To,
IITD-AIA Foundation of Smart Manufacturing

Subject: **Weekly Progress Report for Week 1**

Dear Sir,
Following is the weekly progress report dating from 5th to 11th of June, 2023. I went ahead and researched on the equipment failure prediction and predictive maintenance.

My Understanding of the Project: INTP23-ML-5: Equipment Failure Prediction for Predictive Maintenance

Predictive Maintenance is the procedure of using already existing data of various factors which might cause equipment failure and using those data available to us to predict when an equipment might fail in the future. It basically works on the principle of Condition Monitoring. Condition-monitoring tools combined with artificial intelligence and machine learning techniques forecast expected machine failure.

Predictive maintenance helps in:

- reducing maintenance costs
- maintenance scheduling and planning
- improving reliability.

With the help of such technologies, we can predict and perform maintenance activities without disrupting normal machine activities,

Weekly Progress:

5th June 2023:
Went on to Kaggle to get various resources on computer vision.

- Went through the resources mentioned in this Kaggle discussion tab: https://www.kaggle.com/discussions/general/201931
- Explored a course on YouTube called Computer Vision by Michigan University.

6th June 2023:
Researched on Tensorflow.

- Went through the basic functions of Tensorflow
- Went through Tensorflow via the following website: https://rubikscode.net/2021/08/03/introduction-to-tensorflow-with-python-example/

7th June 2023:
Went through the project assigned:
INTP23-ML-5: Equipment Failure Prediction for Predictive Maintenance.

- Went through all of the listed tools in the problem file, researched more about Flask and Scikit-Learn due to lack of adequate knowledge.
- Installed Flask and went through its docs.
- Installed Scikit-Learn and went through its docs.

8th June 2023:
Went through the NASA Turbofan Engine Degradation Simulation Dataset

- I was having a bit of a hard time understanding all the terminologies used. I have been working on understanding what the dataset contained.
- Link to the database: https://data.nasa.gov/Aerospace/CMAPSS-Jet-Engine-Simulated-Data/ff5v-kuh6
- Tried to gather an understanding of the dataset via blogs and research papers. https://medium.com/@rohit.malhotra67/predictive-maintenance-on-nasas-turbofan-engine-degradation-dataset-cmapss-c066ee427931
- Went through all the Tools provided to us and focused mainly on practicing Numpy, Pandas in order to get a better understanding of them.

9th June 2023:
Practiced on the Mcdonalds Nutrition Dataset

- Learned the various Exploratory Data Analytics functions.
- Learned the various terminologies provided in the NASA Engine Degradation Dataset.
- I went ahead and practiced on Mcdonalds Nutrition dataset from Kaggle.
- File Practice: https://www.fsmskills.in/draftfile.php/24378/user/draft/962974832/9th%20June%20Practice.ipynb

10th June 2023:
Practiced on Company Bankruptcy Dataset

- Learned Flask since I think my flask knowledge is not up to the mark.
- Practiced on Company Bankruptcy Dataset via kaggle.
- Practice file: https://www.fsmskills.in/pluginfile.php/26645/mod_diary/entry/8286/10th%20June%20Practice.ipynb

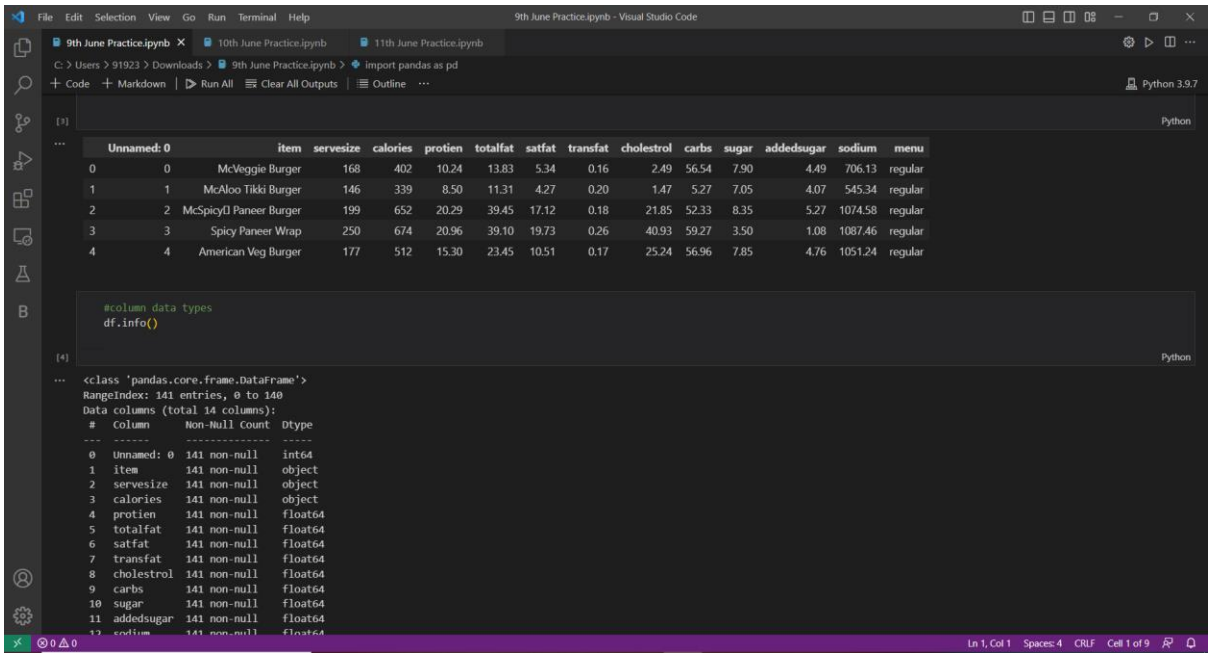
11th June 2023:
Practiced on Salary Prediction Dataset

- I went ahead and practiced on Salary Prediction Dataset to improve my skills.
- I utilized the use of various functionalities of matplotlib and sk.learn.
- Practice file: https://www.fsmskills.in/pluginfile.php/26645/mod_diary/entry/8372/11th%20June%20Practice.ipynb

Installation of all modules:

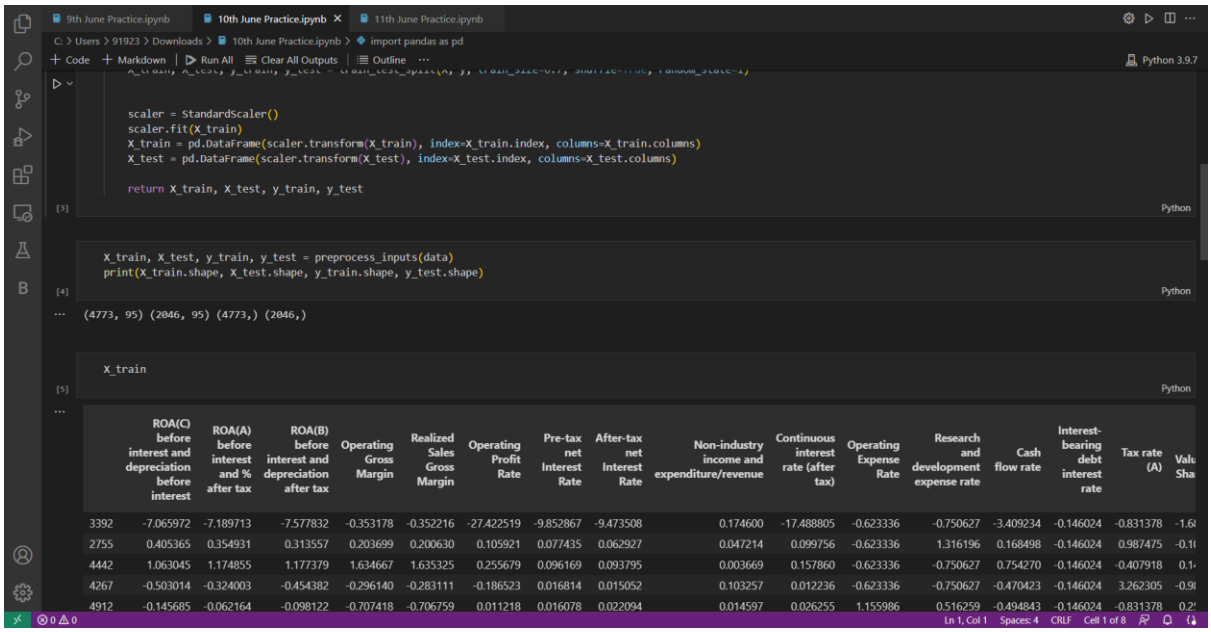
```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\91923>virtualenv flask
created virtual environment CPython3.9.7.final.0-64 in 2700ms
  creator CPython3Windows(dest=C:\Users\91923\flask, clear=False, no_vcs_ignore=False, global=False)
  seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=C:\Users\91923\AppData\Local\pypa\virtualenv)
    added seed packages: pip==21.2.4, setuptools==58.1.0, wheel==0.37.0
  activators BashActivator,BatchActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator

C:\Users\91923>cd flask

C:\Users\91923\flask>source bin/activate
'source' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\91923\flask>bin/activate
'bin' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\91923\flask>Scripts\activate

(flask) C:\Users\91923\flask>pip install flask
Collecting flask
  Using cached Flask-2.3.2-py3-none-any.whl (96 kB)
Collecting blinker>=1.6.2
  Using cached blinker-1.6.2-py3-none-any.whl (13 kB)
Collecting Werkzeug>=2.3.3
  Using cached Werkzeug-2.3.5-py3-none-any.whl (242 kB)
Collecting importlib-metadata>=3.6.0
  Using cached importlib_metadata-6.6.0-py3-none-any.whl (22 kB)
Collecting click>=8.1.3
  Using cached click-8.1.3-py3-none-any.whl (96 kB)
Collecting Jinja2>=3.1.2
  Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
Collecting itsdangerous>=2.1.2
  Using cached itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting colorama
  Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Collecting zipp>=0.5
  Using cached zipp-3.15.0-py3-none-any.whl (6.8 kB)
Collecting MarkupSafe>=2.0
  Using cached MarkupSafe-2.1.3-cp39-cp39-win_amd64.whl (17 kB)
Installing collected packages: zipp, MarkupSafe, colorama, Werkzeug, Jinja2, itsdangerous, importlib-metadata, click, blinker, flask
Successfully installed Jinja2-3.1.2 MarkupSafe-2.1.3 Werkzeug-2.3.5 blinker-1.6.2 click-8.1.3 colorama-0.4.6 flask-2.3.2 importlib-metadata-6.6.0 itsdangerous-2.1.2 zipp-3.15.0
WARNING: You are using pip version 21.2.4; however, version 23.1.2 is available.
You should consider upgrading via the 'C:\Users\91923\flask\Scripts\python.exe -m pip install --upgrade pip' command.
```

9th June 2023:



	Unnamed: 0	item	servesize	calories	protien	totalfat	satfat	transfat	cholestrol	carbs	sugar	addedsugar	sodium	menu
0	0	McVeggie Burger	168	402	10.24	13.83	5.34	0.16	2.49	56.54	7.90	4.49	706.13	regular
1	1	McAloo Tikki Burger	146	339	8.50	11.31	4.27	0.20	1.47	5.27	7.05	4.07	545.34	regular
2	2	McSpicy Paneer Burger	199	652	20.29	39.45	17.12	0.18	21.85	52.33	8.35	5.27	1074.58	regular
3	3	Spicy Paneer Wrap	250	674	20.96	39.10	19.73	0.26	40.93	59.27	3.50	1.08	1087.46	regular
4	4	American Veg Burger	177	512	15.30	23.45	10.51	0.17	25.24	56.96	7.85	4.76	1051.24	regular

10th June 2023



```
scaler = StandardScaler()
scaler.fit(X_train)
X_train = pd.DataFrame(scaler.transform(X_train), index=X_train.index, columns=X_train.columns)
X_test = pd.DataFrame(scaler.transform(X_test), index=X_test.index, columns=X_test.columns)

return X_train, X_test, y_train, y_test
```

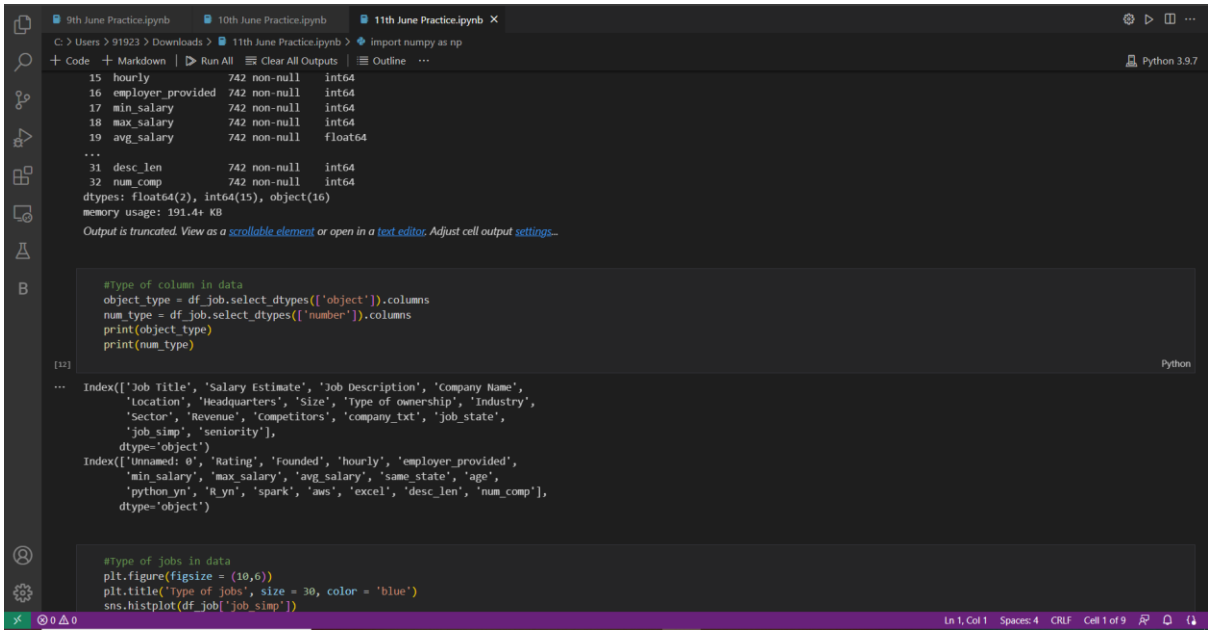
```
X_train, X_test, y_train, y_test = preprocess_inputs(data)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(4773, 95) (2046, 95) (4773,) (2046,)
```

```
X_train
```

	ROA(C) before interest and depreciation before interest	ROA(A) before interest and % after tax	ROA(B) before interest and depreciation after tax	Operating Gross Margin	Realized Sales Gross Margin	Operating Profit Rate	Pre-tax net Interest Rate	After-tax net Interest Rate	Non-industry income and expenditure/revenue	Continuous interest rate (after tax)	Operating Expense Rate	Research and development expense rate	Cash flow rate	Interest-bearing debt interest rate	Tax rate (A)	Valu Sha
3392	-7.065972	-7.189713	-7.577832	-0.353178	-0.352216	-27.422519	-9.852867	-9.473508	0.174600	-17.488805	-0.623336	-0.750627	-3.409234	-0.146024	-0.831378	-1.6
2755	0.405365	0.354931	0.313557	0.203699	0.200630	0.105921	0.077435	0.062927	0.047214	0.099756	-0.623336	1.316196	0.168498	-0.146024	0.987475	-0.1
4442	1.063045	1.174855	1.177379	1.634667	1.635325	0.255679	0.096169	0.093795	0.003669	0.157860	-0.623336	-0.750627	0.754270	-0.146024	-0.407918	0.1
4267	-0.503014	-0.324003	-0.454382	-0.296140	-0.283111	-0.186523	0.016814	0.015052	0.103257	0.012236	-0.623336	-0.750627	-0.470423	-0.146024	3.262305	-0.9
4912	-0.145685	-0.062164	-0.098122	-0.707418	-0.706759	0.011218	0.016078	0.022094	0.014597	0.026255	1.155986	0.516259	-0.494843	-0.146024	-0.831378	0.2

11th June 2023



```
 15  hourly             742 non-null    int64
 16  employer_provided  742 non-null    int64
 17  min_salary         742 non-null    int64
 18  max_salary         742 non-null    int64
 19  avg_salary         742 non-null    float64
...
 31  desc_len           742 non-null    int64
 32  num_comp           742 non-null    int64
dtypes: float64(2), int64(15), object(16)
memory usage: 191.4+ KB
```

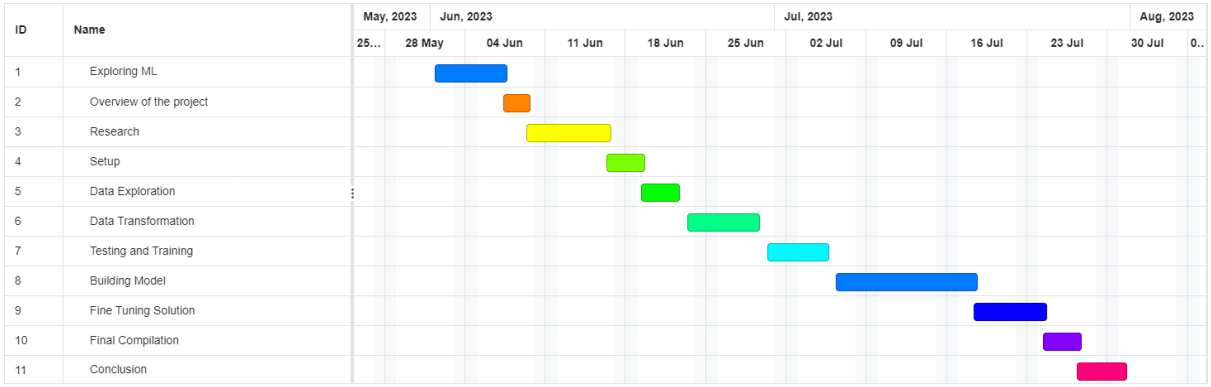
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```
#Type of column in data
object_type = df_job.select_dtypes(['object']).columns
num_type = df_job.select_dtypes(['number']).columns
print(object_type)
print(num_type)
```

```
Index(['Job Title', 'Salary Estimate', 'Job Description', 'Company Name',
       'Location', 'Headquarters', 'Size', 'Type of ownership', 'Industry',
       'Sector', 'Revenue', 'Competitors', 'company_txt', 'job_state',
       'job_simp', 'seniority'],
      dtype='object')
Index(['Unnamed: 0', 'Rating', 'Founded', 'hourly', 'employer_provided',
       'min_salary', 'max_salary', 'avg_salary', 'same_state', 'age',
       'python_yn', 'R_yn', 'spark', 'aws', 'excel', 'desc_len', 'num_comp'],
      dtype='object')
```

```
#Type of jobs in data
plt.figure(figsize = (10,6))
plt.title('Type of jobs', size = 30, color = 'blue')
sns.histplot(df_job['job_simp'])
```

Gantt Chart



ID	Name
1	Exploring ML
2	Overview of the project
3	Research
4	Setup
5	Data Exploration
6	Data Transformation
7	Testing and Training
8	Building Model
9	Fine Tuning Solution
10	Final Compilation
11	Conclusion