helpful found_helpf 2736.0 5330.0 1223.0 983.0 1885.0	ful_rate rating neg 0.88 1.0 0.036 0. 0.95 1.0 0.051 0. 0.85 1.0 0.054 0. 0.85 1.0 0.127 0. 0.90 1.0 0.163 0.	834 0.115 0.9213 812 0.134 0.9829 637 0.237 0.9923	4 1 9 2			
<pre>uni = df["genres"].unique() unique_genres_set = set() for genres_list in uni: genres = genres_list.strip('[unique_genres_set.update(genr unique_genres_list = list(unique_ for genre in unique_genres_list: df[genre] = 0 # Her bir film için türleri kontr</pre>	es) genres_set)	. a							
<pre>for index, row in df.iterrows(): genres_str = row['genres'] if genres_str is not None: genres = genres_str.strip for genre in genres: df.at[index, genre] = import matplotlib.pyplot as plt import seaborn as sns numeric_df = df.select_dtypes(incorr_matrix = numeric_df.corr() plt.figure(figsize=(15, 10)) sns.heatmap(corr_matrix, annot=Tr</pre>	1 lude=['float64', 'int64', "in								
plt.title('Cross-Correlation Map' plt.show() Unnamed: 0 -1.00-0.200.330 movie_title -0.201.00 0.030 found_helpful -0.330.03 1.00 0 found_helpful_rate -0.120.120.021 rating -0.150.40 0.040 neg -0.05-0.400.000) .120.150.050.02-0.060.090.05 .12 <mark>0.40</mark> -0.400.00 <mark>0.320.46</mark> -0.21 .020.04-0.000.000.010.02-0.02	Cross-Correla 0.000.020.04-0.030.010.0 0.07 0.00-0.09 0.010.01-0.010.010.0 0.020.010.01-0.000.01-0.0 0.000.010.010.01	02 0.10-0.030.04-0.020.1 04 0.09-0.160.050.03-0.0 02-0.040.01-0.060.01-0.0 010.00-0.020.010.02 0.0 01 0.01-0.040.03 0.05 0.0	060.180.21 <mark>0.010.030.00-0.0</mark> 020.10-0.050.010.060.020.0 020.05-0.020.00-0.000.010.0 060.020.010.00-0.020.010.0	030.020.000.01 2-0.010.100.01 00.010.02-0.01 20.01-0.030.02 10.000.01-0.00	- 1.0 - 0.8			
pos -0.06 <mark>0.32</mark> 0.010 compound -0.09 <mark>0.46</mark> 0.02 0 'Horror' -0.05-0.21-0.02 0 'Family' -0.00 0.07-0.010 'Western' -0.02 0.00-0.010 'War' -0.04-0.090.000 'Animation' -0.030.11-0.010 'Music' -0.010.02-0.010	.07 <mark>0.27</mark> -0.400.68 <mark>1.00</mark> 0.59-0.06 .11 <mark>0.30</mark> -0.680.04 <mark>0.59</mark> 1.00-0.09 .020.080.10-0.020.060.09 <mark>1.00-0.030.010.010.030.010.010.030.030.010.01</mark>	0.01-0.010.040.050.010.0 0.030.00-0.040.050.010.0 0.030.020.040.060.010.0 1.00-0.010.020.020.000.0 0.011.00-0.010.020.000.0 0.020.011.00-0.030.010.0 0.020.020.031.00-0.010.0	010.040.060.010.000.0 020.040.070.020.01-0.0 030.050.060.150.040.1 02-0.020.060.080.020.0 010.05-0.050.070.000.0 020.010.040.100.010.1 020.030.090.100.020.1	040.04 0.13 0.00 0.01-0.010.0 030.08 0.10 0.01 0.01 0.00-0.0 180.120.040.020.040.05-0.0 050.07 0.00-0.010.030.010.0 050.030.040.010.020.020.0 10-0.070.060.000.040.030.0 02-0.020.000.000.010.010.0	020.010.020.01 010.010.000.01 040.010.10-0.05 020.000.040.03 1-0.000.040.07 000.000.050.04 030.010.050.02	- 0.6			
'Romance' -0.100.09-0.040 'Thriller' -0.030.160.010 'Crime' -0.04-0.050.060 'Biography' -0.020.030.010 'Drama' -0.15-0.060.020 'Adventure' -0.200.180.100 'Comedy' -0.010.21-0.050 'Musical' -0.060.01-0.010	.000.01-0.040.000.040.040.05 .020.040.060.01-0.060.070.06 .010.030.010.00-0.01-0.020.15 .020.05-0.020.02-0.000.01-0.04 .020.06-0.000.04-0.040.030.18 .050.02-0.070.020.040.08-0.12 .020.01-0.060.08	0.020.05-0.010.030.010.0 0.060.050.040.090.020.0 0.080.070.100.100.010.0 0.020.000.010.020.010.0 0.050.050.10-0.110.020.0 0.07-0.030.070.18-0.020.0 0.00-0.040.060.070.000.0	02 <mark>1.00</mark> -0.080.100.020.0 040.08 <mark>1.00</mark> -0.030.070.1 060.100.03 <mark>1.00</mark> 0.00 0.1 020.020.070.00 <mark>1.00</mark> 0.0 050.03-0.170.100.05 <mark>1.0 070.060.230.330.050.2 030.00-0.230.00-0.060.2</mark>	03-0.060.00 0.05-0.040.010.0 170.230.230.030.110.01-0.0 10-0.330.00-0.020.150.030.1 05-0.050.060.01-0.040.030.0 00-0.270.290.01-0.130.060.1 27 <mark>1.00</mark> 0.00-0.030.12-0.070.0 290.00 1.00-0.010.070.080.0	030.010.070.06 080.020.050.04 000.03-0.210.02 010.000.050.03 5 0.02-0.21-0.11 050.020.050.03 080.020.160.03	- 0.4 - 0.2			
'Mystery' -0.040.000.020 'History' -0.00-0.030.020 'Film-Noir' -0.03-0.020.010 'Sci-Fi' -0.150.000.100 'Action' -0.010.01-0	.01-0.01-0.010.010.010.010.010.04 .01-0.010.010.010.010.010.010.04 .010.010.000.01-0.010.010.010.010.010.01	0.010.020.030.030.010.0 0.020.01-0.000.030.010.0 0.000.000.000.010.000.0 0.040.040.050.050.010.0	020.010.01-0.030.030.0 020.030.080.100.010.1 000.010.020.03-0.000.0 030.070.050.210.050.2	060.070.080.010.05 <mark>1.00</mark> -0.0 15-0.050.080.010.040.03 <mark>1.0 02-0.020.020.000.010.010.0 210.05-0.160.020.070.010.0 110.030.03-0.020.010.110.0</mark>	030.010.010.11 0-0.000.060.04 001.00-0.010.08 060.01 <mark>1.00</mark> 0.00 040.080.00 <mark>1.00</mark>	- 0.0			
<pre># df.drop(columns="genres", inpla # df.drop(columns="review", inpla # df.to_csv("imdb_prepared_data.columns import pandas as pd data = pd.read_csv("imdb_prepared df = data.copy() def multiply_by_10(x): return x * 10</pre>	ce=True) sv")								
<pre>df['rating'] = df['rating'].apply from sklearn.linear_model import from sklearn.metrics import r2_sc from sklearn.model_selection impo from sklearn.ensemble import Rand from sklearn.ensemble import Grad df = df.dropna() X, y = df.drop(columns=["rating", X_train, X_test, y_train, y_test</pre>	LinearRegression ore rt train_test_split omForestRegressor ientBoostingRegressor "found_helpful", "compound"] = train_test_split(X, y, test		47)						
<pre>print("Linear Regression:") model_linear = LinearRegression() model_linear.fit(X_train, y_train y_pred_train_linear = model_linear y_pred_test_linear = model_linear r2_train_linear = r2_score(y_train) r2_test_linear = r2_score(y_test, print("Training R^2 Score:", r2_test_linear)</pre>	r.predict(X_train) .predict(X_test) n, y_pred_train_linear) y_pred_test_linear)								
<pre>print("Testing R^2 Score:", r2_te print("</pre>	<pre>st_linear)") tRegressor(n_estimators=100, y_train) rest.predict(X_train) est.predict(X_test) _pred_train_rf)</pre>	nin_samples_split=2)							
<pre>print("Training R^2 Score:", r2_t print("Testing R^2 Score:", r2_te print("</pre>	st_rf)	les_split=2)							
<pre>model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict</pre>	(X_test)								
<pre>model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y) r2_test_gb = r2_score(y_test, y_p) print("Training R^2 Score:", r2_test_resting R^2 Score:", r2_test_resting R^2 Score:", r2_test_resting R^2 Score:", r2_test_resting R^2 Score: 0.226585586259</pre> Testing R^2 Score: 0.2366357352633	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)")								
<pre>model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y) r2_test_gb = r2_score(y_test, y_p) print("Training R^2 Score:", r2_teprint("Testing R^2 Score:", r2_teprint("</pre>	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)")								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
<pre>model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p) print("Training R^2 Score:", r2_test_resting R^2 Score:", r2_test_resting R^2 Score:", r2_test_resting R^2 Score: ", r2_test_resting R^2 Score: ", r2_test_resting R^2 Score: 0.2265855862590 Testing R^2 Score: 0.23663573526330 Random Forest: Training R^2 Score: 0.9028743539400 Testing R^2 Score: 0.32099366276244 Gradient Boosting:</pre>	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y) r2_test_gb = r2_score(y_test, y_p) print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y) r2_test_gb = r2_score(y_test, y_p) print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y) r2_test_gb = r2_score(y_test, y_p) print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y r2_test_gb = r2_score(y_test, y_p print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								
model_gb.fit(X_train, y_train) y_pred_train_gb = model_gb.predict y_pred_test_gb = model_gb.predict r2_train_gb = r2_score(y_train, y) r2_test_gb = r2_score(y_test, y_p) print("Training R^2 Score:", r2_te print("Testing R^2 Score:", r2_te print("	t(X_train) (X_test) _pred_train_gb) red_test_gb) rain_gb) st_gb)") 99941 8603 8336 827								