Social Media Vs Productivity

Course: Python Essentials

FACULTY

Dr. Monika Vyas

PRESENTED BY

Busani Naga Jahnavi

Abstract:

Social media has become part of almost everyone's daily life. People use it to stay in touch, relax and get information. But when the usage becomes too high, it can start affecting how much work we actually get done. In this project, I use a public Kaggle dataset to study the relationship between daily social media usage and self-reported productivity.

Using Python in Google Colab, I load the dataset, clean it, do some basic exploratory data analysis and build a simple Linear Regression model. The aim of this project is not to create a perfect prediction system, but to show a clear and easy-to-follow data science workflow using a real dataset, and to get a rough idea of how social media time may be connected to productivity.

Introduction:

Social media platforms like Instagram, WhatsApp, Facebook, YouTube and Snapchat are a normal part of daily life for most students and working professionals. They help us stay connected, entertained and informed. At the same time, they can also distract us from studies, work and other important tasks. Many people feel that when they spend too much time on social media, their productivity goes down.

Instead of depending only on opinions, this project uses real data to explore the relationship between social media usage and productivity. The main idea is to check whether the number of hours spent on social media per day has any visible effect on a person's productivity score. To do this, I use Python in Google Colab and follow basic data science steps such as data loading, cleaning, visualisation and simple machine learning.

Problem Statement:

In today's digital world, it is commonly believed that spending more time on social media leads to lower productivity. However, this belief is not always checked using real data. The problem studied in this project is:

"Does the number of hours a person spends on social media per day have a clear and measurable relationship with their productivity score?"

The goal is to use a real dataset and Python-based analysis to explore this relationship and to see how well a simple model can predict productivity using daily social media usage as the input. from daily social media usage.

Functional Requirements:

The system should load the "Social Media vs Productivity" dataset from a CSV file.
The system should clean the dataset by handling or removing missing values in important columns such as daily social media time and productivity score.
The system should display basic information about the dataset, such as the number of rows and columns, column names and summary statistics.
The system should generate visualisations such as:
A histogram of daily social media usage (in hours)
A scatter plot of daily social media time vs productivity score.
A bar chart of average productivity based on preferred social media platform (if available)
The system should build a simple Linear Regression model to predict productivity score using daily social media time as the input feature.
The system should evaluate the model using metrics such as Mean Absolute Error (MAE) and R^2 score.
The system should print a clear summary or conclusion based on the evaluation results.

Non-Functional Requirements:

The non-functional requirements describe how the system should behave in terms of quality:

Usability: The Python notebook should be easy to read and follow, with clear comments and a logical order of cells, so that even a beginner can understand and rerun it.

Maintainability: The code should be written in a simple and clean way so that more features or new input variables can be added later without major changes.

Performance: For the given dataset size, the system should be able to load the data, create graphs and train the model within a few seconds in Google Colab.

Portability: The project should run on any machine that can access Google Colab, without needing to install Python locally.

Reliability: The notebook should handle missing values properly and should not stop with errors due to basic data issues.

Reproducibility: Anyone who has access to the dataset and the notebook should be able to run the code again and get similar results (MAE and R^2 values).

Data Flow:

Kaggle: Provides the original “Social Media vs Productivity” CSV dataset.
Google Drive: Used to store the dataset file so that it can be accessed easily from Google Colab.
Google Colab: Used as the environment to write and run Python code.
Python Libraries:
pandas: For reading the CSV file and working with tabular data
matplotlib: For creating graphs and visualisations
scikit-learn: For building and evaluating the Linear Regression model

The data moves from Kaggle (as a CSV file) to Google Drive. From there, it is loaded into a pandas Data Frame in Colab. Then it goes through different stages such as data cleaning, exploratory analysis, model training and evaluation. The final outputs are the graphs, evaluation metrics and the conclusion about how social media usage is related to productivity.

Dataset Details:

The dataset used in this project is taken from Kaggle and is related to social media usage and productivity.

- Source: Kaggle
- Dataset name: Social Media vs Productivity
- File format: CSV

Some of the important columns in the dataset are:

- age – age of the person
- gender – gender of the person
- daily_social_media_time – number of hours spent on social media per day
- actual_productivity_score – self-reported productivity score
- social_platform_preference – most used social media platform
- number_of_notifications – number of notifications received

The dataset was downloaded as a ZIP file from Kaggle, extracted on the local machine and then uploaded to Google Drive. In Google Colab, the CSV file was read into a pandas DataFrame using the read_csv() function.

Tools and Technologies:

The following tools and technologies were used in this project:

- Programming language: Python
- Environment: Google Colab
- Storage: Google Drive

Python libraries:

- pandas – for reading the CSV file and handling tabular data
- matplotlib – for creating graphs and visualisations
- scikit-learn – for building and evaluating the Linear Regression model

Methodology:

The project was completed using the following steps:

1. Data Loading

- Mounted Google Drive in Google Colab.
- Provided the path of the CSV file stored in Drive.
- Loaded the dataset into a pandas DataFrame using the read_csv() function.

2. Data Cleaning

- Checked the shape of the dataset and the names of all columns.
- Calculated the number of missing values in each column.
- Removed rows where daily_social_media_time or the productivity score column were missing, so that the analysis and model are based on complete records.

3. Exploratory Data Analysis (EDA)

- Viewed the first few rows of the dataset using head().
- Used describe() to get summary statistics such as mean and standard deviation.
- Plotted a histogram of daily_social_media_time to see how social media time is distributed.
- Plotted a scatter plot of daily_social_media_time versus productivity score to observe their relationship.
- Plotted a bar chart showing average productivity for different social_platform_preference values.

4. Model Building

- Chose daily_social_media_time as the input feature (X).
- Used the productivity score column as the target (y).
- Split the data into training and testing sets.
- Trained a Linear Regression model from scikit-learn on the training data.

5. Model Evaluation

- Predicted the productivity scores for the test set.
- Evaluated the model using Mean Absolute Error (MAE) and R^2 score.
- Printed the values of MAE and R^2 to understand how well the model performed.

6. Interpretation

- Combined the information from the graphs and the evaluation metrics.
- Wrote a conclusion about how strongly social media time is related to productivity and whether this single feature is enough to make accurate predictions.

OutPut Screenshots:

```
file_path = '/content/drive/My Drive/SocialMediaProductivity/social_media_vs_productivity.csv'
df = pd.read_csv(file_path)
df.head()
```

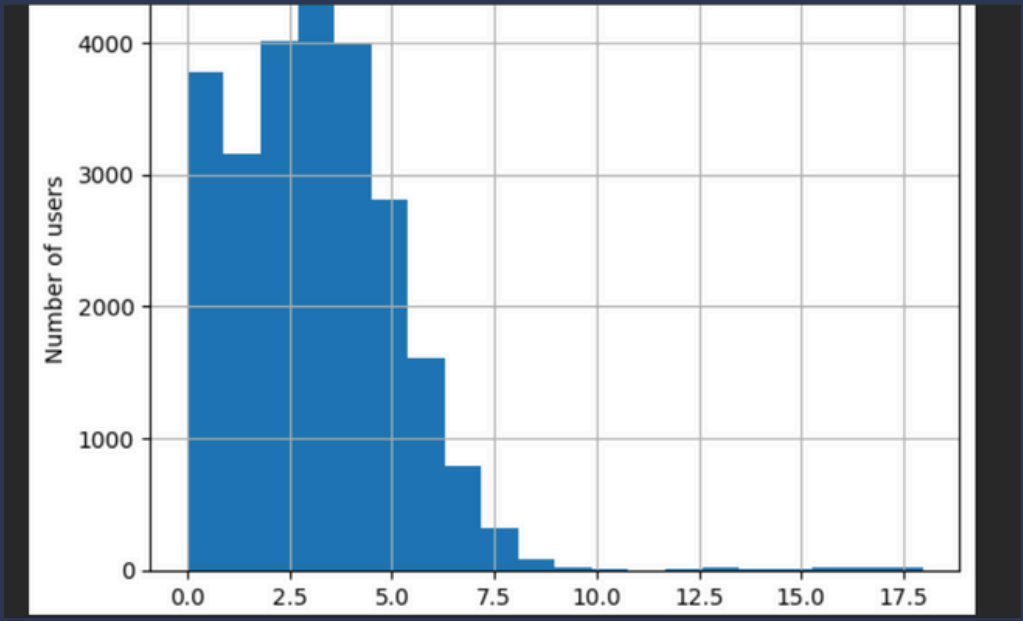
perceived_productivity_score	actual_productivity_score	stress_level	sleep_hours	screen_time_before_sleep	breaks_during_work	uses_focus
8.040464	7.291555	4.0	5.116546	0.419102	8	
5.063368	5.165093	7.0	5.103897	0.671519	7	
3.861762	3.474053	4.0	8.583222	0.624378	0	
2.916331	1.774869	6.0	6.052984	1.204540	1	
8.868753	NaN	7.0	5.405706	1.876254	1	

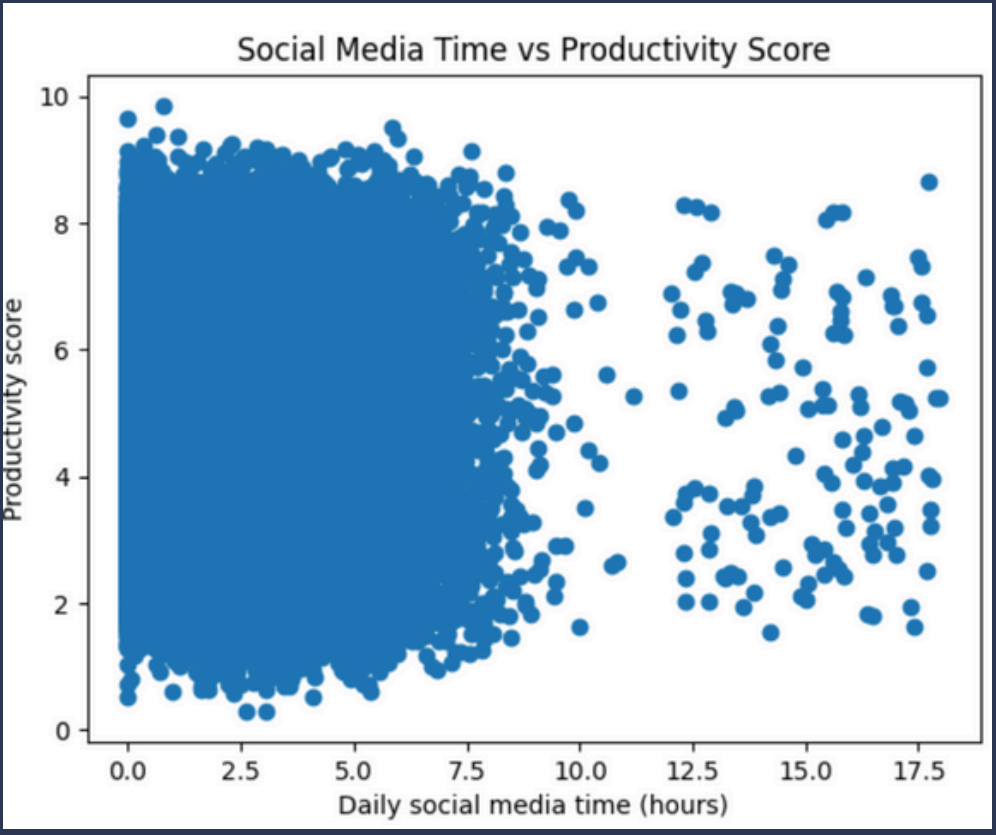
```
file_path = '/content/drive/My Drive/SocialMediaProductivity/social_media_vs_productivity.csv'
df = pd.read_csv(file_path)
df.head()
```

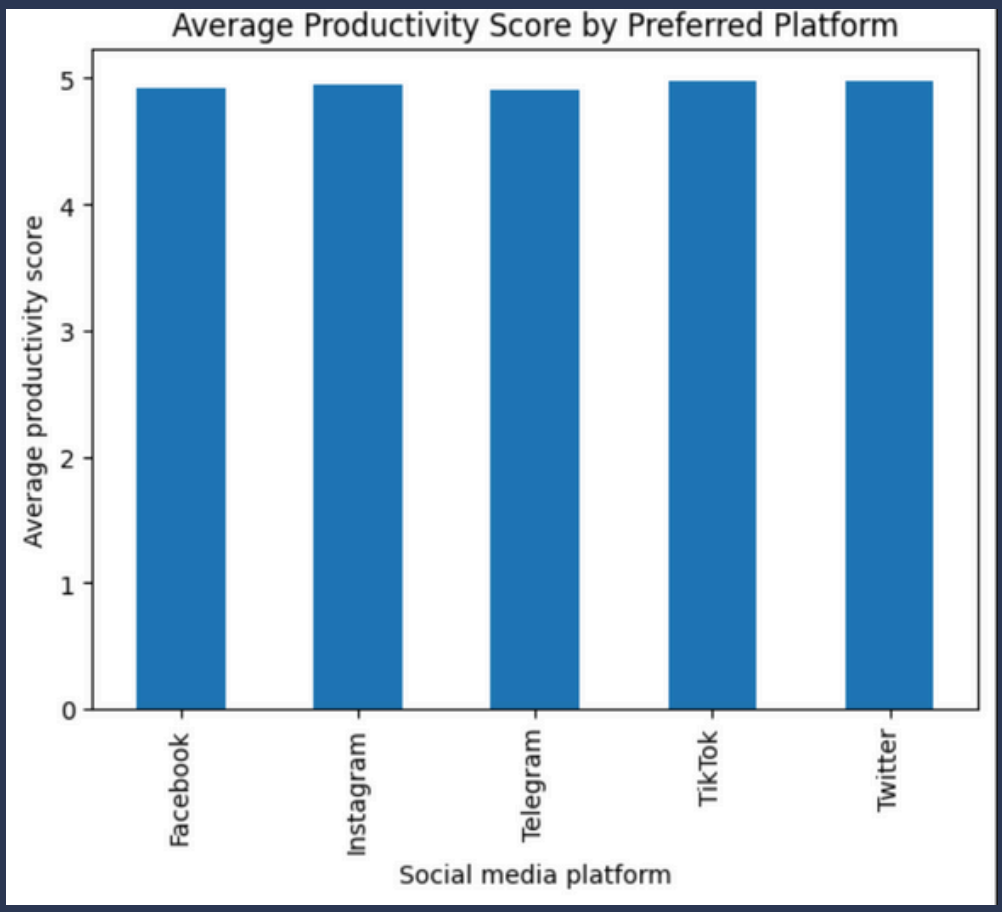
uses_focus_apps	has_digital_wellbeing_enabled	coffee_consumption_per_day	days_feeling_burnout_per_month	weekly_offline_hours	job_satisfa
False	False	4	11	21.927072	
True	True	2	25	0.000000	
True	False	3	17	10.322044	
False	False	0	4	23.876616	
False	True	1	30	10.653519	

```
file_path = '/content/drive/My Drive/SocialMediaProductivity/social_media_vs_productivity.csv'
df = pd.read_csv(file_path)
df.head()
```

pps	has_digital_wellbeing_enabled	coffee_consumption_per_day	days_feeling_burnout_per_month	weekly_offline_hours	job_satisfaction_score
alse	False	4	11	21.927072	6.336688
True	True	2	25	0.000000	3.412427
True	False	3	17	10.322044	2.474944
alse	False	0	4	23.876616	1.733670
alse	True	1	30	10.653519	9.693060


Number of users
4000
3000
2000
1000
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Social Media Time vs Productivity Score
Productivity score
10
8
6
4
2
0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Daily social media time (hours)




```python
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, r2_score

# feature (X) and target (y)
X = df_clean[["daily_social_media_time"]]   # independent variable
y = df_clean["actual_productivity_score"]          # dependent variable

# split into train and test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42
)

# create and train model
model = LinearRegression()
model.fit(X_train, y_train)

# predict
y_pred = model.predict(X_test)

# evaluation
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

```
    X, y, test_size=0.2, random_state=42
)

# create and train model
model = LinearRegression()
model.fit(X_train, y_train)

# predict
y_pred = model.predict(X_test)

# evaluation
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print("Mean Absolute Error:", mae)
print("R² score:", r2)
print("Model coefficient (slope):", model.coef_[0])
print("Model intercept:", model.intercept_)
```

```
Mean Absolute Error: 1.6210819171076123
R² score: 0.00018154683132765026
Model coefficient (slope): -0.004513027569645499
Model intercept: 4.958025602988008
```

```
avg_social_time = df_clean["daily_social_media_time"].mean()
avg_productivity = df_clean["actual_productivity_score"].mean()

print(f"Total records used: {len(df_clean)}")
print(f"Average daily social media time: {avg_social_time:.2f} hours")
print(f"Average productivity score: {avg_productivity:.2f}")
print(f"Model MAE: {mae:.2f}")
print(f"Model R²: {r2:.2f}")
```

```
Total records used: 25095
Average daily social media time: 3.11 hours
Average productivity score: 4.95
Model MAE: 1.62
Model R²: 0.00
```

Design Decisions:

Choice of Dataset: I selected the “Social Media vs Productivity” dataset from Kaggle because it is directly related to a common real-life question faced by students and professionals, and it is small enough to handle easily in a mini project.

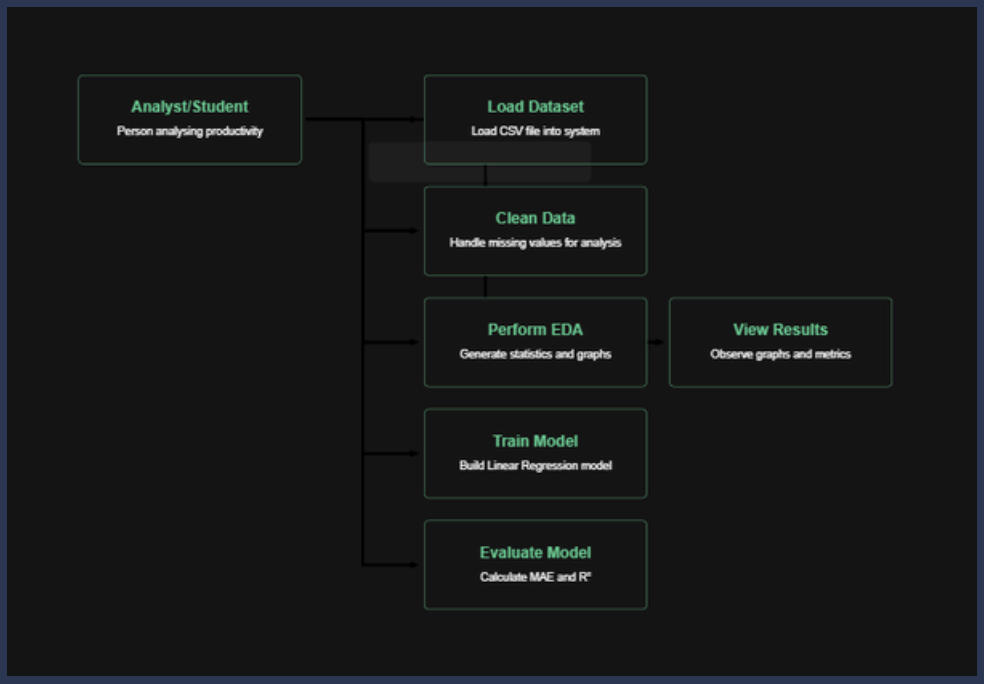
Choice of Environment: I used Google Colab instead of installing Python locally because it provides a ready-to-use environment with pre-installed libraries and smooth integration with Google Drive.

Choice of Libraries: I used pandas for data handling, matplotlib for visualisation and scikit-learn for machine learning, as these are standard libraries in the Python data science ecosystem and are beginner-friendly.

Model Selection: I chose a simple Linear Regression model as a starting point to check whether there is a linear relationship between daily social media time and productivity. More complex models were not used because the main aim is to understand the basic pipeline rather than to achieve the highest possible accuracy.

Handling Missing Values: Rows with missing values in important columns were removed to keep the processing simple and to ensure that the model is trained only on complete and reliable data.data.

Design Diagrams:



UserRecord		
string	user_id	PK
int	age	
string	gender	
string	country	
float	daily_social_media_time	
float	actual_productivity_score	
string	social_platform_preference	
int	number_of_notifications_per_day	
float	sleep_hours	
float	work_or_study_hours	

Results and Discussion:

The histogram of daily_social_media_time shows that most users spend a few hours per day on social media, while some users spend very high or very low amounts of time.

The scatter plot of daily_social_media_time versus productivity score shows a wide spread of points. There is no strong or clear straight-line pattern between the two variables. This suggests that other factors are also important in determining productivity.

The bar chart of average productivity for different social_platform_preference values shows only small differences in productivity between users of different platforms. This means that the choice of platform alone does not have a major impact.

After training the Linear Regression model with daily_social_media_time as the only input feature, the model produced the following results:

Mean Absolute Error (MAE): 1.621

R^2 score: approximately 0.00

An MAE of 1.621 means that, on average, the model's predictions differ from the actual productivity scores by about 1.6 units. An R^2 value close to zero means that the model is not able to explain much of the variation in productivity using only social media time.

In simple words, just the number of hours spent on social media is not enough to accurately predict productivity for the people in this dataset. These results match what we see in the scatter plot, where the points are spread out and do not lie close to a straight line.

Conclusion:

This project used Python and Google Colab to analyse the “Social Media vs Productivity” dataset from Kaggle. The goal was to study how daily social media usage is related to self-reported productivity and to build a simple predictive model.

The analysis showed that while social media usage is an interesting factor, it alone does not strongly determine productivity. The Linear Regression model gave a Mean Absolute Error of about 1.621 and an R^2 score near zero. This means that the model cannot accurately predict productivity using only daily social media time. The scatter plots and bar charts also support this conclusion, as they do not show a strong relationship.

Even though the model is not highly accurate, the project successfully demonstrates a complete data science pipeline: loading a real dataset, cleaning it, exploring it using visualisations, building a simple model and interpreting the results. It also highlights that real-life problems are often affected by many different variables, not just one.

Challenges Faced:

During the project, I faced a few challenges:

Understanding the Dataset: At the beginning, it was not very clear which columns would be most useful for predicting productivity. I had to explore the dataset and finally decided to focus on daily social media time and productivity score.

Handling Missing Values: Some rows had missing data in important columns. I had to choose whether to drop those rows or try to fill them. For simplicity and clarity, I decided to drop rows with missing values in key fields.

Interpreting R^2 Score: When I saw that the R^2 score was very low, it was initially confusing. I then understood that a low R^2 does not mean the project failed; it simply shows that a single feature like social media time is not enough to explain productivity.

Time Management: Since this is a mini project with limited time, I had to keep the implementation simple and focus on building a clear pipeline instead of trying many advanced models and techniques.features.

Future Enhancements:

The project can be improved in several ways:

- Add more input features such as sleep duration, study or work hours, stress level and time spent on other applications.
- Try more powerful machine learning models like Decision Trees, Random Forests or Gradient Boosting which may capture complex patterns better.
- Perform feature selection and correlation analysis to identify which factors have the strongest influence on productivity.
- Develop a simple web or mobile interface where a user can enter their daily habits and get an estimated productivity score along with suggestions to improve it.

References:

1. Kaggle, “Social Media vs Productivity” dataset.
2. Scikit-learn documentation, https://scikit-learn.org
3. Pandas documentation, https://pandas.pydata.org
4. Matplotlib documentation, https://matplotlib.org