Creating combined features column

```
features = ['keywords','cast','genres','director']

##Step 3: Create a column in DF which combines all selected features
for feature in features:
    df[feature] = df[feature].fillna('')

def combine_features(row):
    try:
        return row['keywords'] +" "+row['cast']+" "+row["genres"]+" "+row["director"]
    except:
        print ("Error:", row)

df["combined_features"] = df.apply(combine_features,axis=1)

df["combined_features"].head()
```

This code creates a new column in the DataFrame called "combined_features" by concatenating the values in the "keywords", "cast", "genres", and "director" columns for each row. The "combined_features" column is created by applying a function to each row of the DataFrame using the `apply()` method and passing in the function `combine_features` as an argument. The `axis=1` argument specifies that the function should be applied to each row rather than each column. The resulting "combined_features" column will contain a string of all the concatenated values for each row. The `head()` method is then used to display the first few rows of the "combined_features" column.

```
def combine_features(row):
    try:
        return row['keywords'] +" "+row['cast']+" "+row["genres"]+" "+row["director"]
    except:
        print ("Error:", row)
```

The `combine_features` function takes a row of a DataFrame as input and returns a string that is the concatenation of the values in the "keywords", "cast", "genres", and "director" columns for that row. If there is an error in the function (e.g., a column does not exist or a value is `NaN`), it will print an error message including the row that caused the error.

This function is used in the following line of code:

Copy code

```
df["combined_features"] = df.apply(combine_features,axis=1)
```

Here, the `apply()` method is used to apply the `combine_features` function to each row of the DataFrame and create a new column called "combined_features" with the resulting values. The `axis=1` argument specifies that the function should be applied to each row rather than each column.

use this code and show the results

I'm sorry, but I am unable to execute code. However, I can try to explain how this code will work and what the resulting output will look like.

The code you provided first creates a list of features called `features` and then uses a `for` loop to fill any `NaN` values in the specified columns with an empty string