University of Bradford

Faculty of Engineering and Informatics

Academic Year (2024-2025)

COS7004- MSc Dissertation

Prediction of Mars Surface Temperature Depending on Different Altitude using Machine Learning Algorithms Associate with Feature Selection on Emirates Mars InfraRed Spectrometer (EMIRS) Dataset from Emirates Mars Mission (EMM)

Submitted by:

Htoo Thit Sinn – UB22070584

Abstract

As the improvement of the space technology, the space idealists, scientists and space researchers are beginning to explore more about the neighbour planets within the solar system. Among those planets, Mars or also known as the red planet, is the most popular planet due to the similarity of the weather information and other data collected by the previous researchers. Emirates Mars Mission (EMM) had launched a Mars obiter name ‘Hope’ in 2020. This machine has three instruments to observe the Mars climate activities and of all these three instruments, the data collected by the Emirates Mars InfraRed Spectrometer (EMIRS) instrument will be applied in this research to predict the Mars surface temperature to different altitudes along with the features selected from analysing the correlation matric and applying feature selection. As Mars altitude from the surface is not in the dataset, this feature will also be extracted by using the Mars Digital Elevation Model (DEM) GeoTIFF format file. The one challenge when doing this research is that there are no previous studies which create new features of ‘altitude’ using the datasets of EMIRS from EMM.

Table of Contents

List of Figures

List of Tables

List of Equations

Chapter 1 – Introduction

As the technologies are improved lately, scientists and space enthusiasts are more likely to venture into space and explore. These space missions encourage them to explore more about the neighbouring planets. Among these explorations, Martian exploration missions have happened to be one of the most popular explorations. According to Priyadarshini and Puri 2021, scientific people have predicted about human extinction in the future. Therefore, scientists are willing to search for another suitable planet. They also stated that Mars is the only option among other planets to colonize due to several scientific facts. Scientists and Space Organizations such as NASA and European Space Agency have been sending multiple missions to Mars ever since last decades. One important fact about Martian exploration is its weather patterns. Understanding and being able to predict the Mars weather patterns can change the conditions of future Martian exploration missions. With the data collected by the previous missions and the use of advanced machine learning algorithms, can create new opportunities and achieve new goals not only for understanding the trend of Mars weather patterns but also for extracting new important information to make accurate predictions (Pant et al. 2023). By predicting accurately of Mars weather patterns, people can make plans and foreseen what the conditions be like for manned missions before landing on Mars.

A spacecraft called 'Hope' was launched by Emirates Mars Mission on July 20, 2020, to study the Martian atmosphere using three scientific instruments that are the Emirates Mars InfraRed Spectrometer (EMIRS), the Emirates eXploration Imager (EXI) and the Emirates Mars Ultraviolet Spectrometer (EMUS) (Amiri et al. 2022). According to Amiri et al. 2022, these instruments are used to make measurements at infrared, visible and the wavelengths of ultraviolet. This research will mainly focus on experimenting with the data of surface temperature to the latitude with the conditions meeting with the other seven useful features which are selected after applying feature selection on the original dataset collected by Emirates Mars InfraRed Spectrometer (EMIRS) instrument. Although there are studies carried out on the data collected by EMIRS, the aspects of predicting Mars surface temperatures to the variation of the altitude still yet to be explored.

1.1 Research Purpose

Researchers and professionals had been carried out on predicting the Mars weather pattern. Moreover, there are studies about predicting the Mars weather using the data collected from the Emirates Mars InfraRed Spectrometer (EMIRS) instrument. However, the research on predicting surface temperatures on different altitudes using the data collected form the EMIRS has yet to be explore. The dataset collected and used in this research starts from February 2021 to August 2022 and level 3 atmospheric reading dataset and level 3 surface temperature reading dataset is merged to obtain more complex and robust data to explore. In contrast, with this enrich data this research will focus on predicting the surface temperature to the latitude along with the other features selected carefully after performing the analysis. Therefore, the questions to be answered from this research will be

- Can the surface temperature of the Mars be predicted on different altitudes using Machine Learning Algorithms based on the data collected from the Emirates Mars InfraRed Spectrometer (EMIRS)?
- Can meaningful and effective visualization and statistical analysis be performed based on the results from using Machine Learning Algorithms?

1.2 Aims and Objective

The aims of the research can be divided into three categories. Since there are two types of datasets, level 3 atmospheric reading data and level 3 surface temperature reading data, collected from year 2021 and 2022, these datasets need to merge to achieve a new complex and big dataset. Merging these datasets is going to be the first aim to achieve for this research. Furthermore, the next aim is to create a new feature, altitude, and this is going to be extracted and calculated using latitude and longitude data from EMIRS instrument and append it to the new dataset derived from merging the original datasets.

The third aim is to build or develop machine learning algorithms to predict the surface temperature based on the features selected by doing feature analysis using the new dataset obtained from the previous step. As a conclusion to this aim, different machine learning algorithms will be developed using the new dataset which can give a new insight or trend of the dataset.

Lastly, the final aim is to evaluate each result from different machine learning algorithms to determine the best algorithm’s result to do the further data analysis which would include statistical analysis and visualizations.

The objective of this research is to create a new complex and robust dataset and apply feature selection to select only important features to test on different machine learning algorithms and evaluate the results. By doing so, the best algorithm which has the best score of predicting the Mars surface temperature on different altitude will be obtained.

1.3 Justification of Research

This research will create a newly formed updated dataset along with new data preprocessing, visualizations and machine learning models which will predict the surface temperature with the other important or necessary features using the updated dataset collected from the Emirates Mars Mission (EMM) and its Emirates Mars Infrared Spectrometer (EMIRS) instrument. The study will also carry out training and model evaluations of enhanced machine learning models which will provide accurate predictions of surface temperatures of Mars depending on different altitudes. With the Mars exploration missions are getting more popular, getting the prediction of different temperatures depending on different altitudes can be handy for the future Mars missions by planning or managing the exploration based on the type of operation that is going to be carried out. With the help of advanced machine learning algorithms using on the updated dataset, the target of this research is to develop the most accurate machine learning algorithm to predict the surface temperature of Mars on different altitudes. Furthermore, useful visualisations and meaningful statistical analysis will be carried out to investigate and extract meaningful insights from the dataset.

1.4 Scope of Research

The scope of this research can be distinguished into three parts. Firstly, creating a newly formed dataset emerged from using level 3 atmospheric and level 3 surface emission data collected from EMIRS instrument of EMM. Secondly, developing meaningful visualizations and statistical analysis to understand the trend or relation between the features of the updated dataset using feature importance and correlation analysis. Lastly and the main aim is to develop an advanced machine learning algorithm which can predict accurately of the surface temperature of Mars on the different altitudes using a newly formed updated dataset. Moreover, the study will also include the development of multiple prediction models and evaluate the performance of each model to see which model can perform best using the updated dataset.

1.5 Research Outlines

The detailed outlines used in this research are stated step by step below:

- **Investigating the Dataset from Emirates Mars Infrared Spectrometer (EMIRS) Instrument –** Studying the domain of Mars weather exploration to obtain a complete understanding of its features.
- **Studies of Emirates Mars Mission (EMM) and EMIRS Instrument –** Studying and exploring about the Emirates Mars Mission (EMM) and one of its instruments, Emirates Mars Infrared Spectrometer (EMIRS).
- **Reviewing the Publications from EMM –** Reviewing and analyzing the publications from the Emirates Mars Mission (EMM) thoroughly to understand and take out the important knowledge about it.
- **Studying and Investigating Machine Learning Models for Mars Weather Prediction –** Exploring different Predictive Machine Learning Models to choose the most appropriate models to use for this research.
- **Merging the Datasets and Data Preprocessing –** Merging level 3 atmospheric and level 3 surface temperature dataset to obtain a newly robust dataset and carrying out the preprocessing steps on the updated dataset.

- **Statistical Analysis –** To achieve valuable and important insights, statistical analysis is carried out.
- **Data Visualizations –** developing meaningful and useful visualizations is important to understand the trend of the data contained in the updated dataset.
- **Training, Testing and Evaluation of the Machine Learning Models –** Developing models and evaluation between each model performance to check the best model's performance score.
- **Comparison and Discussion between the Existing Studies –** Critical evaluation and discussion between the existing literatures to current study.
- **Exploring for Future Works –** Recommendation of the future works that could be implemented or carried out in addition to the current research.

Chapter 2 – Literature Review

2.1 Related Works

Several studies had done on the dataset obtained from EMM EMIRS instrument. These studies are officially published from the Emirates Mars Mission in their website. A study called ‘First Assimilation of Atmospheric Temperatures from the Emirates Mars InfraRed Spectrometer’, carried out by Young et al. (2022), apply the data collected from the EMIRS (Emirates Mars Infrared Spectrometer) to research on Mars atmosphere. To obtain a better understanding of the atmosphere of Mars, Young et al. (2022) used a combination of monitoring the atmosphere with a model named ‘the Mars Planetary Climate Model). This study evidenced that to analyse the Mars weather system and to analyse warm fronts like structures with focusing on the everyday temperature and wind patterns of Mars can be done by using the data from EMIRS.

Moreover, the dust storm that occurred on Mars during September 10, 2021, which was not during the usual dust storm season on Mars, was studied by Gebhardt et al. (2022). The nature of the dust storm’s immobilizing state was captured by Emirate Mars Mission’s series of four images within a short amount of time. In summary, Gebhardt et al. (2022)’s research showed that the relationship between the dynamics of the atmosphere and the dust storm. This special observation insisted new and meaningful insights into the characteristics of unusual dust storms on the Martian atmosphere.

In addition, another research investigated by Atwood et al. (2022) studied to analyse the day-time variation of optical depths of water ice clouds on Mars during aphelion season using the data observed by the Emirates Mars Infrared Spectrometer (EMIRS) from Emirates Mars Mission spacecraft called ‘Hope’. This research aims to understand the nature of the cloud using different local times and latitudes from the dataset. One notable cloud type, the aphelion cloud belt (ACB), which occurred only in Martian cold season and at low latitudes indicates a midday lowest along with the greater optical depths in the morning and afternoon. This study concluded by offering new valuable insights about the nature of the Martian cloud which can associate with atmospheric modelling and the development of the in-depth study of the Martian weather.

With the use of data collected from the Emirates Mars InfraRed Spectrometer from the Emirates Mars Mission, another study carried out by Fan et al. (2022), undergoes to investigate the variation of the day-time temperature in the atmosphere of Mars. A significant achievement done by this research was detecting a new feature that is ter-diurnal tide, which was undetected in the previously research.

In contrast, the data collected by the EMIRS is used to perform various research and study not only to explore the existing nature of the weather patterns on Mars but also to extract new features like unusual dust storm and the characteristic of the aphelion cloud belt which occurred within the unusual season in the atmosphere of Mars.

2.2 Machine Learning on Mars Weather Prediction

Priyadarshini and Puri (2021) purposed several models when forecasting Mars Weather such as Support Vector Regression (SVR), Convolutional Neural Network (CNN), Long Short – Term Memory (LSTM), and proposed – residual CNN – LSTM. This paper aimed to predict the temperature for the year 2018 from training the models on the data in 2012 – 2017. In addition, this experiment aimed to explore the prediction of the four different aspects of maximum and minimum temperatures using terrestrial dates and maximum and minimum temperatures using Solar Days (SOL). Then they evaluated the models using R-squared, Mean Absolute Error, Mean Squared Error and Root Mean Squared Error. Priyadarshini and Puri (2021)'s paper concluded by developing the advanced machine learning models and applying them on the four experiments as mentioned above resulting the Long Short – Term Memory to be the best model in forecasting Mars weather with the R-squared and MAE value of 0.8640 and 0.1257 respectively.

Another research carried out by Pant et al. (2023) studied to forecast the Mars' weather patterns which can help in the future Mars Missions. This paper focuses on implementing clustering and regression algorithms on the data set provided by the Curiosity Rover's Rover Environmental Monitoring Station (REMS). For the regression model Pant et al. (2023) used Linear Regression and for the clustering model using k-Means. The authors concluded their research by stating that the linear regression model performed very well on the dataset used in their research with the accuracy of 85%.

2.3 Suitable Machine Learning Algorithms for Mars Weather Prediction on EMIRS Instrument Dataset

2.3.1 Linear Regression

The requirement of using linear regression is to have the parameters in a linear shape. This regression allows to study the relationship between once or more dependent variables to the predicting variables. Although there are three types of linear regression model. They are known as simple linear regression model, multiple linear regression models and nonlinear regression models. However, this research will only focus on utilizing simple linear regression models to determine the baseline of the means. The simple linear regression can be understood as the following equation.

$$y = \beta_0 + \beta_1 x + \varepsilon,$$

Equation 2.3.1 Simple Linear Regression

From the Equation 2.3.1, y is defined as the dependent variable, β_0 is defined as y interception. β_1 is the gradient of the regression line, x is usually called the independent variable and ε is defined as a random error. In the simple linear regression, the error is usually assumed to be distributed normally with E=0 and a constant variance Var (ε) = σ_2 (Yan and Su 2009).

2.3.2 Gradient Boosting Regressor

Due to Gradient Boosting Regressor's advantages proposed by (Cai et al. 2020), that is the low prediction errors and its nice stability, this method was chose to apply in this research. Singh et al. (2021) also agreed that using gradient boosting regression algorithm can reduce the error rate as it consists of an ensemble learning approach. According to Singh et al. (2021), Gradient Boosting Regressor can be defined as the following equation.

$$F_n(x_t) = \sum_{i=1}^{n} f_i(x_t)$$

Equation 2.3.2 Gradient Boosting Regressor

Here, every $f_i(x_t)$ indicates every decision tree.

2.3.3 Random Forest Regressor

According to Singh et al. (2021), Random Forest Regressor is one of the most famous decision tree algorithms which produced numerous decision trees based on the given input data. When applying Random Forest Regressor, it then first split the dataset into several sub-dataset where the decision trees are created. The model then combined the predicted results of every decision tree to achieve a more reliable and accurate prediction.

2.3.4 K-Nearest Neighbors Regressor

Sumayli (2023) stated in the paper that K-Nearest Neighbors Regressor was used as one of the machine learning models for optimization of biofuel production. Author also explained that this model is easy to interpret and understand as the model uses the K nearest data points values from the training data to predict the new value of an observation. He also mentioned that K-Nearest Neighbors Regressor is widely used in forecasting weather patterns. Therefore, this model was chosen to use in this research. The calculation of the most popular distance metric applied to KNN regression is Euclidean distance and its equation is shown below.

$$d(xi, xj) = \sqrt{\sum_{k=1}^{p} \left(x_{ik} - x_{jk}\right)^2}$$

Equation 2.3.4 K-Nearest Neighbors Regression

2.3.5 Artificial Neural Network Regressor (Multilayer Perceptron)

Moreover, Sumayli (2023) applied Multilayer Perceptron model for optimization of biofuel production. According to Sumayli (2023) MLP model is a kind of Artificial Neural Network especially used to do the supervised learning problems such as classification and regression. Therefore, this model was also chosen to use in this research. The formula for the MLP model, when assuming there is a single output and single hidden layer, is shown below.

$$\widetilde{\mathbf{y}} = \delta_2 \left(\sum_{i=1}^{m} \left(w_i^{(2)} \delta_1 \left(\mathbf{X}\right)\right) + \mathbf{b}^{(2)}\right) \mathbf{X} = \sum_{j=1}^{n} \left(\mathbf{x}_j \mathbf{w}_{xj}^{(1)}\right) + \mathbf{b}^{(1)}$$

Equation 2.3.5 Artificial Neural Network (MLP) model

Chapter 3 – Methodology

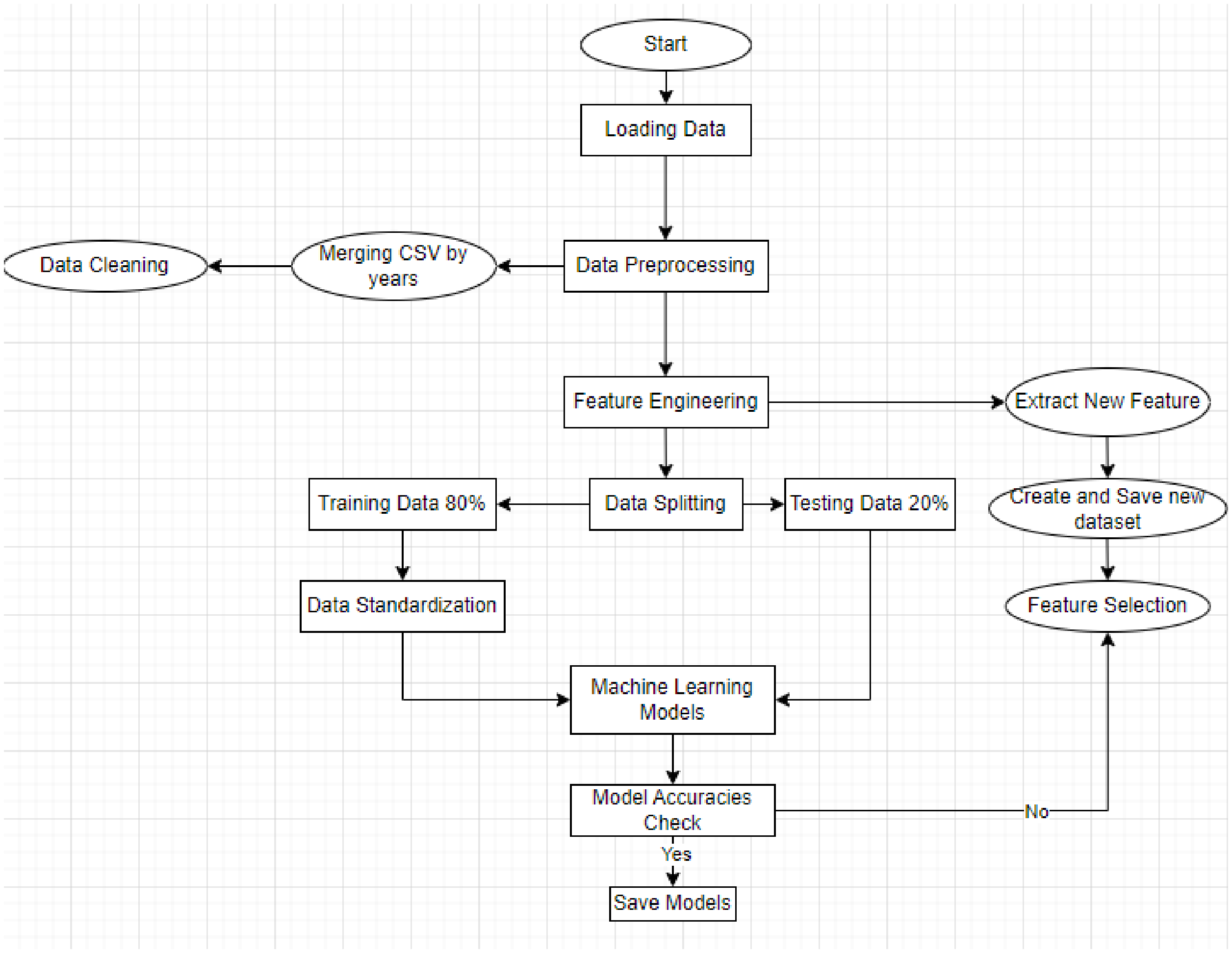


Fig 3 Experiment Flow Diagram

3.1 Dataset

The dataset used in this research consists of the data collected from the Emirates Mars Infrared Spectrometer (EMIRS) instrument which is one of the instruments from the Emirates Mars Mission 'Hope'. These data are collected from February 2021 to August 22 which consists of level 3 derived atmospheric and surface emission readings. The original datasets are in the format of FITS (Flexible Image Transport System) which can be obtained by downloading from official Emirates Mars Mission (EMM) website at https://sdc.emiratesmarsmission.ae/data/science. From studying the datasets, it showed that there are two main subsets in them which are the 'Level 3 Atmospheric' data which is referred to as 'l3atm' and the 'Lever 3 Emissivity and Surface' data which is referred to as 'l3emiss'. To extract the meaningful and useful features, understanding the meaning of the data columns are necessary. The table below shows the names and meanings of the features included in this research.

Feature	Level	Data Type	Description
temp	l3atm	float	Atmospheric temperatures collected in the form of array type. The values of 19 are scaling heights from every 0.25 pressure starting at 0.25 and ending at 4.75 scale heights above the surface.
chi2temp	l3atm	float	Quality of atmospheric temperature fit, the chi-squared is calculated between best fit and observed.
pres0	l3atm	float	Surface pressure estimation.
taudust	l3atm	float	Extinction optical depth of dust column at 1075cm-1.
tauice	l3atm	float	Extinction optical depth of ice column at 825cm-1
chi2a	l3atm	float	Aerosol fit's quality which is the chi-squared difference.
tsurf			
latitude	l3atm, l3emiss	float	Center of the IFOV footprint's latitude.
kinetic_temp	l3emiss	integer	Kinetic temperature of the surface.

Table 3.1 Data Description provided by EMM EMIRS Data Product Guide

3.2 Requirements for the Experiment

Due to the high volume and big size of the datasets, it is unable to load the dataset using free browser-based tools such as Google Collaboration or Colab to carry out the preprocessing steps. Therefore, the initial dataset merging was carried out on a desktop PC powered by an Intel Core i7 processor, equipped with 16GB of RAM and featuring with SSDs with 2 TB and 512 GB respectively. Data merging process was carried out using internal memory and storage using Jupyter Notebook with the help of the RAM and storage space as mentioned above. After merging the datasets into a newly formed dataset, the process of feature engineering, data splitting, model training and model evaluation were carried out using Google Colab. The import libraries used throughout this research are listed below:

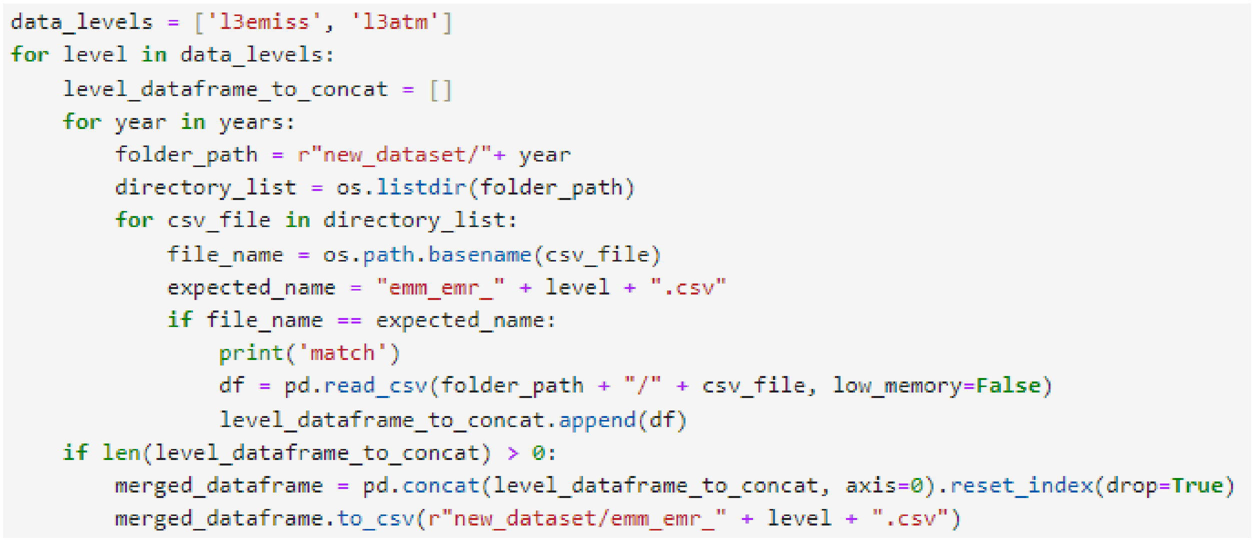
- **pandas**: Used to manipulate and analyse the data.
- **numpy**: Used to do the calculations of the numerical data and operates array.
- **seaborn**: Used to create useful visualization of data statistical.
- **matplotlib**: Used to create advanced and interactive visualizations.
- **Rasterio**: Used to read GeoTIFF format file.
- **sklearn** (scikit-learn): Used to deploy machine learning models, data splitting, feature selection, model selection and evaluation.
- **mpl_toolkits**: Used to apply matplotlib's extensions and special tools.
- **tensorflow**: Used to apply deep learning model.

3.3 Data Preprocessing

3.3.1 CSV merged by Years

The initial datasets provided for this research were already merged by months and converted into csv files. Therefore, there are four datasets of Level 3 surface emission readings for 2021 and 2022 and Level 3 derived atmospheric readings for 2021 and 2022. However, these datasets are needed to be merged to obtain a single dataset for each reading. The process of merging the datasets of reading based on the level is shown in

Figure. Below. This step took about fifteen to twenty minutes to run on the Jupyter notebook as the size of the file is big to run on Google Colab.



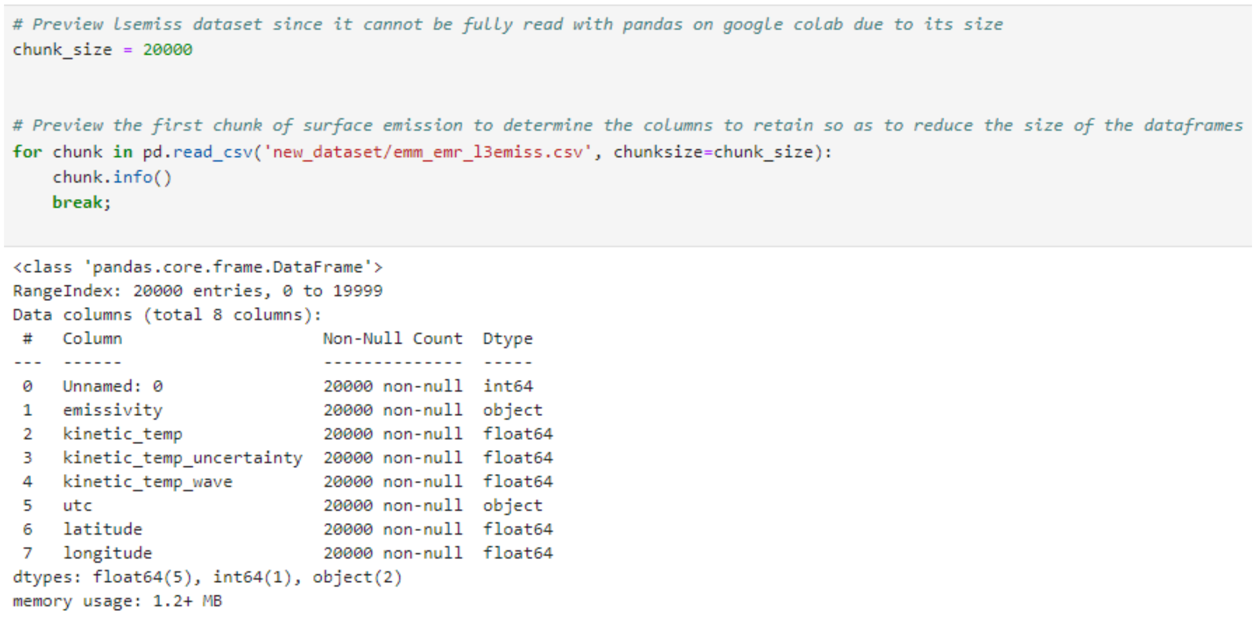
```
data_levels = ['l3emiss', 'l3atm']
for level in data_levels:
    level_dataframe_to_concat = []
    for year in years:
        folder_path = r"new_dataset/"+ year
        directory_list = os.listdir(folder_path)
        for csv_file in directory_list:
            file_name = os.path.basename(csv_file)
            expected_name = "emm_emr_" + level + ".csv"
            if file_name == expected_name:
                print('match')
                df = pd.read_csv(folder_path + "/" + csv_file, low_memory=False)
                level_dataframe_to_concat.append(df)
    if len(level_dataframe_to_concat) > 0:
        merged_dataframe = pd.concat(level_dataframe_to_concat, axis=0).reset_index(drop=True)
        merged_dataframe.to_csv(r"new_dataset/emm_emr_" + level + ".csv")
```

Fig 3.3.1: Merging the datasets from each year based on level

3.3.2 Data Cleaning

After successfully merging the CSV data files of each year based on each level, the new of Level 3 surface emission reading and Level 3 derived atmospheric reading datasets are created with the sizes of 13.8 gigabytes and 991 megabytes respectively. Hereafter, the preprocessing step of data cleaning was carried out using Jupyter notebook by installing through commend prompt since it was unsuccessful to achieve by applying Google Colab as the size of the datasets are too big to upload them on google drive.

After importing the required libraries, the dataset of Level 3 surface emission was generated as a partial view of chunk to investigate which features to use and which features to drop according to the description of the data given by the EMM for the EMIRS instrument. By doing so, the memory usage will be mitigated. The process of this step and the method used in this step is shown in Figure 3.3.2a below.

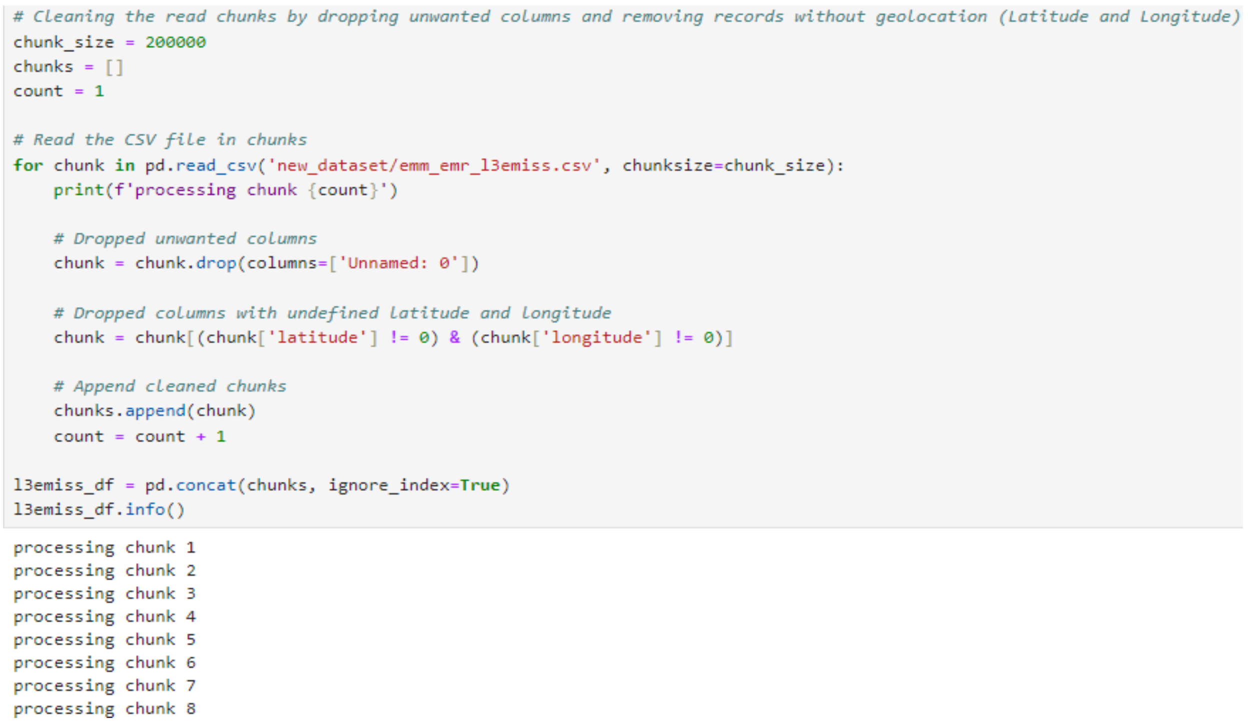


```
# Preview lsemiss dataset since it cannot be fully read with pandas on google colab due to its size
chunk_size = 20000


# Preview the first chunk of surface emission to determine the columns to retain so as to reduce the size of the dataframes
for chunk in pd.read_csv('new_dataset/emm_emr_l3emiss.csv', chunksize=chunk_size):
    chunk.info()
    break;
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 8 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Unnamed: 0                20000 non-null  int64
 1   emissivity                20000 non-null  object
 2   kinetic_temp              20000 non-null  float64
 3   kinetic_temp_uncertainty  20000 non-null  float64
 4   kinetic_temp_wave         20000 non-null  float64
 5   utc                       20000 non-null  object
 6   latitude                  20000 non-null  float64
 7   longitude                 20000 non-null  float64
dtypes: float64(5), int64(1), object(2)
memory usage: 1.2+ MB
```

Fig 3.3.2a Partial view of a l3emiss data chunk

The next step is to append the chunks to form into a data frame. However, before transforming them into a data frame, some unnecessary columns must be excluded as they are considered irrelevant for this research according to the domain knowledge based on the EMM EMIRS Data Product Guide. The process of data chunks transforming into a data frame is shown in Figure 3.3.2c below:

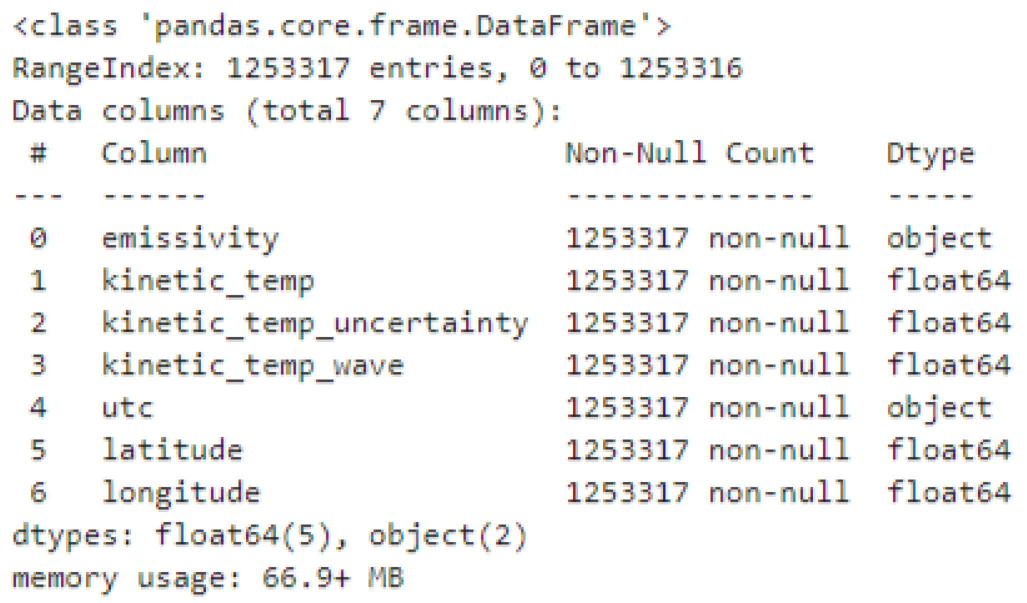


```python
# Cleaning the read chunks by dropping unwanted columns and removing records without geolocation (Latitude and Longitude)
chunk_size = 200000
chunks = []
count = 1

# Read the CSV file in chunks
for chunk in pd.read_csv('new_dataset/emm_emr_l3emiss.csv', chunksize=chunk_size):
    print(f'processing chunk {count}')

    # Dropped unwanted columns
    chunk = chunk.drop(columns=['Unnamed: 0'])

    # Dropped columns with undefined latitude and longitude
    chunk = chunk[(chunk['latitude'] != 0) & (chunk['longitude'] != 0)]

    # Append cleaned chunks
    chunks.append(chunk)
    count = count + 1

l3emiss_df = pd.concat(chunks, ignore_index=True)
l3emiss_df.info()
```

```
processing chunk 1
processing chunk 2
processing chunk 3
processing chunk 4
processing chunk 5
processing chunk 6
processing chunk 7
processing chunk 8
```

Fig 3.3.2b Appending chunks and excluding unwanted columns

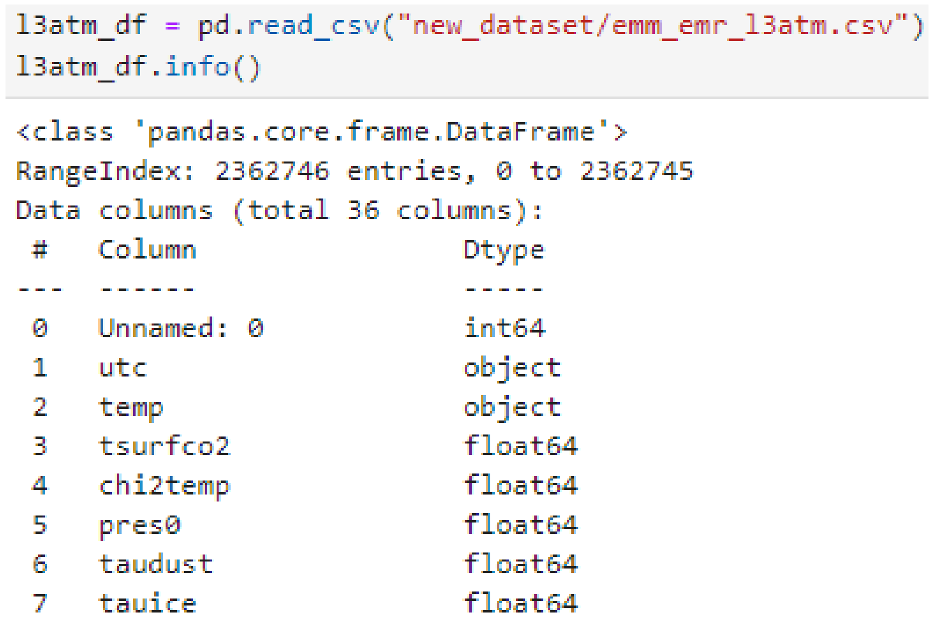
The data frame resulting from transformation of the data chunks consists of seven features which can be seen in Figure 3.3.2d. The remaining columns after dropping the unwanted columns are emissivity, kinetic temperature, kinetic temperature uncertainty, kinetic temperature wave, UTC, longitude and latitude.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1253317 entries, 0 to 1253316
Data columns (total 7 columns):
 #   Column                     Non-Null Count    Dtype
---  ------                     --------------    -----
 0   emissivity                 1253317 non-null  object
 1   kinetic_temp               1253317 non-null  float64
 2   kinetic_temp_uncertainty   1253317 non-null  float64
 3   kinetic_temp_wave          1253317 non-null  float64
 4   utc                        1253317 non-null  object
 5   latitude                   1253317 non-null  float64
 6   longitude                  1253317 non-null  float64
dtypes: float64(5), object(2)
memory usage: 66.9+ MB
```

Fig 3.3.2c Resulting data frame with its features

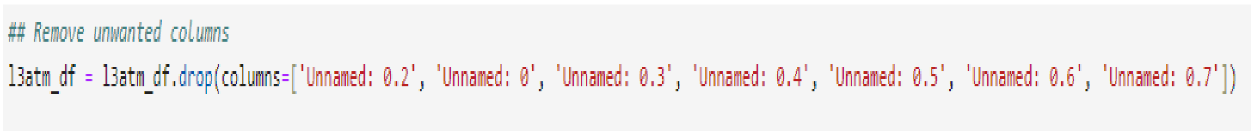
The next step is to import the level 3 atmospheric reading dataset into a data frame for analyzing and to explore the data features and to drop the irrelevant features which are not necessary for this research.



```
l3atm_df = pd.read_csv("new_dataset/emm_emr_l3atm.csv")
l3atm_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2362746 entries, 0 to 2362745
Data columns (total 36 columns):
 #   Column                 Dtype
---  ------                 -----
 0   Unnamed: 0             int64
 1   utc                    object
 2   temp                   object
 3   tsurfco2               float64
 4   chi2temp               float64
 5   pres0                  float64
 6   taudust                float64
 7   tauice                 float64
```

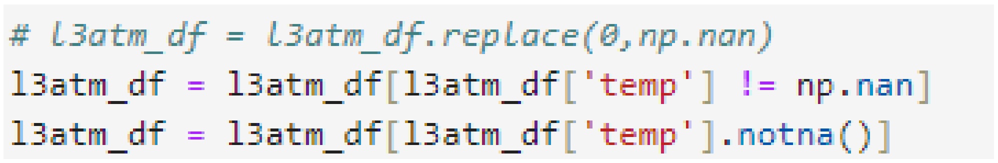
Fig 3.3.2d Loading the CSV file of l3atm dataset



```
## Remove unwanted columns
l3atm_df = l3atm_df.drop(columns=['Unnamed: 0.2', 'Unnamed: 0', 'Unnamed: 0.3', 'Unnamed: 0.4', 'Unnamed: 0.5', 'Unnamed: 0.6', 'Unnamed: 0.7'])
```

Fig 3.3.2e Dropping unwanted columns

In addition, the Level 3 derived atmospheric dataset was set to do a filtering process to ensure that there are no missing values in the data frame.



```
# l3atm_df = l3atm_df.replace(0,np.nan)
l3atm_df = l3atm_df[l3atm_df['temp'] != np.nan]
l3atm_df = l3atm_df[l3atm_df['temp'].notna()]
```

Fig 3.3.2f replacing 0 with nan values

3.3.3 Outliers

The method used in this research to identify outliers is box plotting the features. From this, the data are analyzed carefully and manually to determine the outliers. To predict the surface temperature to the different altitudes, the main features to look for outliers would be ‘temp’ and ‘latitude’. A decision was made to consider the temperature values of less than 140 and over 220 to be outliers. The details of the box plot can be seen in Figure 3.3.3 below.

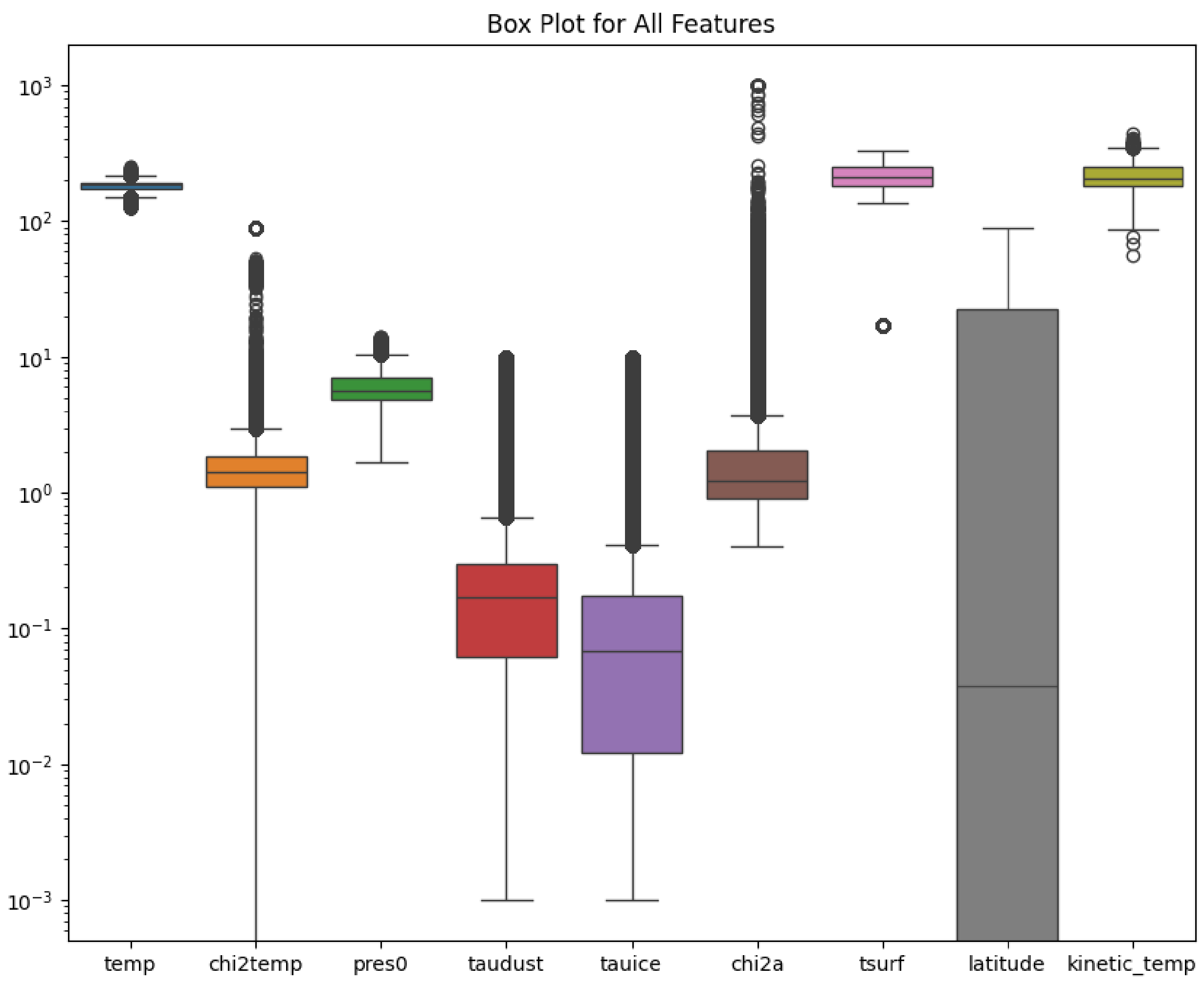
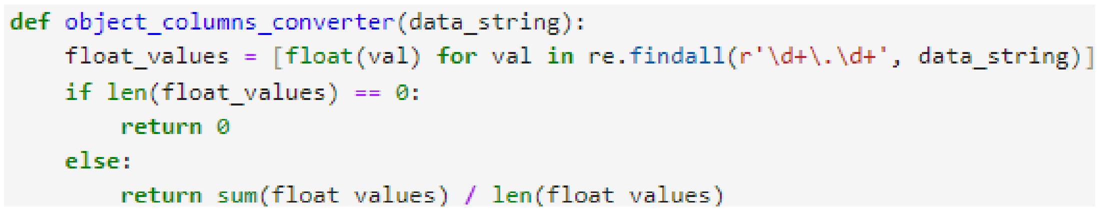


Fig 3.3.3 Box plot of every feature to check outliers

3.4 Feature Engineering and Merging Datasets

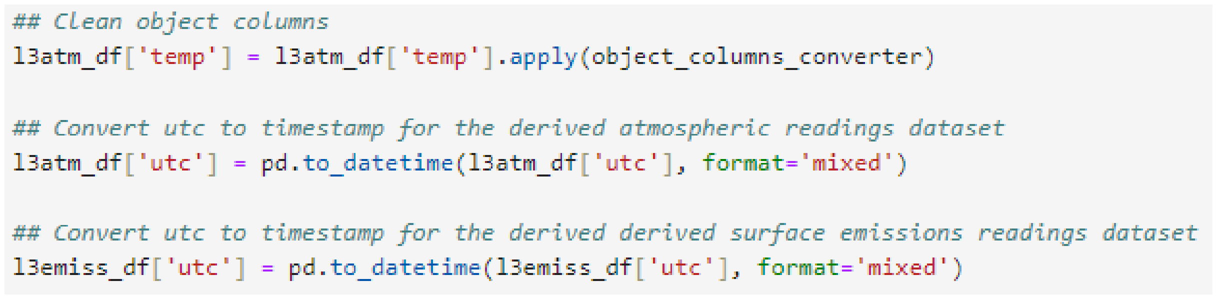
According to the EMM EMIRS Data Product Guide, the feature 'temp' consists of an array of retrieved atmospheric temperatures which contains 19 temperature values corresponding to a specific pressure scale height and the spaced values at intervals of 0.25 pressure scale heights, in a range starting from 0.25 to 4.75 scale heights above the planet surface. By setting the means of these temperatures, they can be used as the temperature data for this study. To carry out this process a helper method was used, and the details of the helper method deployed can be seen in Figure 3.4a.



```
def object_columns_converter(data_string):
    float_values = [float(val) for val in re.findall(r'\d+\.\d+', data_string)]
    if len(float_values) == 0:
        return 0
    else:
        return sum(float_values) / len(float_values)
```

Fig 3.4a Helper method to calculate mean of features with a range of readings

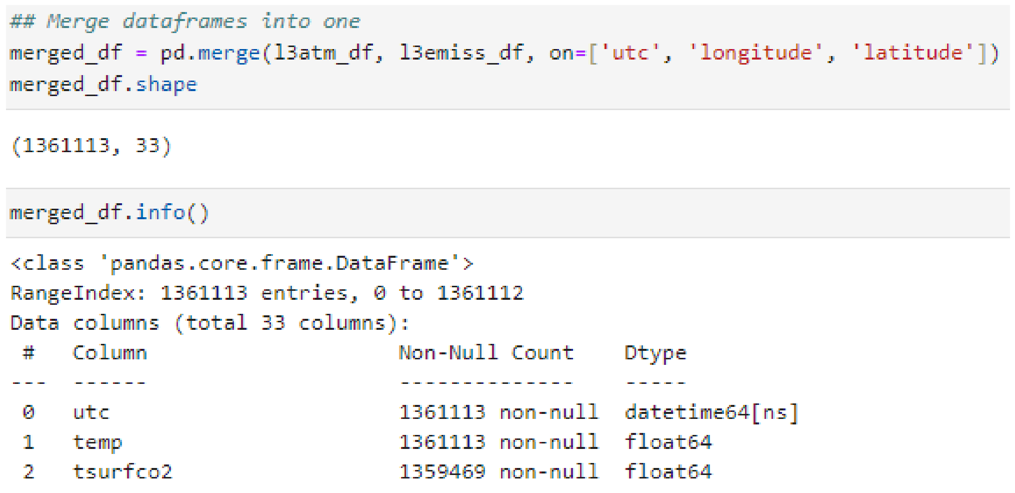
After using the helper method, the ‘temp’ feature was transformed. Moreover, the ‘UTC’ feature from both l3atm and l3emiss datasets which represents the data collected date and time, was also converted into datetime format.



```
## Clean object columns
l3atm_df['temp'] = l3atm_df['temp'].apply(object_columns_converter)

## Convert utc to timestamp for the derived atmospheric readings dataset
l3atm_df['utc'] = pd.to_datetime(l3atm_df['utc'], format='mixed')

## Convert utc to timestamp for the derived derived surface emissions readings dataset
l3emiss_df['utc'] = pd.to_datetime(l3emiss_df['utc'], format='mixed')
```

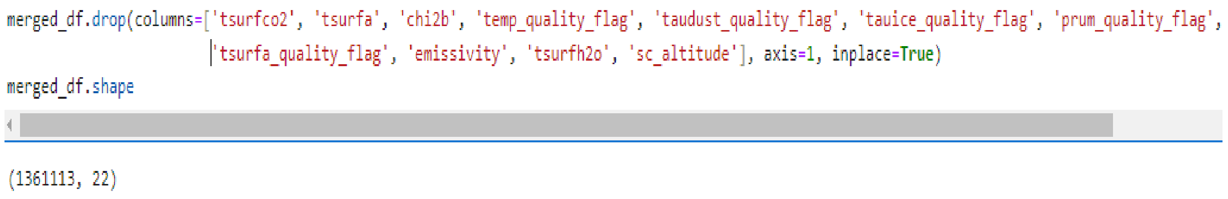
Fig 3.4b Feature engineering

The next step for this stage is to merge the two datasets to obtain a single robust and complex dataset which the machine learning algorithms are going to use on it. To do so, the UTC (datetime data type) feature and the coordinates (Latitude and Longitude) were used as the key identifiers.



```
## Merge dataframes into one
merged_df = pd.merge(l3atm_df, l3emiss_df, on=['utc', 'longitude', 'latitude'])
merged_df.shape

(1361113, 33)

merged_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1361113 entries, 0 to 1361112
Data columns (total 33 columns):
 #   Column                    Non-Null Count    Dtype
---  ------                    --------------    -----
 0   utc                       1361113 non-null  datetime64[ns]
 1   temp                      1361113 non-null  float64
 2   tsurfco2                  1359469 non-null  float64
```

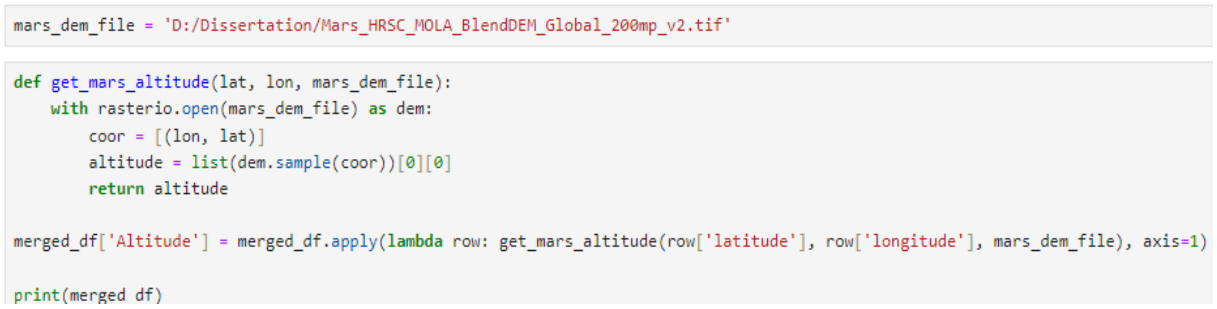
Fig 3.4c Merging Dataset using key identifiers



```
merged_df.drop(columns=['tsurfco2', 'tsurfa', 'chi2b', 'temp_quality_flag', 'taudust_quality_flag', 'tauice_quality_flag', 'prum_quality_flag',
                        'tsurfa_quality_flag', 'emissivity', 'tsurfh2o', 'sc_altitude'], axis=1, inplace=True)
merged_df.shape

(1361113, 22)
```

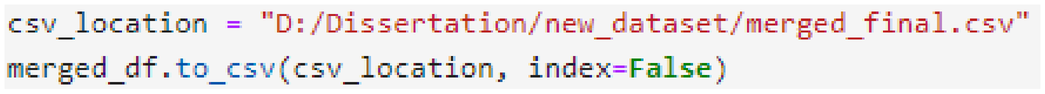
Fig 3.4d Feature Engineering

Subsequently, using latitude and longitude data derived from datasets merging and with the help of Rasterio Library the value of the altitude is calculated and appended to the merged dataset. To calculate the value of the altitude Mars Digital Elevation Model (DEM) GeoTIFF format file was downloaded first. This file was officially published by USGS Astrogeology Science Center and can be downloaded from this website https://astrogeology.usgs.gov/search/map/mars_mgs_mola_mex_hrsc_blended_dem_global_200m. Moreover, after downloading the file, it was loaded using Rasterio Library. Then a function is created to calculate and append the altitude of Mars based on the latitude and longitude data provided by EMIRS instrument. Then the new dataset which includes the altitude of Mars data is saved as in CSV format file.



```
mars_dem_file = 'D:/Dissertation/Mars_HRSC_MOLA_BlendDEM_Global_200mp_v2.tif'

def get_mars_altitude(lat, lon, mars_dem_file):
    with rasterio.open(mars_dem_file) as dem:
        coor = [(lon, lat)]
        altitude = list(dem.sample(coor))[0][0]
        return altitude

merged_df['Altitude'] = merged_df.apply(lambda row: get_mars_altitude(row['latitude'], row['longitude'], mars_dem_file), axis=1)

print(merged_df)
```

Fig 3.4e Extracting Altitude feature and append it to the data frame



```
csv_location = "D:/Dissertation/new_dataset/merged_final.csv"
merged_df.to_csv(csv_location, index=False)
```

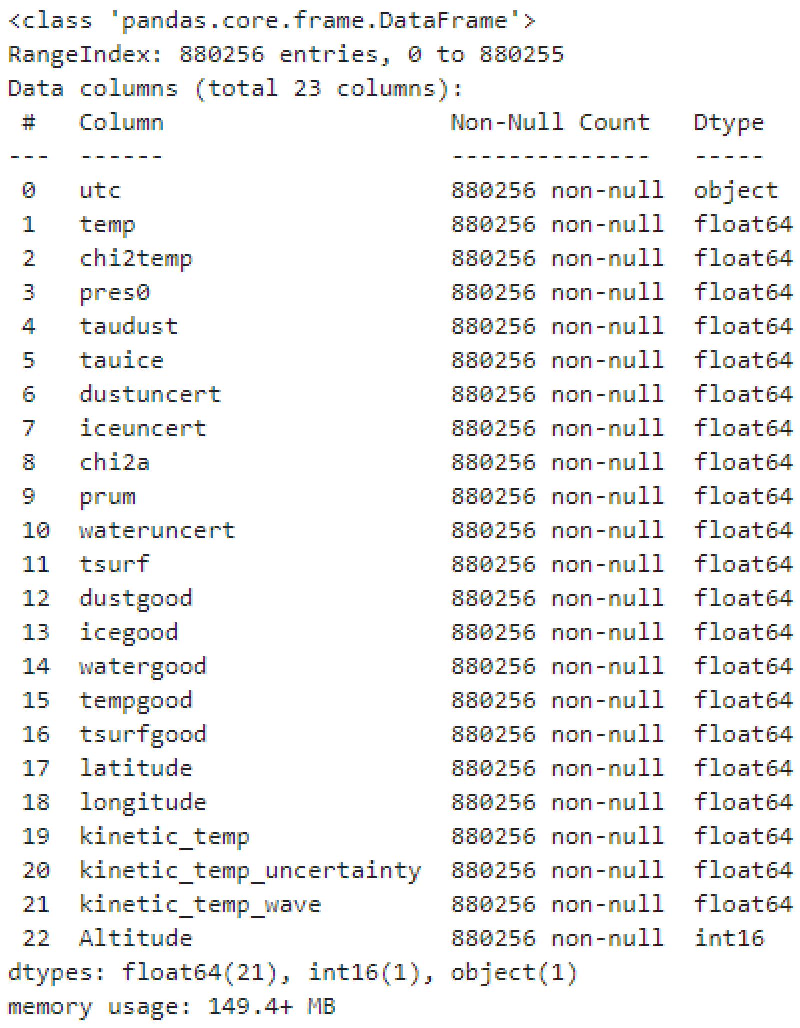
Fig 3.4f Saving CSV file of new merged data frame

3.5 Descriptive Analysis

Understanding the data derived from the new dataset is crucial as this is the dataset where the models will be trained on. Analysis of the dataset includes various methods such as analysing statistical data computations, analysing correlation relationships between each feature and visualising the data to understand and extract important trend. Moreover, different techniques of visualisation were used in this research such as correlation matrix using seaborn library, box plot, histogram and 3D scatter plot.

3.5.1 Data Summary

The data types were initially investigated to verify if the data types were correct to use for this research. After confirming that the data types were correct and the non-value counts for all features are matching with the total amount of data in the dataset.

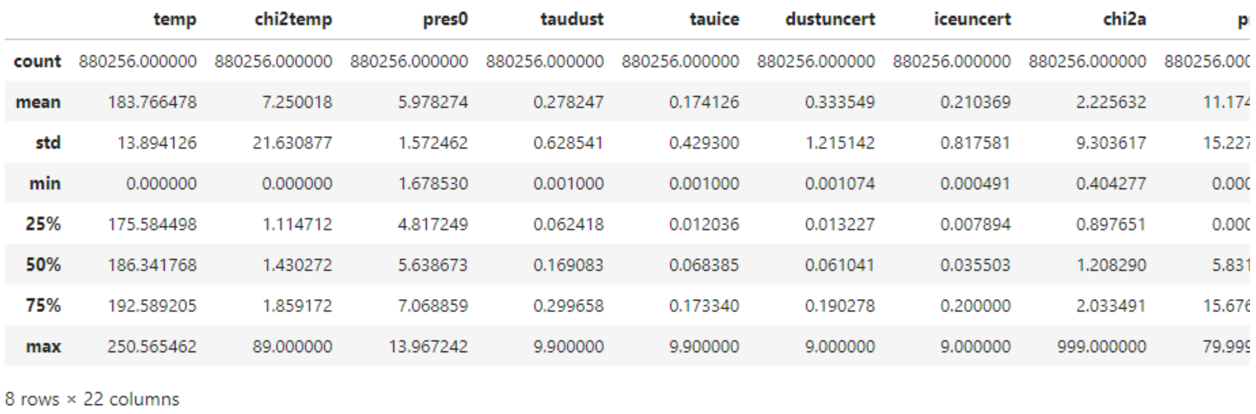


```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 880256 entries, 0 to 880255
Data columns (total 23 columns):
 #   Column                    Non-Null Count   Dtype  
---  ------                    --------------   -----  
 0   utc                       880256 non-null  object 
 1   temp                      880256 non-null  float64
 2   chi2temp                  880256 non-null  float64
 3   pres0                     880256 non-null  float64
 4   taudust                   880256 non-null  float64
 5   tauice                    880256 non-null  float64
 6   dustuncert                880256 non-null  float64
 7   iceuncert                 880256 non-null  float64
 8   chi2a                     880256 non-null  float64
 9   prum                      880256 non-null  float64
 10  wateruncert               880256 non-null  float64
 11  tsurf                     880256 non-null  float64
 12  dustgood                  880256 non-null  float64
 13  icegood                   880256 non-null  float64
 14  watergood                 880256 non-null  float64
 15  tempgood                  880256 non-null  float64
 16  tsurfgood                 880256 non-null  float64
 17  latitude                  880256 non-null  float64
 18  longitude                 880256 non-null  float64
 19  kinetic_temp              880256 non-null  float64
 20  kinetic_temp_uncertainty  880256 non-null  float64
 21  kinetic_temp_wave         880256 non-null  float64
 22  Altitude                  880256 non-null  int16  
dtypes: float64(21), int16(1), object(1)
memory usage: 149.4+ MB
```

Fig 3.5.1 Checking Data Summary

3.5.2 Descriptive Statistics

The previewed of the statistical computation of the data in each feature is shown in the Figure 3.5.2 below. The statistical value of each feature includes their count, mean, standard deviation, minimum value, first quartile value, second quartile value or median value, third quartile value and maximum value.



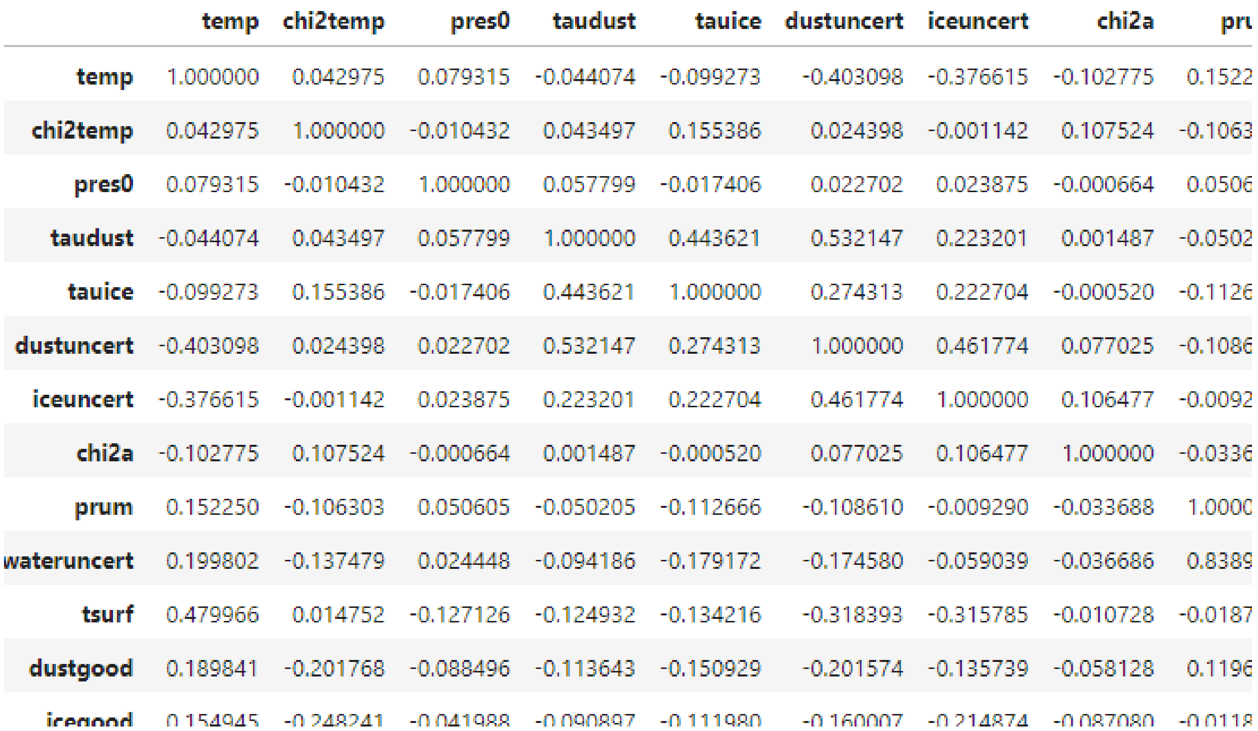
	temp	chi2temp	pres0	taudust	tauice	dustuncert	iceuncert	chi2a	p
count	880256.000000	880256.000000	880256.000000	880256.000000	880256.000000	880256.000000	880256.000000	880256.000000	880256.000
mean	183.766478	7.250018	5.978274	0.278247	0.174126	0.333549	0.210369	2.225632	11.174
std	13.894126	21.630877	1.572462	0.628541	0.429300	1.215142	0.817581	9.303617	15.227
min	0.000000	0.000000	1.678530	0.001000	0.001000	0.001074	0.000491	0.404277	0.000
25%	175.584498	1.114712	4.817249	0.062418	0.012036	0.013227	0.007894	0.897651	0.000
50%	186.341768	1.430272	5.638673	0.169083	0.068385	0.061041	0.035503	1.208290	5.831
75%	192.589205	1.859172	7.068859	0.299658	0.173340	0.190278	0.200000	2.033491	15.676
max	250.565462	89.000000	13.967242	9.900000	9.900000	9.000000	9.000000	999.000000	79.999

8 rows × 22 columns

Fig 3.5.2 Statistical information of the features

3.5.3 Correlation Analysis

To analyse the correlation between each feature, the correlation matrix is calculated and generated. Figure 3.5.3a shows the correlation score between each feature.



	temp	chi2temp	pres0	taudust	tauice	dustuncert	iceuncert	chi2a	pru
temp	1.000000	0.042975	0.079315	-0.044074	-0.099273	-0.403098	-0.376615	-0.102775	0.1522
chi2temp	0.042975	1.000000	-0.010432	0.043497	0.155386	0.024398	-0.001142	0.107524	-0.1063
pres0	0.079315	-0.010432	1.000000	0.057799	-0.017406	0.022702	0.023875	-0.000664	0.0506
taudust	-0.044074	0.043497	0.057799	1.000000	0.443621	0.532147	0.223201	0.001487	-0.0502
tauice	-0.099273	0.155386	-0.017406	0.443621	1.000000	0.274313	0.222704	-0.000520	-0.1126
dustuncert	-0.403098	0.024398	0.022702	0.532147	0.274313	1.000000	0.461774	0.077025	-0.1086
iceuncert	-0.376615	-0.001142	0.023875	0.223201	0.222704	0.461774	1.000000	0.106477	-0.0092
chi2a	-0.102775	0.107524	-0.000664	0.001487	-0.000520	0.077025	0.106477	1.000000	-0.0336
prum	0.152250	-0.106303	0.050605	-0.050205	-0.112666	-0.108610	-0.009290	-0.033688	1.0000
wateruncert	0.199802	-0.137479	0.024448	-0.094186	-0.179172	-0.174580	-0.059039	-0.036686	0.8389
tsurf	0.479966	0.014752	-0.127126	-0.124932	-0.134216	-0.318393	-0.315785	-0.010728	-0.0187
dustgood	0.189841	-0.201768	-0.088496	-0.113643	-0.150929	-0.201574	-0.135739	-0.058128	0.1196
icegood	0.154945	-0.248241	-0.041988	-0.090897	-0.111980	-0.160007	-0.214874	-0.087080	-0.0118

Fig 3.5.3a Correlation Matrix

Then, for the purpose of getting a clear and easy understanding of the correlation between each feature, the matrix is visualized using seaborn heatmap as this can help generate a clear visualization of the correlation matrix as shown in Figure 3.5.3b.

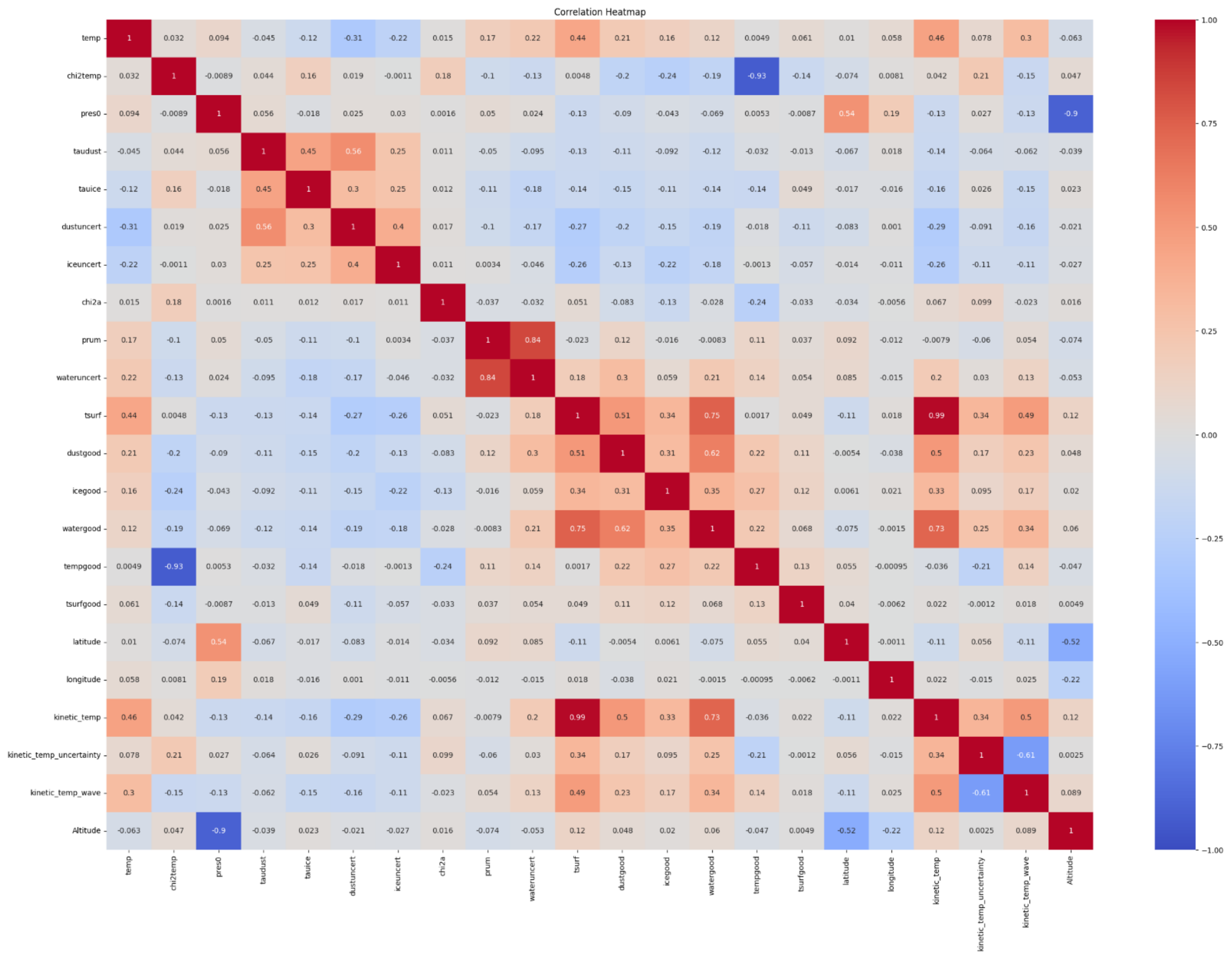
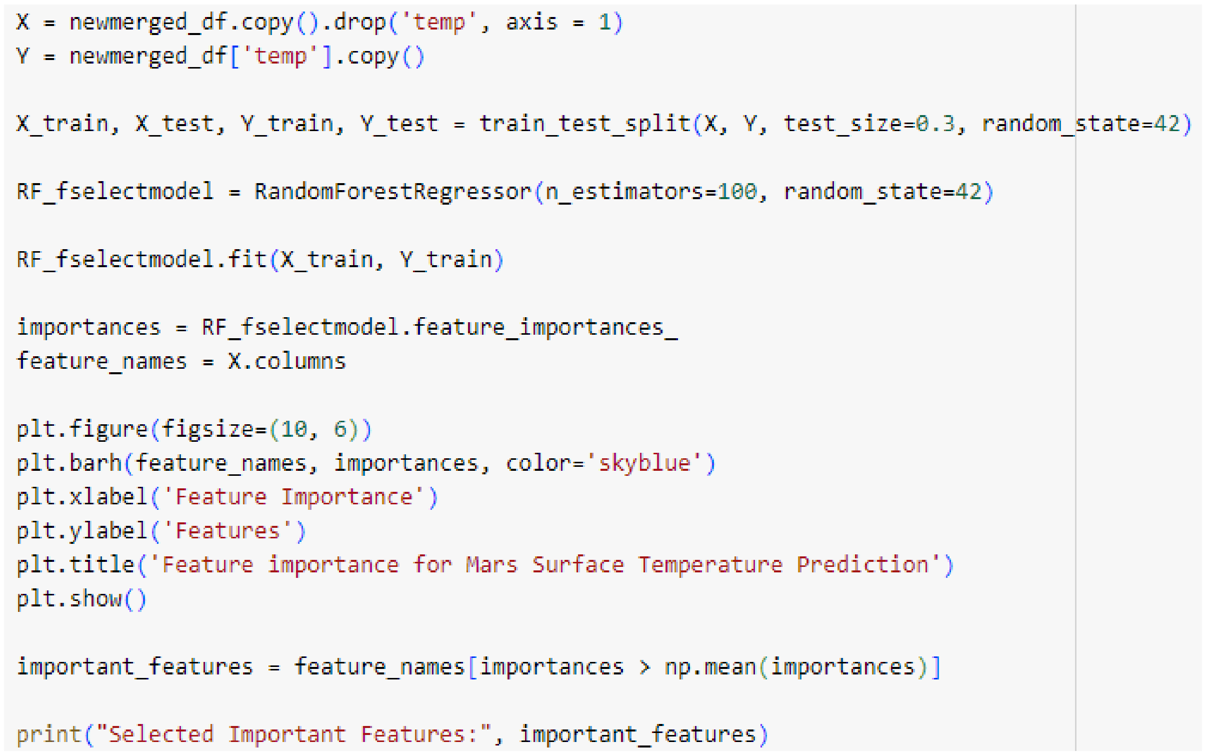


Fig 3.5.3b Correlation Matrix using Seaborn Heatmap

3.5.4 Feature Importance and Selection

Before implementing the models, the process of data splitting into two portions of 70% for training data and 30% of testing data is carried out for the feature importance. This was to validate the importance of each feature compared to the surface temperature. In addition, feature selection using Random Forest Regressor was also used to see which features are necessary to use for this study. Since this study focuses on predicting surface temperature depending on the different latitude, the depended variables should be selected based on the correlation and feature importance of the surface temperature and latitude features. Then, the uncorrelated or the features which has no or less relationship to the main two features were dropped and create a new data frame to obtain the optimum accuracy when using machine learning models.



```
X = newmerged_df.copy().drop('temp', axis = 1)
Y = newmerged_df['temp'].copy()

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state=42)

RF_fselectmodel = RandomForestRegressor(n_estimators=100, random_state=42)

RF_fselectmodel.fit(X_train, Y_train)

importances = RF_fselectmodel.feature_importances_
feature_names = X.columns

plt.figure(figsize=(10, 6))
plt.barh(feature_names, importances, color='skyblue')
plt.xlabel('Feature Importance')
plt.ylabel('Features')
plt.title('Feature importance for Mars Surface Temperature Prediction')
plt.show()

important_features = feature_names[importances > np.mean(importances)]

print("Selected Important Features:", important_features)
```

Fig 3.5.4a Feature Importance using Random Forest Regressor

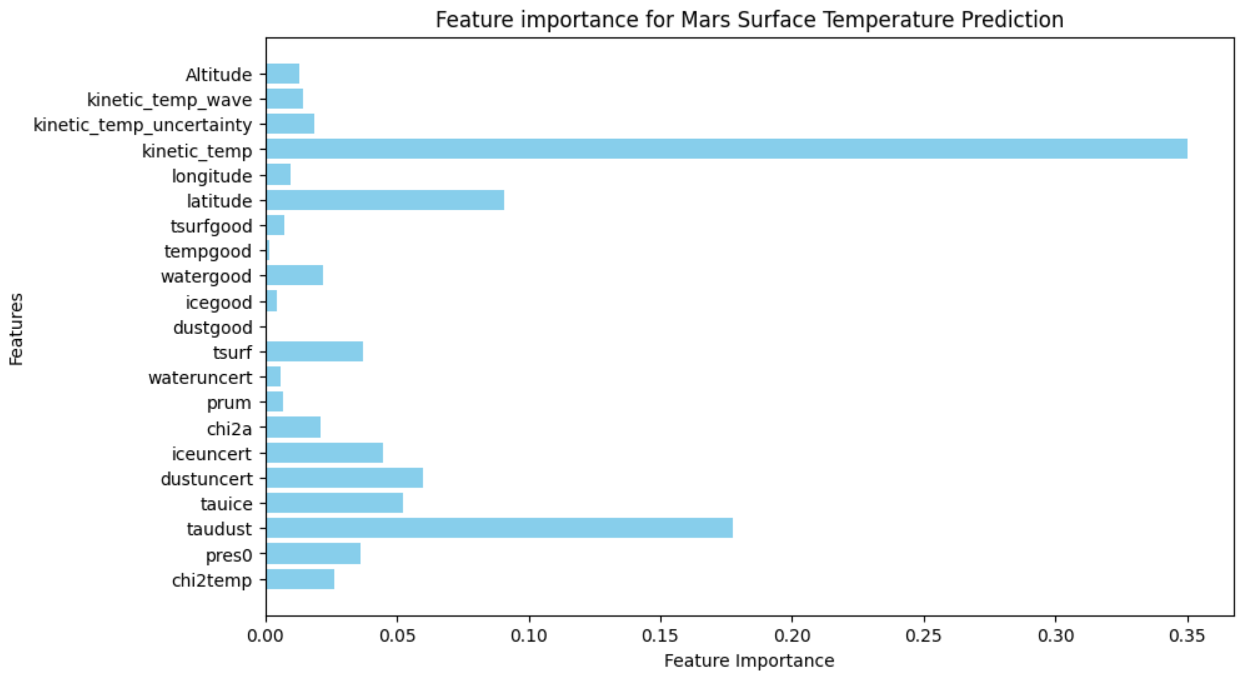
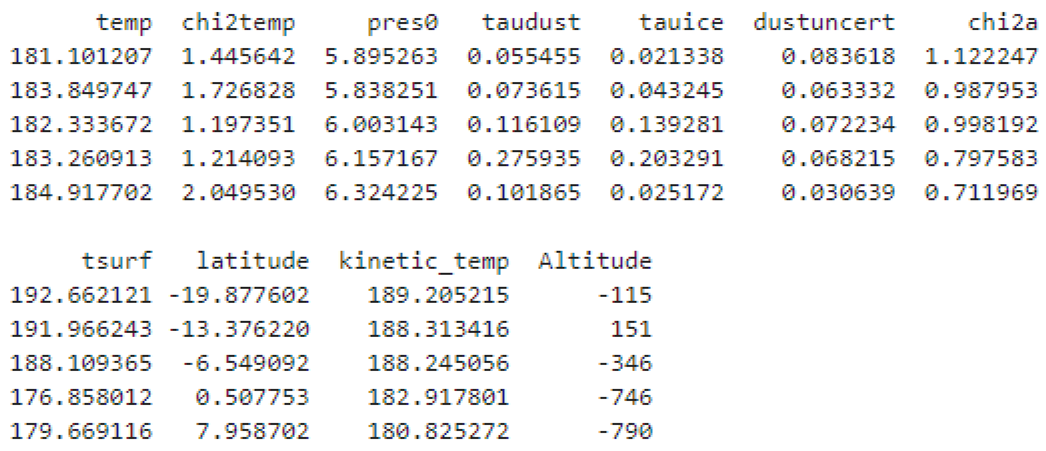


Fig 3.5.4b Horizontal bar graph showing the importance of each feature

After analyzing the correlation, importances of each feature and domain knowledge with the help of EMM EMIRS Data Product Guide, the features ‘longitude’, ‘prum’, ‘iceuncert’, ‘dustgood’, ‘icegood’, watergood’, ‘tempgood’, ‘tsurfgood’, ‘kinetic_temp_uncertainty’, ‘kinetic_temp_wave’, ‘wateruncert’ were dropped. The new data frame is created with the useful features shown in Figure 3.6.1c below.



```
      temp  chi2temp     pres0   taudust    tauice  dustuncert     chi2a
181.101207  1.445642  5.895263  0.055455  0.021338    0.083618  1.122247
183.849747  1.726828  5.838251  0.073615  0.043245    0.063332  0.987953
182.333672  1.197351  6.003143  0.116109  0.139281    0.072234  0.998192
183.260913  1.214093  6.157167  0.275935  0.203291    0.068215  0.797583
184.917702  2.049530  6.324225  0.101865  0.025172    0.030639  0.711969

     tsurf   latitude  kinetic_temp  Altitude
192.662121 -19.877602    189.205215      -115
191.966243 -13.376220    188.313416       151
188.109365  -6.549092    188.245056      -346
176.858012   0.507753    182.917801      -746
179.669116   7.958702    180.825272      -790
```

Fig 3.5.4c Create new data frame with selected features

3.5.5 Data Visualizations

Visualization of the data is then carried out and analysed to get a better understanding of the relationship between the features and to extract the trend. The histograms are plotted to visualize of each feature and their counts. Then, the visualization of 3D scatter plot is used on the features of ‘pres0’, ‘kinetic_temperature’, ‘taudust’, ‘tauice’, ‘latitude’ and ‘temp’ to represent the data.

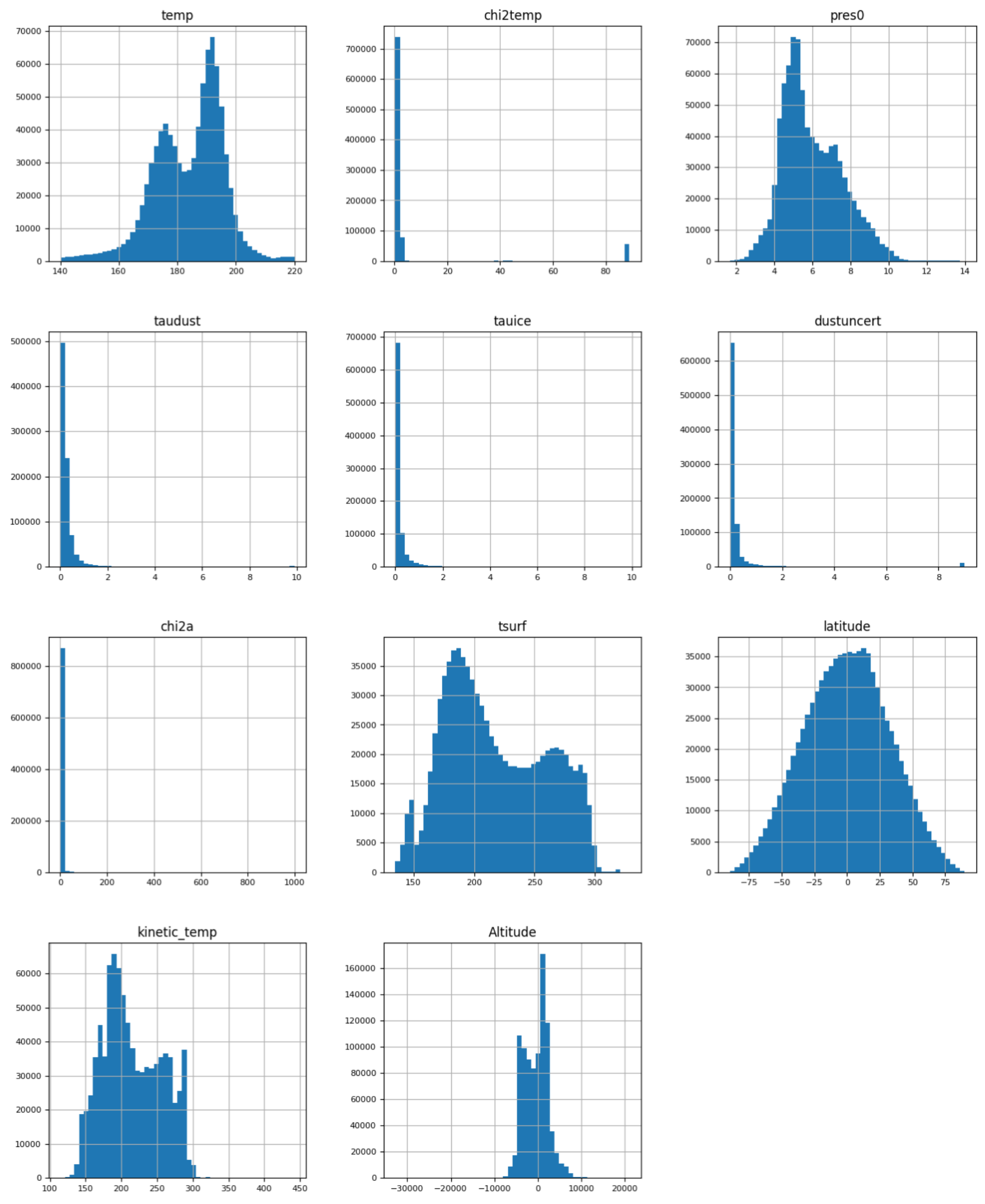


Fig 3.5.5a Histogram to visualize the data count values of each feature

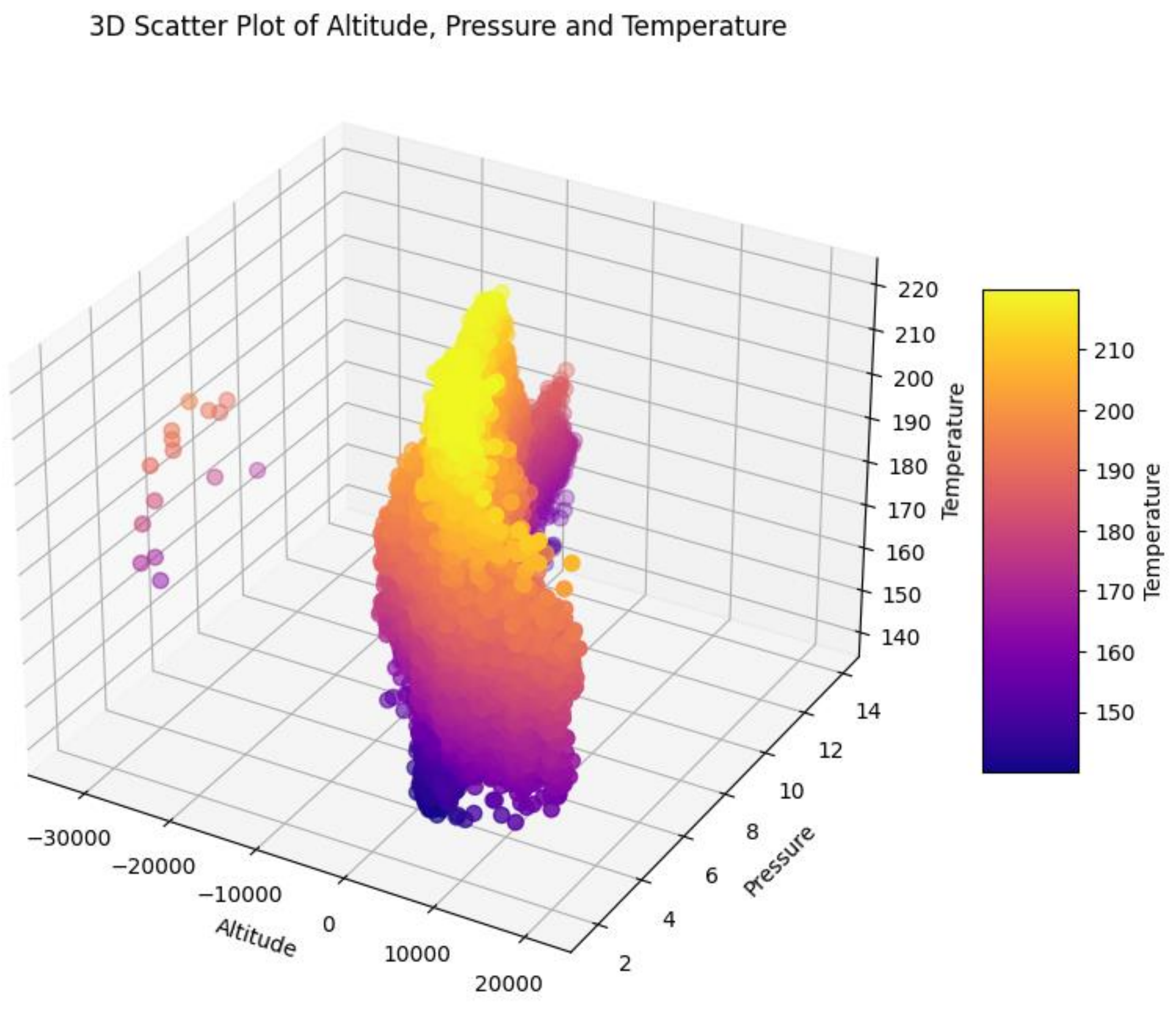


Fig 3.5.5b 3D scatter plot of temperature depending on Altitude and Pressure

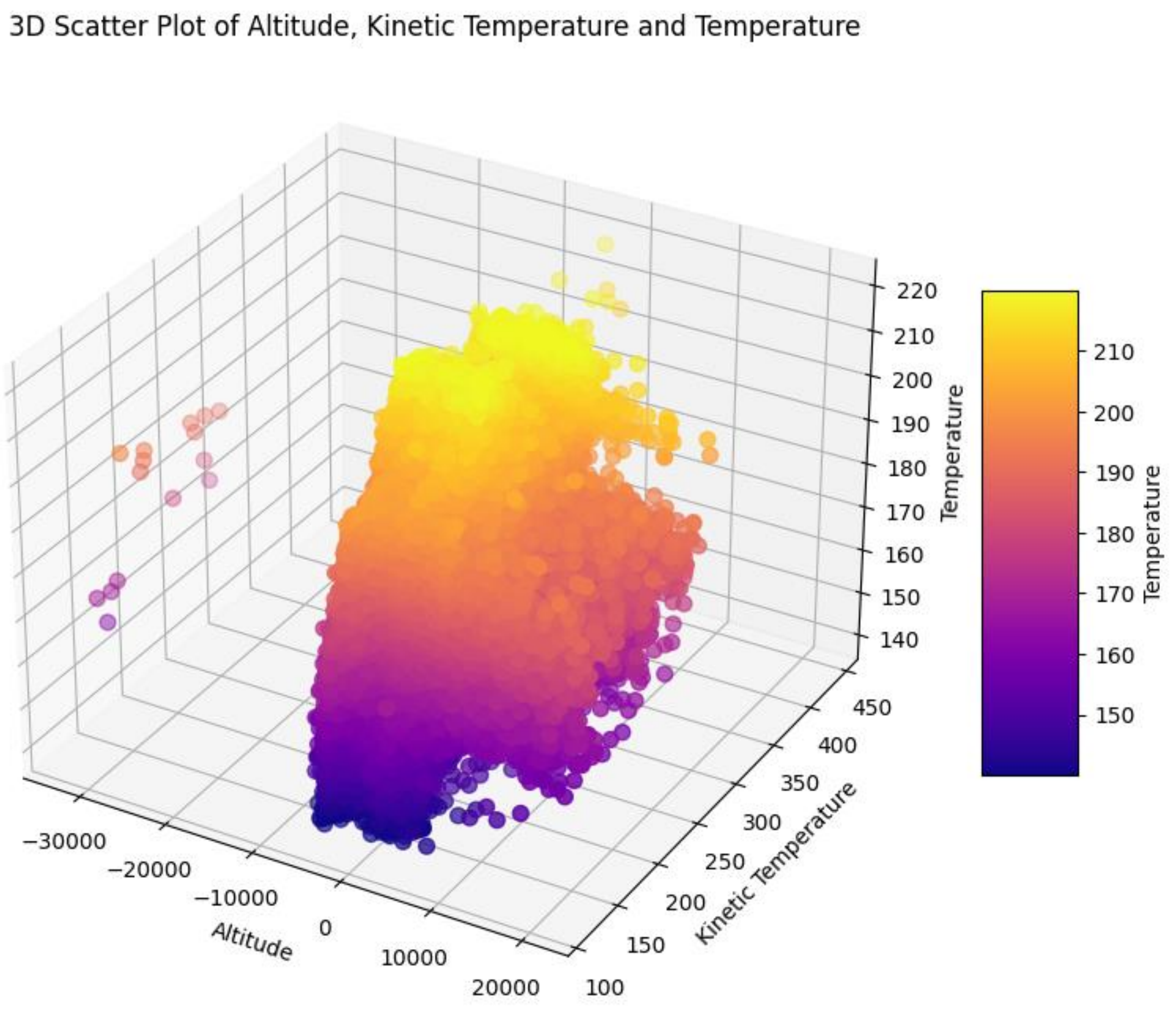


Fig 3.5.5c 3D scatter plot of temperature depending on Altitude and Kinetic Temperature

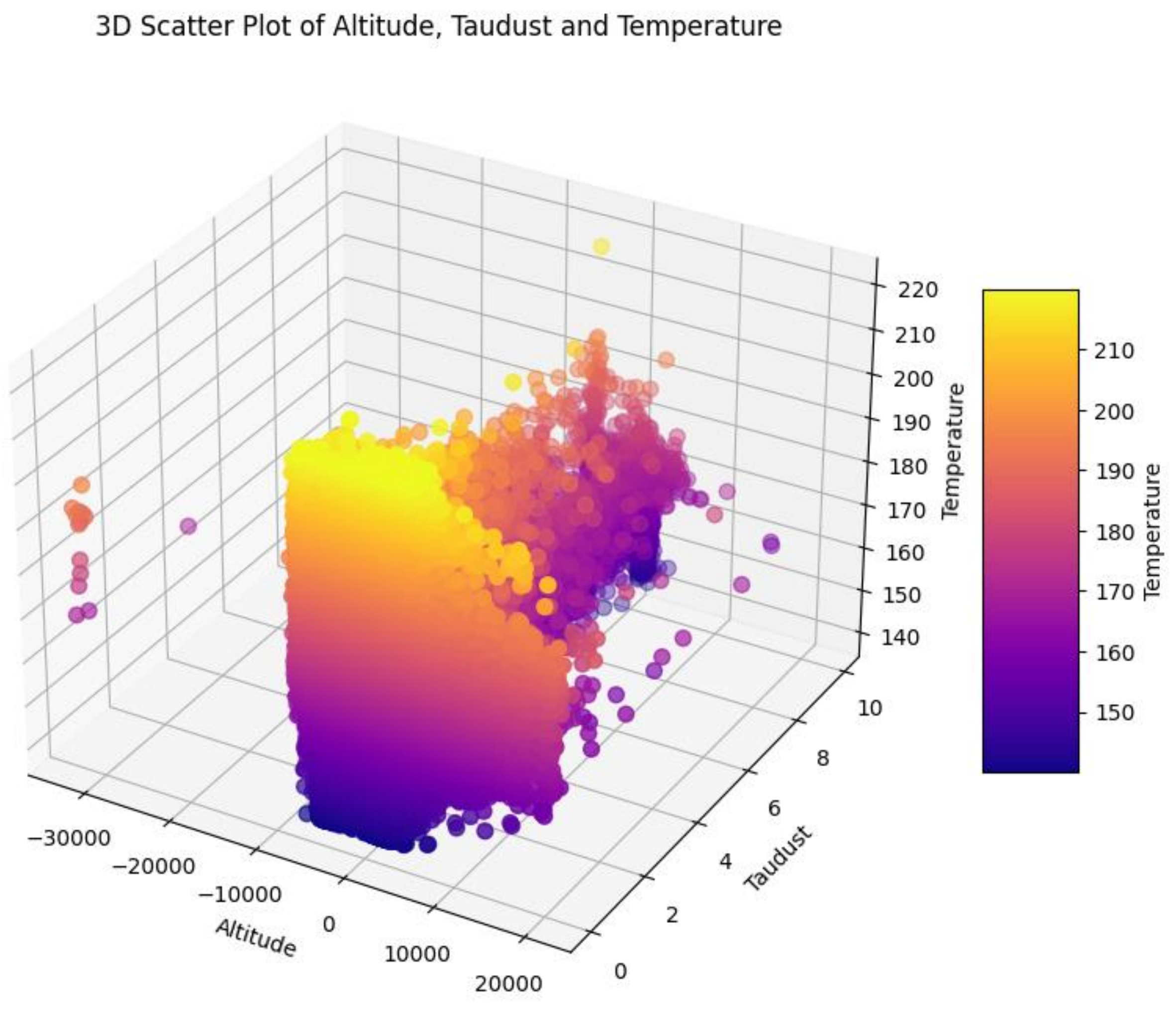


Fig 3.5.5d 3D scatter plot of temperature depending on Altitude and Taudust

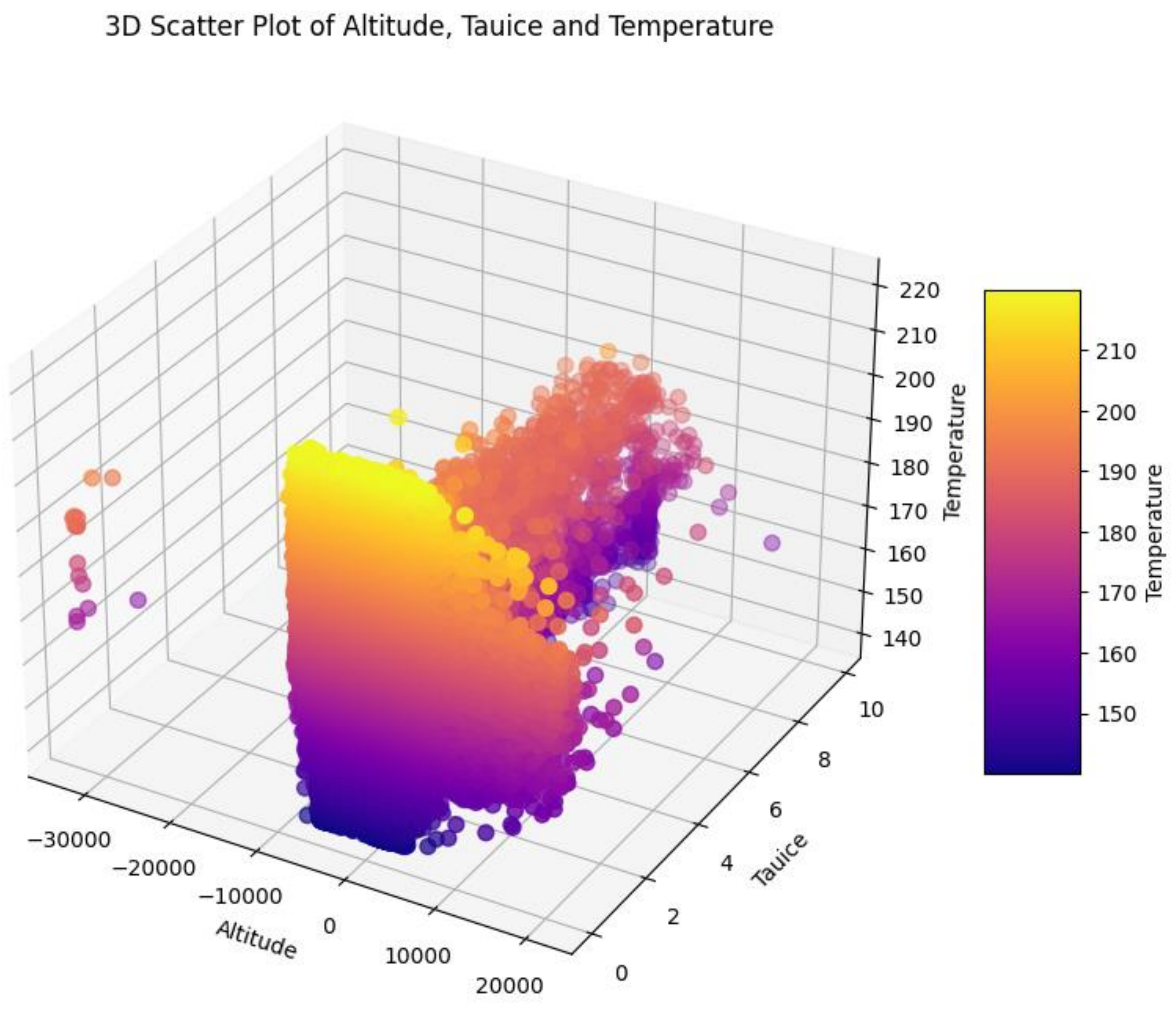


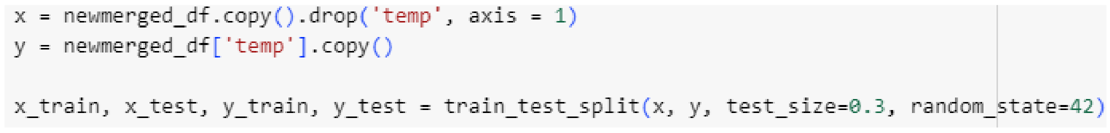
Fig 3.5.5e 3D scatter plot of temperature depending on Altitude and Tauice

3.6 Model Implementation

Six machine learning models, Linear Regression, Gradient Boosting Regressor, K-Nearest Neighbor Regressor, Decision Tree Regressor, Random Forest Regressor and Artificial Neural Network (Multilayer Perceptron) Regressor, were used to predict the surface temperature in this study. The implementation starts off by splitting the data into training and testing data followed by standardizing the data, training the model, testing the model on the test data and evaluating the model. The new data frame is used to split into two portions of 70% for training data and 30% of testing data. After splitting data, those data were standardized to see the potential issues which concerns with the overfitting. Training the models and testing them on the test data and evaluation of the models were done with the help of scikit-learn library.

3.6.1 Data Split

The process of splitting the data into 70% of training data and 30% of testing data with the help of scikit-learn library is shown below in Figure 3.6.2.

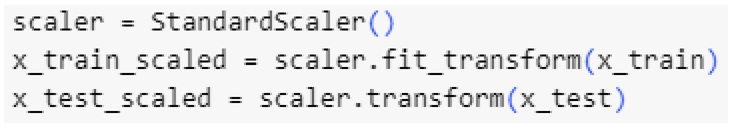


```
x = newmerged_df.copy().drop('temp', axis = 1)
y = newmerged_df['temp'].copy()

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=42)
```

Fig 3.6.1 Data Splitting on new data frame with selected features

3.6.2 Data Standardization

The process of standardizing with the help of scikit-learn library is shown below in Figure 3.6.3.

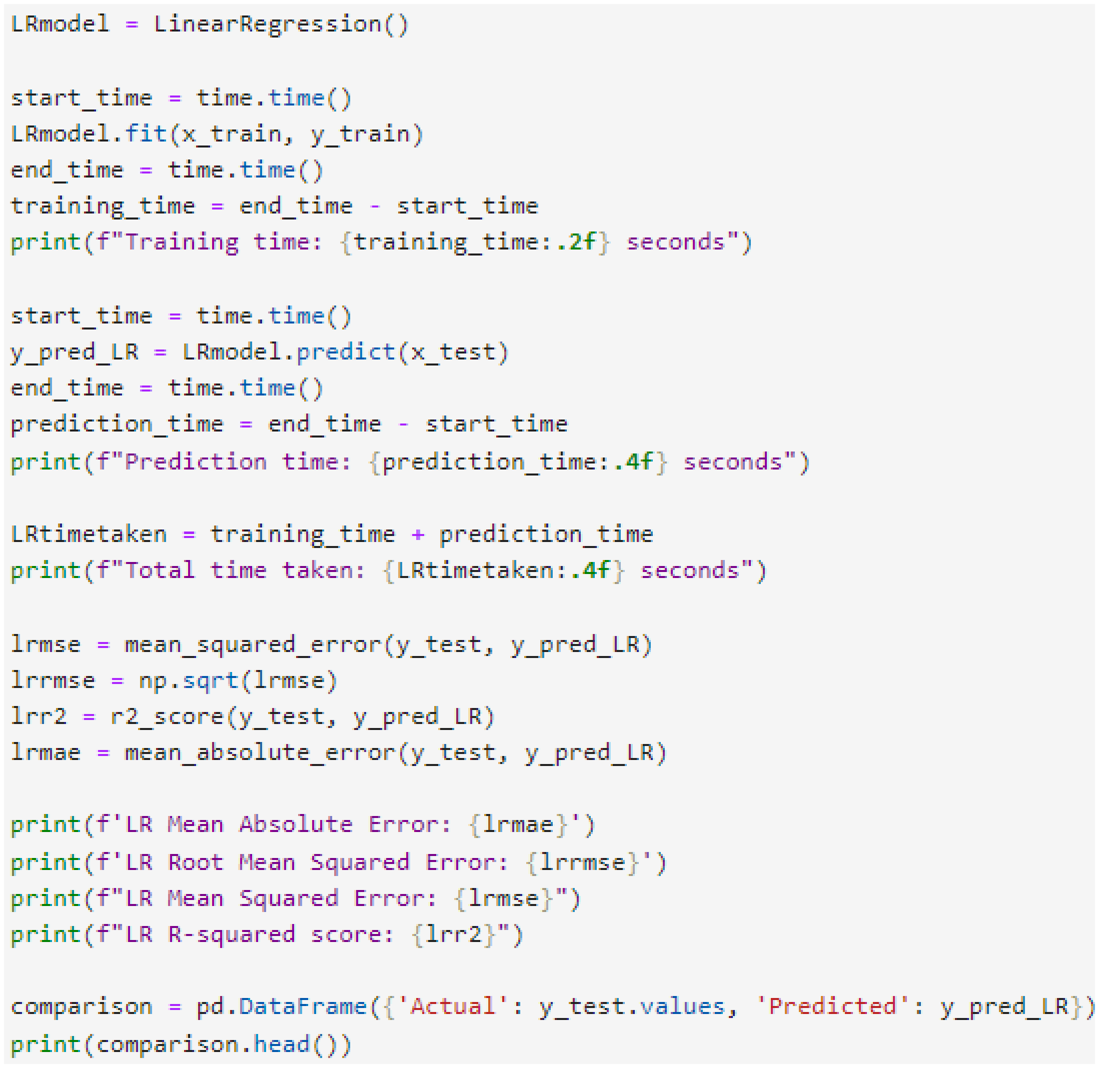


```
scaler = StandardScaler()
x_train_scaled = scaler.fit_transform(x_train)
x_test_scaled = scaler.transform(x_test)
```

Fig 3.6.2 Data Standardization

3.6.3 Linear Regression

3.6.3.1 Model Development

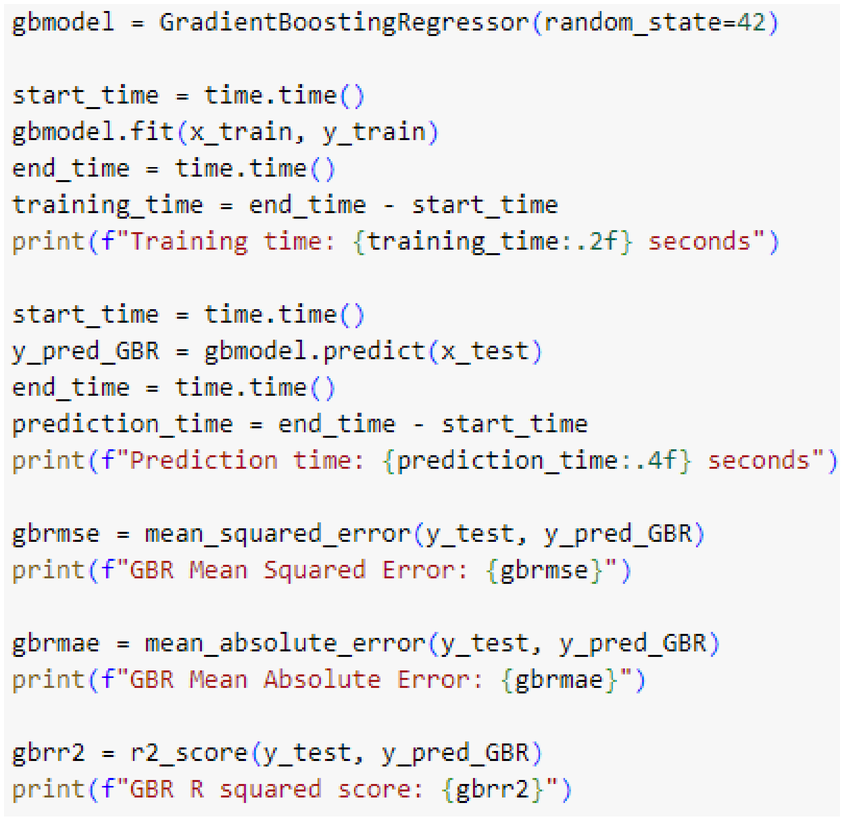


```
LRmodel = LinearRegression()

start_time = time.time()
LRmodel.fit(x_train, y_train)
end_time = time.time()
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")

start_time = time.time()
y_pred_LR = LRmodel.predict(x_test)
end_time = time.time()
prediction_time = end_time - start_time
print(f"Prediction time: {prediction_time:.4f} seconds")

LRtimetaken = training_time + prediction_time
print(f"Total time taken: {LRtimetaken:.4f} seconds")

lrmse = mean_squared_error(y_test, y_pred_LR)
lrrmse = np.sqrt(lrmse)
lrr2 = r2_score(y_test, y_pred_LR)
lrmae = mean_absolute_error(y_test, y_pred_LR)

print(f'LR Mean Absolute Error: {lrmae}')
print(f'LR Root Mean Squared Error: {lrrmse}')
print(f"LR Mean Squared Error: {lrmse}")
print(f"LR R-squared score: {lrr2}")

comparison = pd.DataFrame({'Actual': y_test.values, 'Predicted': y_pred_LR})
print(comparison.head())
```

Fig 3.6.3.1 Linear Regression model development

3.6.4 Gradient Boosting Regressor

3.6.4.1 Model Development

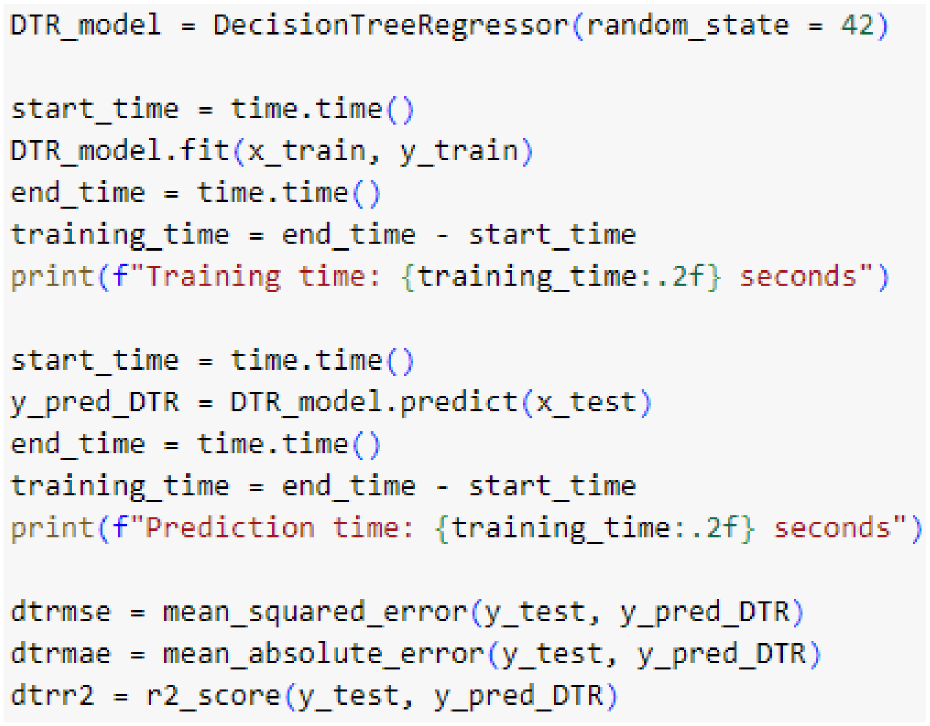


```
gbmodel = GradientBoostingRegressor(random_state=42)

start_time = time.time()
gbmodel.fit(x_train, y_train)
end_time = time.time()
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")

start_time = time.time()
y_pred_GBR = gbmodel.predict(x_test)
end_time = time.time()
prediction_time = end_time - start_time
print(f"Prediction time: {prediction_time:.4f} seconds")

gbrmse = mean_squared_error(y_test, y_pred_GBR)
print(f"GBR Mean Squared Error: {gbrmse}")

gbrmae = mean_absolute_error(y_test, y_pred_GBR)
print(f"GBR Mean Absolute Error: {gbrmae}")

gbrr2 = r2_score(y_test, y_pred_GBR)
print(f"GBR R squared score: {gbrr2}")
```

Fig 3.6.4.1 Gradient Boosting Regressor model development

3.6.5 Decision Tree Regressor

3.6.5.1 Model Development

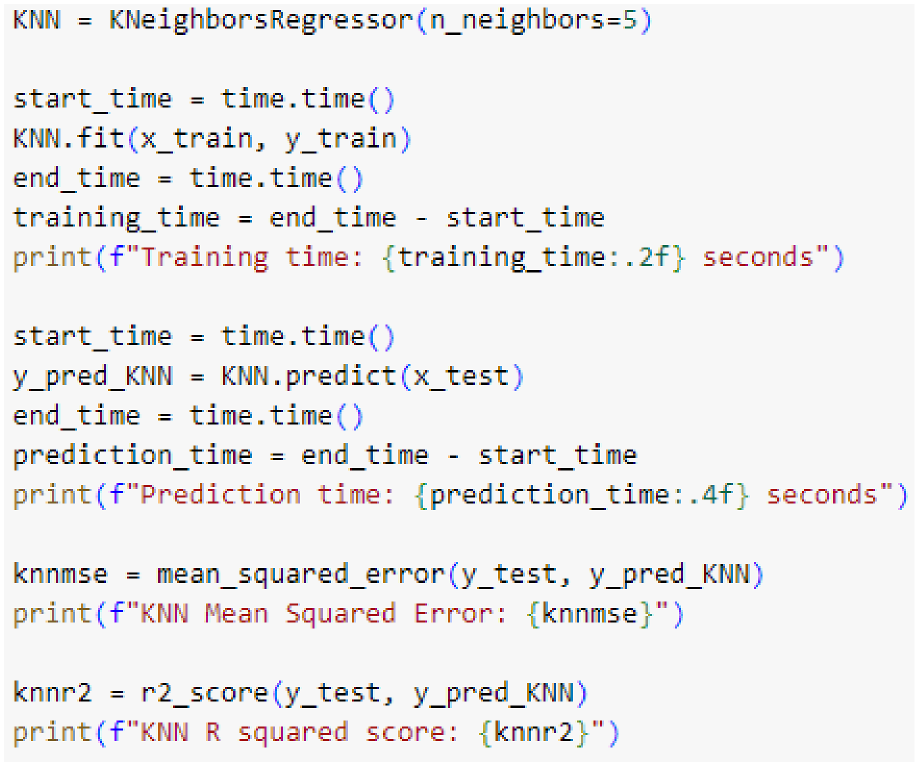


```
DTR_model = DecisionTreeRegressor(random_state = 42)

start_time = time.time()
DTR_model.fit(x_train, y_train)
end_time = time.time()
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")

start_time = time.time()
y_pred_DTR = DTR_model.predict(x_test)
end_time = time.time()
training_time = end_time - start_time
print(f"Prediction time: {training_time:.2f} seconds")

dtrmse = mean_squared_error(y_test, y_pred_DTR)
dtrmae = mean_absolute_error(y_test, y_pred_DTR)
dtrr2 = r2_score(y_test, y_pred_DTR)
```

Fig 3.6.5.1 Decision Tree Regressor model development

3.6.6 K-Neighbors Regressor

3.6.6.1 Model Development

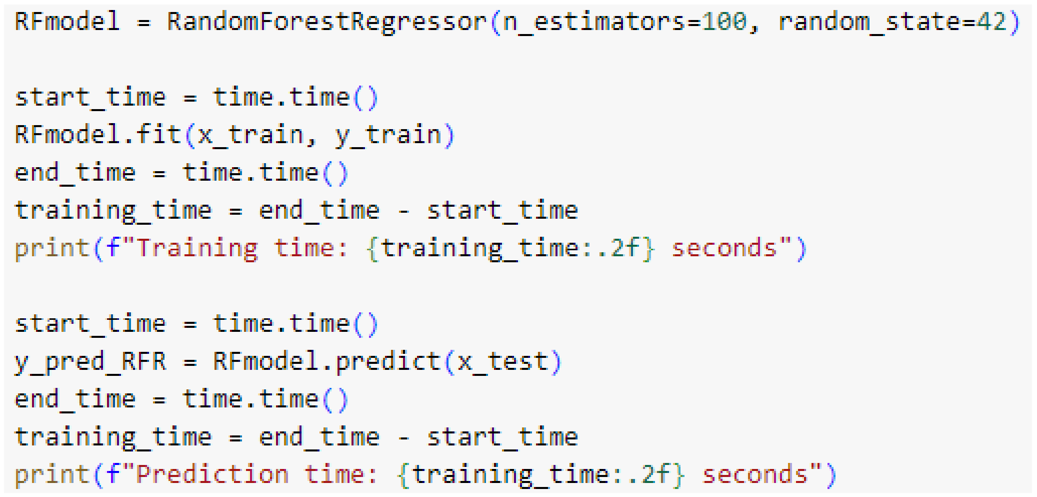


```
KNN = KNeighborsRegressor(n_neighbors=5)

start_time = time.time()
KNN.fit(x_train, y_train)
end_time = time.time()
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")

start_time = time.time()
y_pred_KNN = KNN.predict(x_test)
end_time = time.time()
prediction_time = end_time - start_time
print(f"Prediction time: {prediction_time:.4f} seconds")

knnmse = mean_squared_error(y_test, y_pred_KNN)
print(f"KNN Mean Squared Error: {knnmse}")

knnr2 = r2_score(y_test, y_pred_KNN)
print(f"KNN R squared score: {knnr2}")
```

Fig 3.6.6.1 KNeighbors Regressor model development

3.6.7 Random Forest Tree Regression

3.6.7.1 Model Development

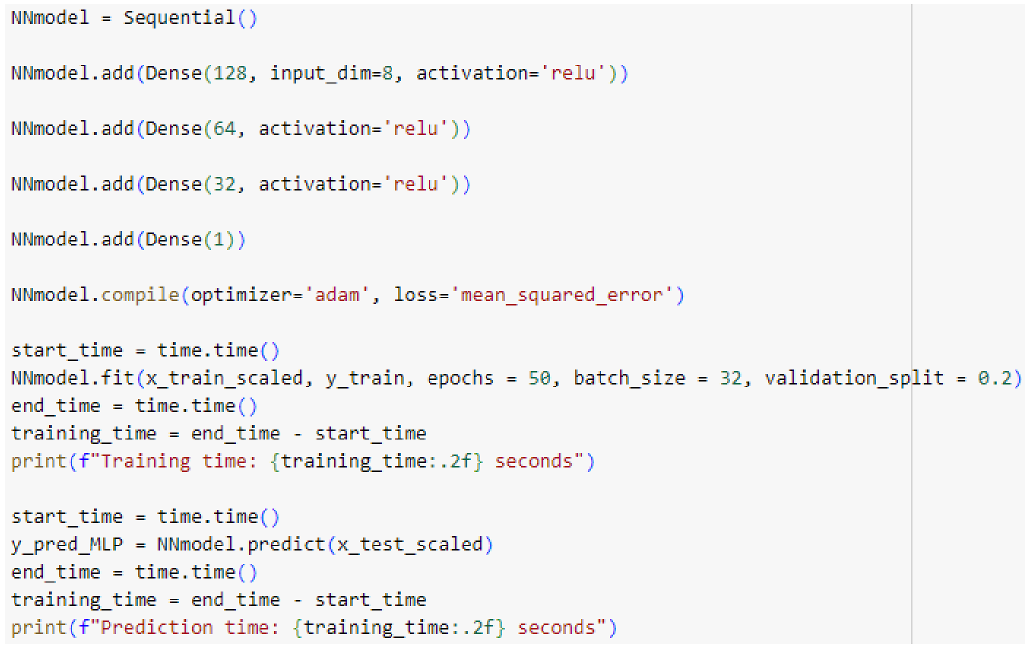


```
RFmodel = RandomForestRegressor(n_estimators=100, random_state=42)

start_time = time.time()
RFmodel.fit(x_train, y_train)
end_time = time.time()
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")

start_time = time.time()
y_pred_RFR = RFmodel.predict(x_test)
end_time = time.time()
training_time = end_time - start_time
print(f"Prediction time: {training_time:.2f} seconds")
```

Fig 3.6.7.1 Random Forest Tree Regression model development

3.6.8 Neural Network Regression (Multilayer Perceptron)

3.6.8.1 Model Development



```
NNmodel = Sequential()

NNmodel.add(Dense(128, input_dim=8, activation='relu'))

NNmodel.add(Dense(64, activation='relu'))

NNmodel.add(Dense(32, activation='relu'))

NNmodel.add(Dense(1))

NNmodel.compile(optimizer='adam', loss='mean_squared_error')

start_time = time.time()
NNmodel.fit(x_train_scaled, y_train, epochs = 50, batch_size = 32, validation_split = 0.2)
end_time = time.time()
training_time = end_time - start_time
print(f"Training time: {training_time:.2f} seconds")

start_time = time.time()
y_pred_MLP = NNmodel.predict(x_test_scaled)
end_time = time.time()
training_time = end_time - start_time
print(f"Prediction time: {training_time:.2f} seconds")
```

Fig 3.6.8.1 Neural Network Regression (MLP) model development

Chapter 4 – Model Evaluation

To evaluate the models, the following metrics of different evaluation parameters were used.

Mean Absolute Error (MAE) - Mean Absolute Error (MAE) can be considered as the sum of the absolute value different between every expected value and predicted value. Priyadarshini and Puri 2021 defined the Mean Absolute Error as ‘the difference between the actual and predicted values obtained by averaging the absolute difference over the dataset’.

$$\mathrm{MAE} = \frac{1}{N} \sum\nolimits_{i=1}^{N} |y_i - \hat{y}|$$

Equation 4a Mean Absolute Error (Priyadarshini and Puri 2021)

Mean Squard Error (MSE) - Mean Squared Error (MSE) is applied to check the difference between real values and predicted values by squaring the average difference over the dataset (Priyadarshini and Puri 2021). The lower the value of Mean Squared Error, the better the model is performing.

$$\mathrm{MSE} = \frac{1}{N} \sum\nolimits_{i=1}^{N} (y_i - \hat{y})^2$$

Equation 4b Mean Squared Error (Priyadarshini and Puri 2021)

Root Mean Squared Error (RMSE) - Root Mean Squared Error can be easily obtained by performing square root to the mean squared error. This is a standard measuring method to check the error of a model when predicting quantitative data (Priyadarshini and Puri 2021). The same concept with the Mean Squared Error, the lower the value of the Root Mean Squared Error, the better the model is predicting.

$$\mathrm{RMSE} = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(y_i - \hat{y})^2}$$

Equation 4c Root Mean Squared Error (Priyadarshini and Puri 2021)

Coefficient of Determination (R-squared) - R squared is used to verify the accuracy of the predictive model (Priyadarshini and Puri 2021). The R squared value is usually calculated between 0 to 1 where 0 has the lowest accuracy and 1 has the highest.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - y)^2}$$

Equation 4d R-squared Score (Priyadarshini and Puri 2021)

Time Taken to Train and Predict the Models – This factor is considered to be a crucial factor when evaluating different models as it gives the idea of how fast the model can perform to train and predict from the models. The time taken values are shown in seconds.

Chapter 5 – Results and Discussion

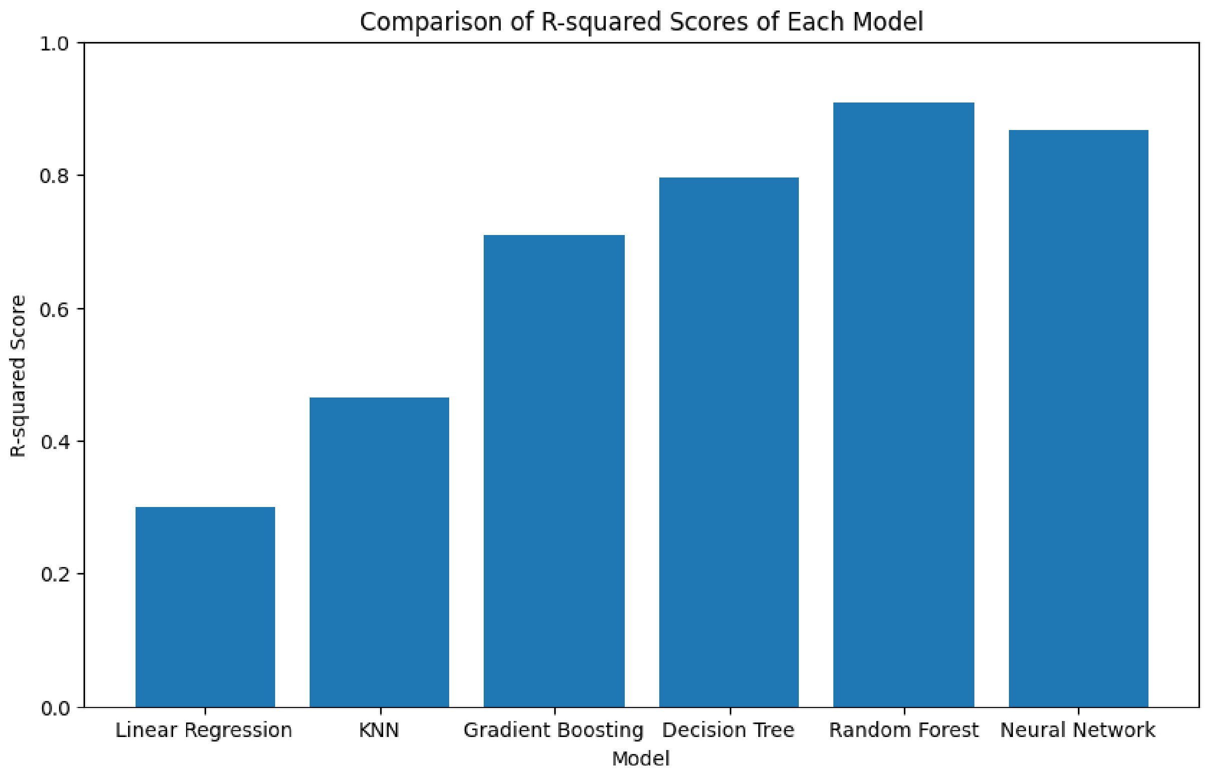
5.1 Results

The models are evaluated after the development to check their performance on the dataset in terms of the evaluation variables as mentioned in chapter 4. Furthermore, these results from each model were compared and critically analysed. The values of the evaluation variables of each model can be seen in Table 5.1 below.

Model Name	**MAE**	**MSE**	**RMSE**	**R2**	**TIMETAKEN (Seconds)**
Linear Regression	7.90	97.00	9.85	0.30	0.67
K-Nearest Neighbours Regressor	6.71	74.01	8.60	0.47	15.42
Gradient Boosting	4.79	40.15	6.34	0.71	384.80
Decision Tree	3.50	28.22	5.31	0.80	23.04
Random Forest Regressor	2.45	12.67	3.56	0.91	1487.73
Neural Network (Multilayer Perceptron) Regressor	2.97	18.50	4.30	0.87	917.17

Table 5.1 Matrix table for model evaluation scores

From this table the performance of the models can be analyzed and verified which model would be the most suitable for the dataset used for this research. Then, the bar graphs were generated to visualize and compare the individual score category of every model. These new visualization graphs of the evaluations are shown below.



Fig

5.1a Comparison of R-squared Scores of the models

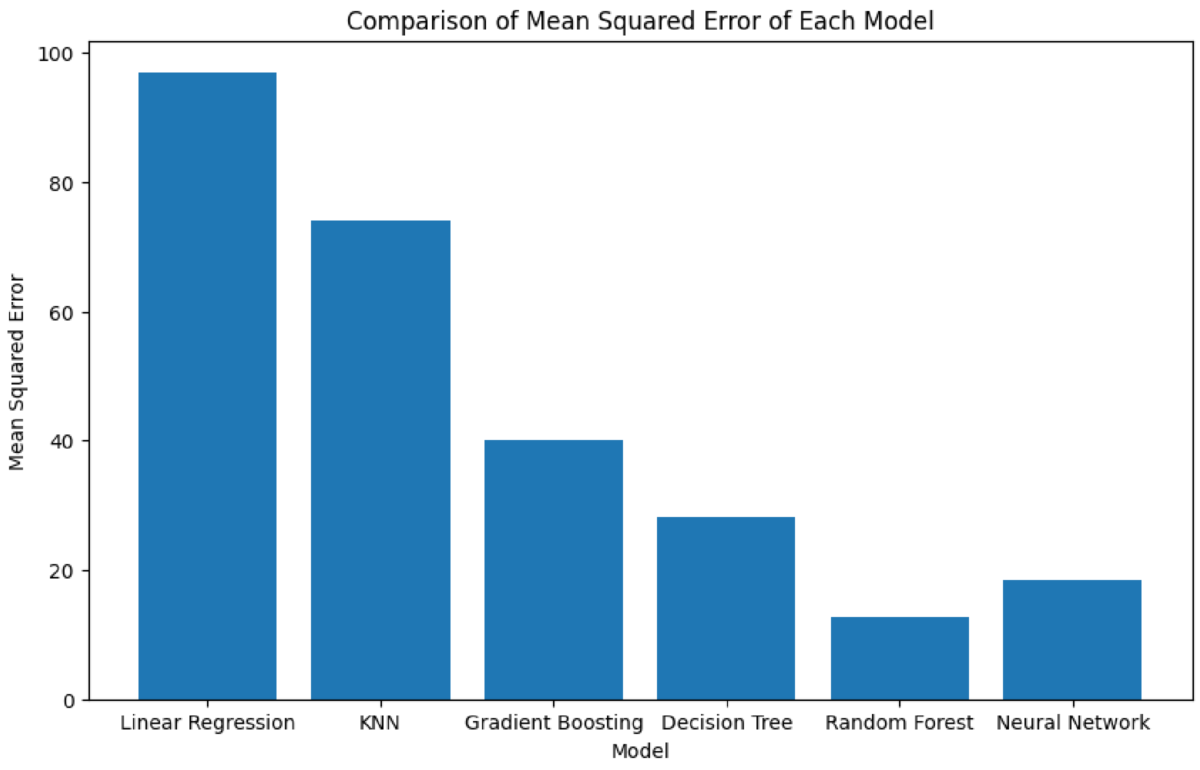


Fig 5.1b Comparison of Mean Squared Error of the models

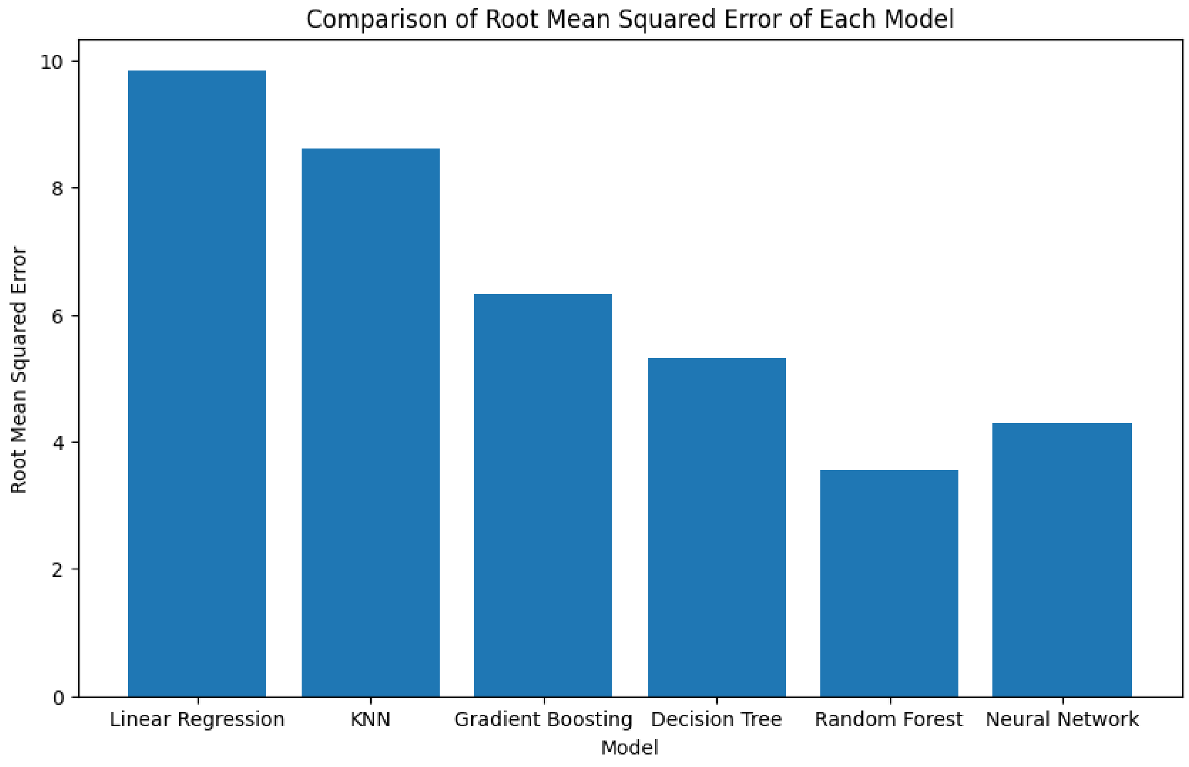


Fig 5.1c Comparison of Root Mean Squared Error of the models

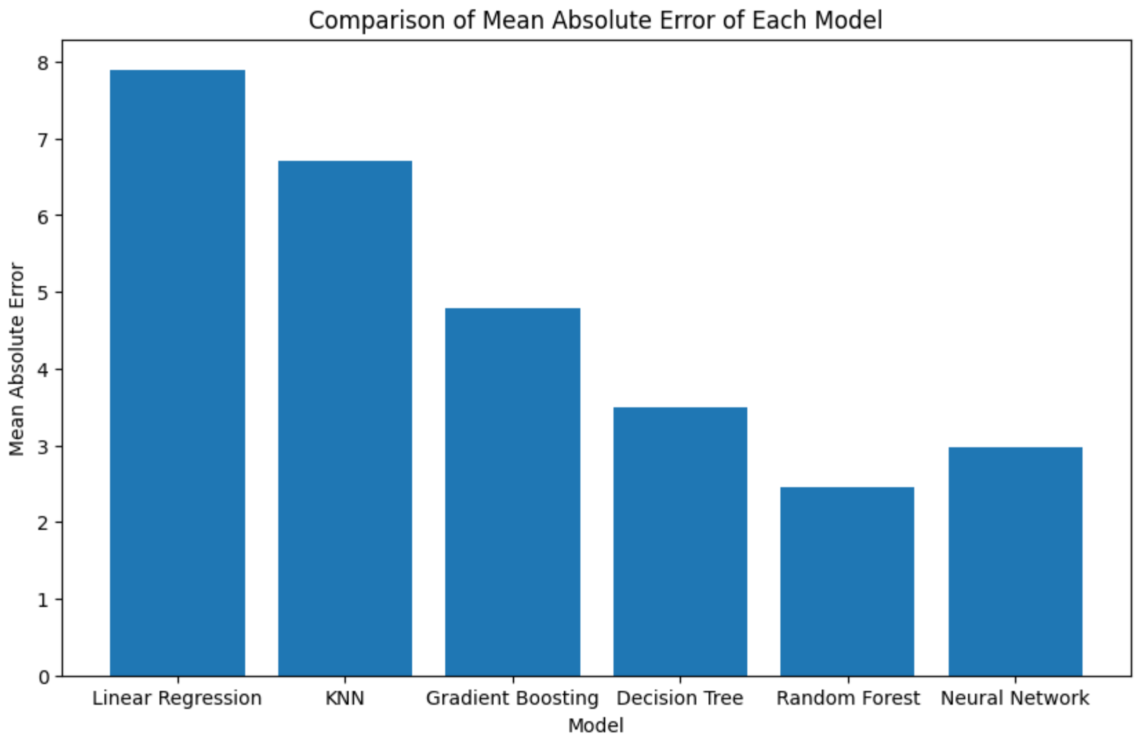


Fig 5.1d Comparison of Mean Absolute Error of the models

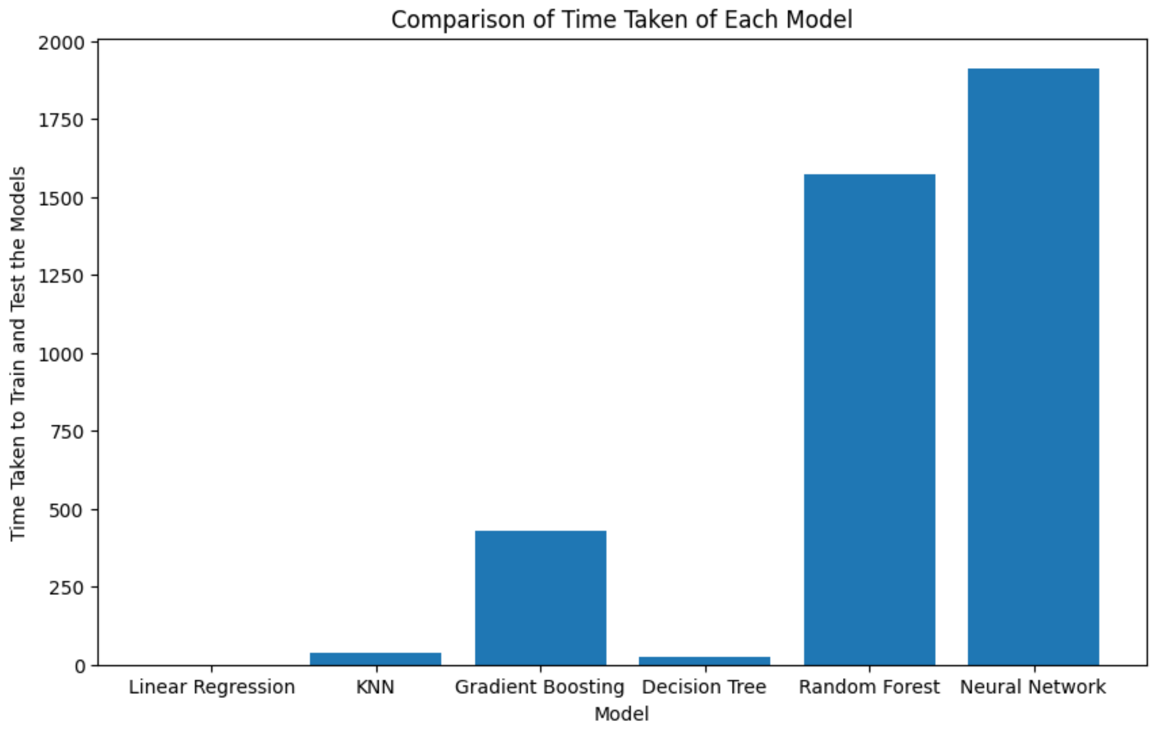


Fig 5.1e Comparison of Time Taken to train and predict from the models

5.2 Discussion

After analysing the model evaluation matrix table, it was clear that the R-squared score of the Linear Regression model is significantly lower than that of the other five models with 0.30. On the other hand, the Linear Regression model which Piyush Pant et al. (2023) applied in their research has the accuracy of 85 percent. Nevertheless, it is notably that this is due to the size of the dataset and the correlation of the features used in the research. The data used in Piyush Pant et al. (2023) paper, obtained from the Mars Science Laboratory (MSL) rover's Rover Environmental Monitoring Station (REMS), has the 1867 count of instances with five features where the dataset used in this research contained more than 874000 instances with eleven features. Due to this reason, it was obvious, since before the implementation, that Linear Regression model would not perform well on the dataset used in this report because of the complexity of the relation between each feature and the number of instances contained in the dataset. However, the main reason to apply this model in this research is to simply observe the mean baseline so that the performance of other models can be compared and evaluated.

Among other five models, apart from Linear Regression, Random Forest Regression has the best accuracy with the R-squared score of 0.91. Then the remaining four models, Artificial Neural Network (Multilayer Perceptron), Decision Tree Regression, Gradient Boosting Regressor and K-Nearest Neighbor Regressor have the R-squared score of 0.87, 0.80, 0.71, 0.47 respectively. The values of the Mean Squared Error of the models are calculated as shown in Figure 5.1b above showing that the Random Forest Regressor has the lowest value, and the Linear Regression with the highest of 12.67 and 97 respectively. The Mean Squared error of the Artificial Neural Network (MLP) has the second lowest value with 18.50 followed by Decision Tree Regressor, Gradient Boosting Regressor and K-Nearest Neighbor Regressor with 28.22, 40.15, 74.01 respectively. In addition, the values of the Root Mean Squared Error and Mean Absolute Error are also generated to compare and can be observed in Figure 5.1c and Figure 5.1d.

Subsequently after the comparison analysis on the evaluation parameters of R-squared Score, Mean Squared Error, Root Mean Squared Error and Mean Absolute Error, there is no doubt that the Random Forest Regressor is the best prediction model for predicting the surface temperature on the features selected in this research. Nevertheless, there is one more factor to consider when evaluating the models and that is the time taken by the model to train and predict. According to the evaluation matrix and the bar chart (Figure 5.1e) illustrating the comparison of the time taken by the model, Linear Regression which has the lowest value of R-squared score has the lowest duration of less than one second (0.67 second) to train and predict the data. The model which consumes the longest duration of time is Random Forest Regressor which takes 1487.73 seconds followed by the Neural Network (MLP) Regressor which takes 917.17 seconds to train the predict the data. Gradient Boosting Regressor, Decision Tree Regressor and K-Nearest Neighbor Regressor takes 384.80, 23.04 and 15.42 seconds respectively.

Chapter 6 - Conclusion, Recommendations and Future Works

6.1 Conclusion

The data collected from the EMM EMIRS instrument contained a large number of instances with numerous features. With a good data cleaning process and correct feature selection, different targets can be achieved. However, as mentioned in Chapter 1, the purpose of this research is to develop the best algorithms to predict the surface temperature depending on the variation of the altitude feature extracted from the latitude and longitude and other correlated features collected from EMIRS instrument using rasterio library. As the conclusion of this research, the Random Forest Regressor model can produce the best accuracy score of predicting the surface temperature depending on the altitude, pressure and another nine features which are carefully chose after critically analysing the relationship between each feature to the surface temperature by using various methods such as feature selection, correlation analysis and domain knowledge. By using the model presented in this paper, future researchers or space enthusiasts can easily implement it to predict the surface temperature.

6.2 Recommendation and Future Works

This research developed a model which can predict the surface temperature on different altitudes with the features selected by using feature selection and checking the correlation matrix. However, the first recommendation to improve the model's performance for future researchers to implement, is to perform a good data sampling technique which is necessary when carrying out the advanced computational problem caused by extensive data. The second recommendation for this research will be applying advanced data preprocessing methods which is also crucial when calculating advanced computation problem. In addition, to perform parallel processing and using advanced computing resources, especially during the implementation of advanced and complicated machine learning algorithms like Artificial Neural Networks, is also one of the highest recommendations to improve this research to another level.

As for the future works for the future space enthusiasts who’s implementing this process carried out in this paper, they can try using more advanced and complicated models as mentioned above such as Artificial Neural Networks along with applying fine-tune models using effective hyperparameter optimization techniques like Bayesian optimization or randomized search to see the improvement of the model’s performance. Implementing different technique of data splitting such as 50% training and 50% testing could produce an effective and meaningful insights which can be considered as another future work.

References

- Al-Saad, M., Aburaed, N., Al Mansoori, S., Mansoor, W. and Al-Ahmad, H. (2022) A Study on Deep Learning Approaches for Mars Weather Forecasting. *2022 5th International Conference on Signal Processing and Information Security (ICSPIS)* IEEE76–79.
- Amiri, H.E.S., Brain, D., Sharaf, O., Withnell, P., McGrath, M., Alloghani, M., Awadhi, M.A., Dhafri, S.A., Hamadi, O.A., Matroushi, H.A., Shamsi, Z.A., Shehhi, O.A., Chaffin, M., Deighan, J., Edwards, C., Ferrington, N., Harter, B., Holsclaw, G., Kelly, M., Kubitschek, D., Landin, B., Lillis, R., Packard, M., Parker, J., Pilinski, E., Pramman, B., Reed, H., Ryan, S., Sanders, C., Smith, M., Tomso, C., Wrigley, R., Mazmi, H.A., Mheiri, N.A., Shamsi, M.A., Tunaiji, E.A., Badri, K., Christensen, P., England, S., Fillingim, M., Forget, F., Jain, S., Jakosky, B.M., Jones, A., Lootah, F., Luhmann, J.G., Osterloo, M., Wolff, M. and Yousuf, M. (2022) The Emirates Mars Mission. *Space Science Reviews* 218 (1), 4.
- Anon (n.d.) *EMM Science Data Center*. https://sdc.emiratesmarsmission.ae/documentation Accessed 10 September 2024.
- Atwood, S.A., Smith, M.D., Badri, K., Edwards, C.S., Christensen, P.R., Wolff, M.J., Forget, F., Anwar, S., Smith, N. and El-Maarry, M.R. (2022) Diurnal Variability in EMIRS Daytime Observations of Water Ice Clouds During Mars Aphelion-Season. *Geophysical Research Letters* 49 (15), .
- Cai, J., Xu, K., Zhu, Y., Hu, F. and Li, L. (2020) Prediction and analysis of net ecosystem carbon exchange based on gradient boosting regression and random forest. *Applied Energy* 262, 114566.
- Edwards, C.S., Christensen, P.R., Mehall, G.L., Anwar, S., Tunaiji, E.A., Badri, K., Bowles, H., Chase, S., Farkas, Z., Fisher, T., Janiczek, J., Kubik, I., Harris-Laurila, K., Holmes, A., Lazbin, I., Madril, E., McAdam, M., Miner, M., O'Donnell, W., Ortiz, C., Pelham, D., Patel, M., Powell, K., Shamordola, K., Tourville, T., Smith, M.D., Smith, N., Woodward,

R., Weintraub, A., Reed, H. and Pilinski, E.B. (2021) The Emirates Mars Mission (EMM) Emirates Mars InfraRed Spectrometer (EMIRS) Instrument. *Space Science Reviews* 217 (7), .

- Fan, S., Forget, F., Smith, M.D., Guerlet, S., Badri, K.M., Atwood, S.A., Young, R.M.B., Edwards, C.S., Christensen, P.R., Deighan, J., Al Matroushi, H.R., Bierjon, A., Liu, J. and Millour, E. (2022) Migrating Thermal Tides in the Martian Atmosphere During Aphelion Season Observed by EMM/EMIRS. *Geophysical Research Letters* 49 (18), .
- Gebhardt, C., Guha, B.K., Young, R.M.B. and Wolff, M.J. (2022) A Frontal Dust Storm in the Northern Hemisphere at Solar Longitude 97—An Unusual Observation by the Emirates Mars Mission. *Geophysical Research Letters* 49 (20), .
- Kassim, N.M., Santhiran, S., Alkahtani, A.A., Islam, M.A., Tiong, S.K., Mohd Yusof, M.Y. and Amin, N. (2023) An Adaptive Decision Tree Regression Modeling for the Output Power of Large-Scale Solar (LSS) Farm Forecasting. *Sustainability* 15 (18), 13521.
- Pant, P., Rajawat, A.S., Goyal, S.B., Kemat, B.B., Mihălţan, T.C., Verma, C. and Răboacă, M.S. (2023) Machine Learning Techniques for Analysis of Mars Weather Data. *2023 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)* IEEE1–7.
- Priyadarshini, I. and Puri, V. (2021) Mars weather data analysis using machine learning techniques. *Earth Science Informatics* 14 (4), 1885–1898.
- Singh, U., Rizwan, M., Alaraj, M. and Alsaidan, I. (2021) A Machine Learning-Based Gradient Boosting Regression Approach for Wind Power Production Forecasting: A Step towards Smart Grid Environments. *Energies* 14 (16), 5196.
- Sumayli, A. (2023) Development of advanced machine learning models for optimization of methyl ester biofuel production from papaya oil: Gaussian process regression (GPR), multilayer perceptron (MLP), and K-nearest neighbor (KNN) regression models. *Arabian Journal of Chemistry* 16 (7), 104833.

- Yan, X. and Su, X.G. (2009) Simple Linear Regression. *Linear Regression Analysis* WORLD SCIENTIFIC. 9–39.
- Young, R.M.B., Millour, E., Forget, F., Smith, M.D., Aljaberi, M., Edwards, C.S., Smith, N., Anwar, S., Christensen, P.R. and Wolff, M.J. (2022) First Assimilation of Atmospheric Temperatures From the Emirates Mars InfraRed Spectrometer. *Geophysical Research Letters* 49 (21), .

Appendices

Appendix A

From the results generated from different algorithms proposed in this research, a 3D scatter plot for each model is created. These visualizations can be seen below, and they represent a clear result and comparison of the predicted output and actual output. By looking at these visualizations, the easy understanding of the model's performance can be witnessed.

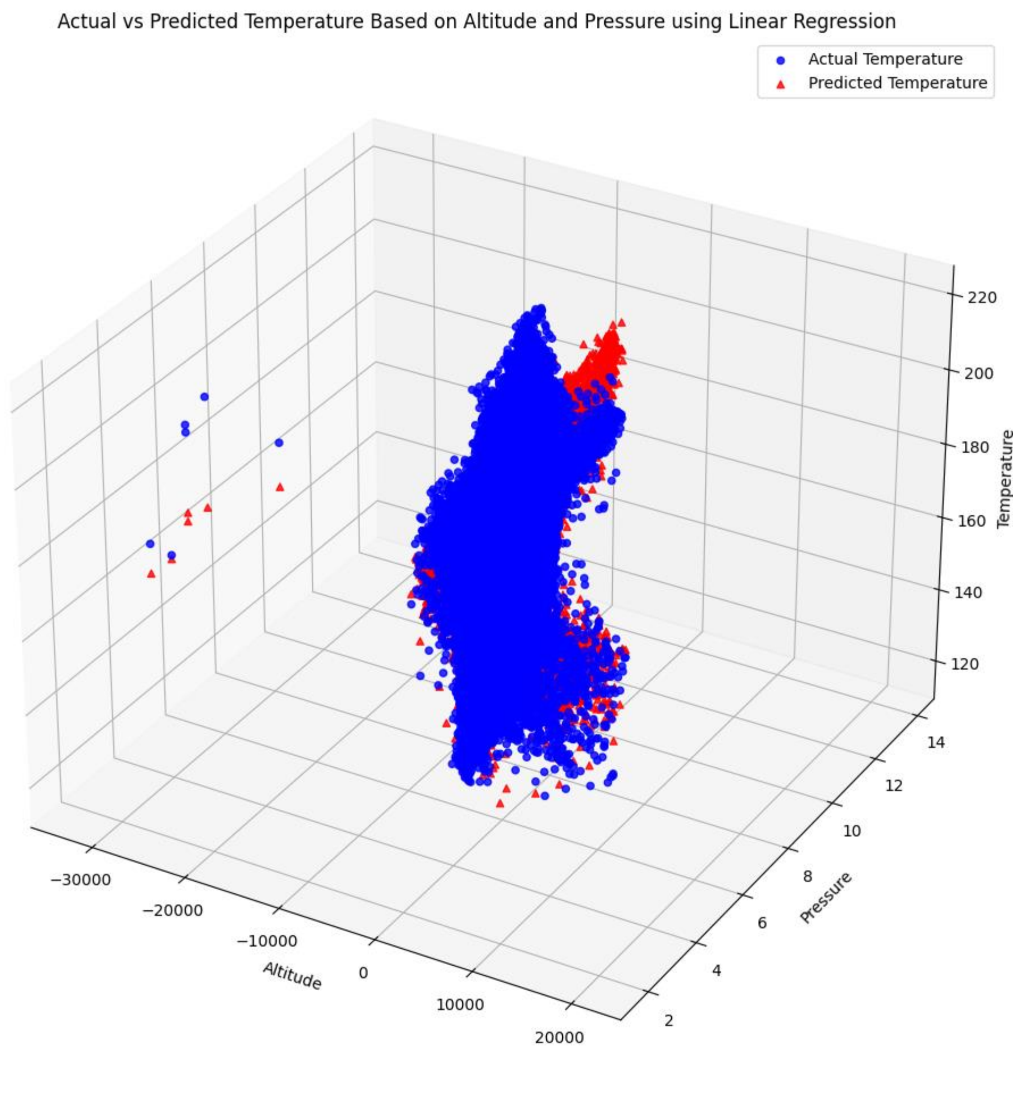


Fig Appendix A.1 Comparison of the actual and predicted temperature based on Altitude and Pressure using Linear Regression

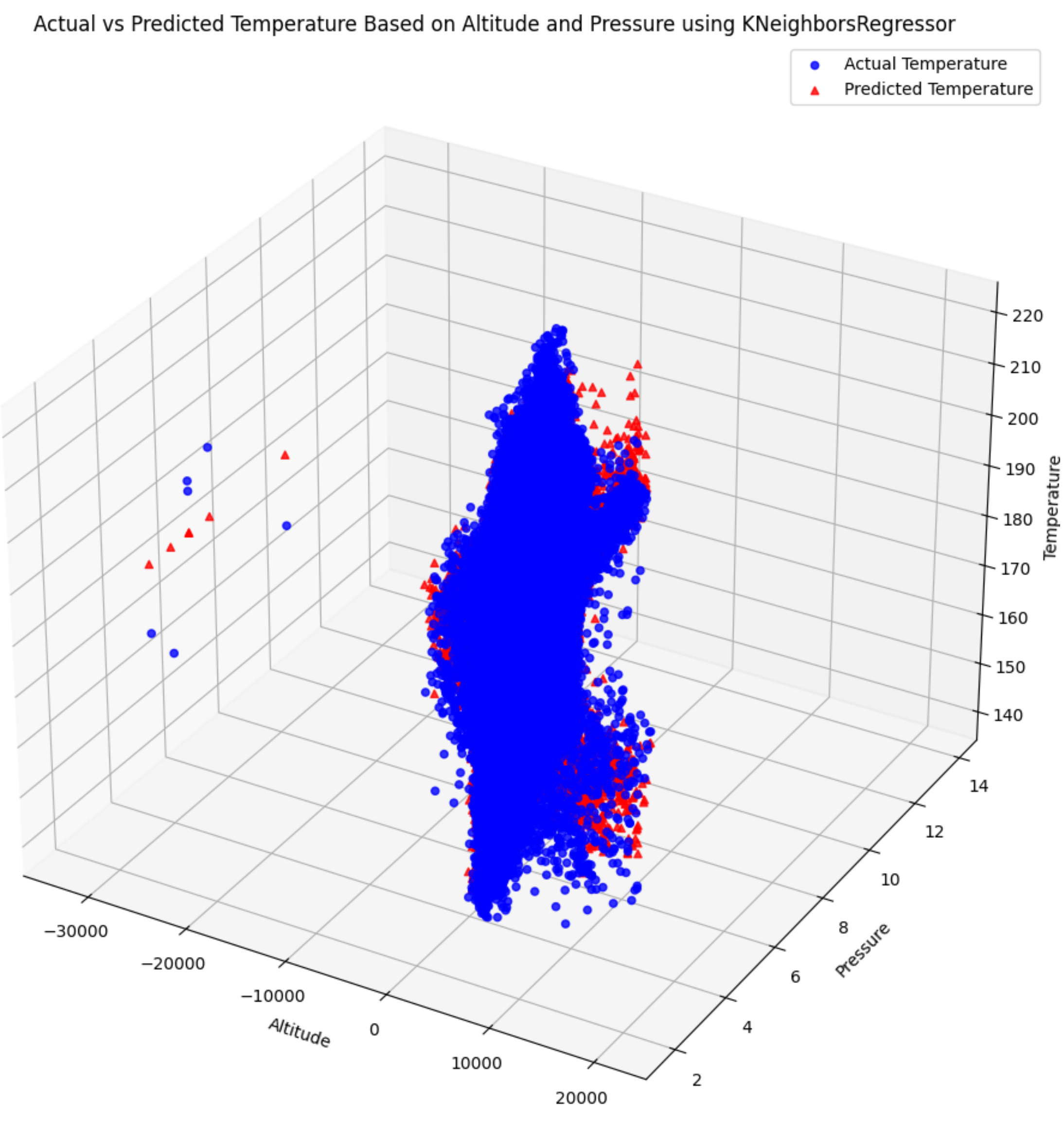


Fig Appendix A.2 Comparison of the actual and predicted temperature based on Altitude and Pressure using K-Nearest Neighbors Regressor

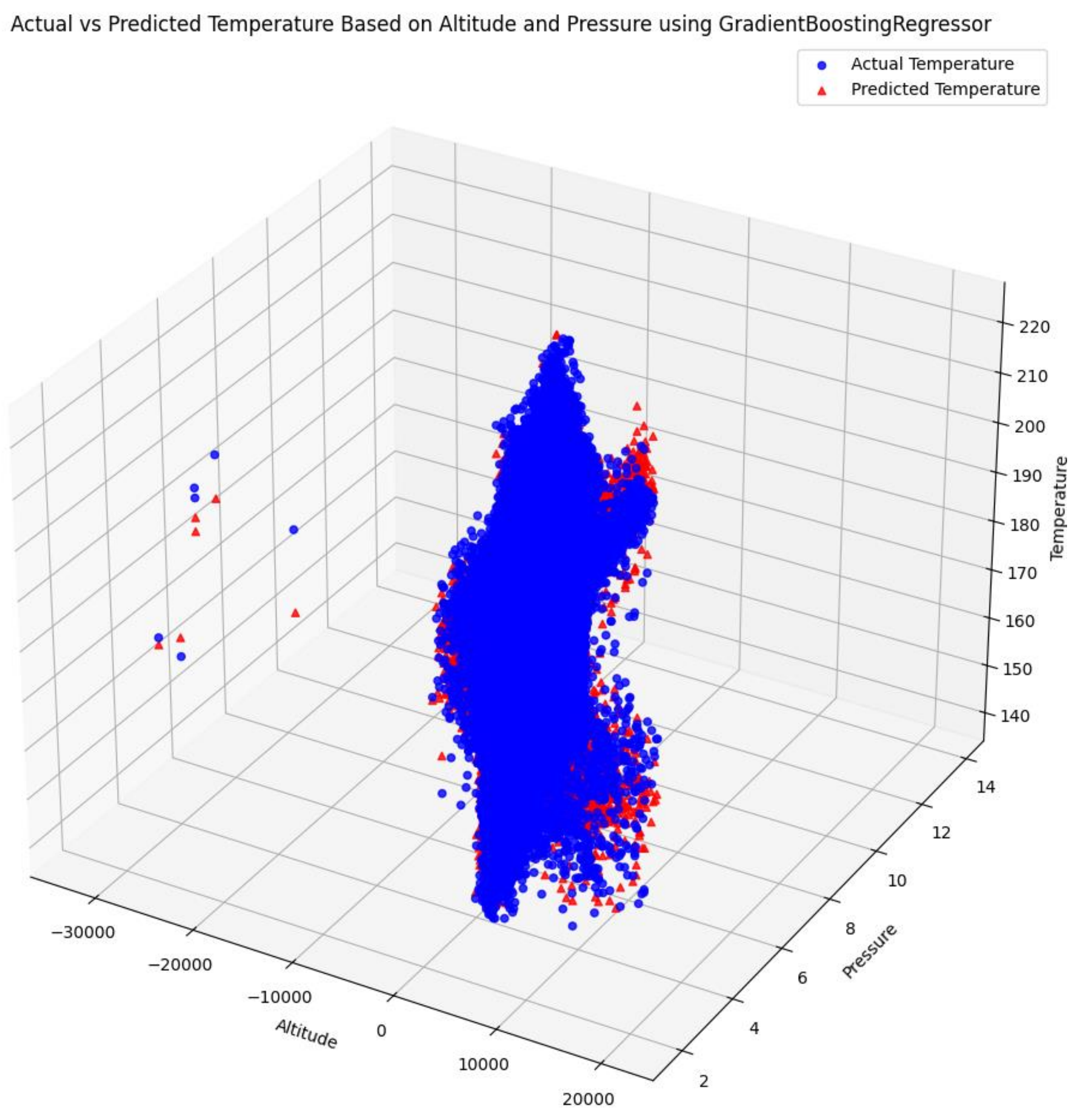


Fig Appendix A.3 Comparison of the actual and predicted temperature based on Altitude and Pressure using Gradient Boosting Regressor

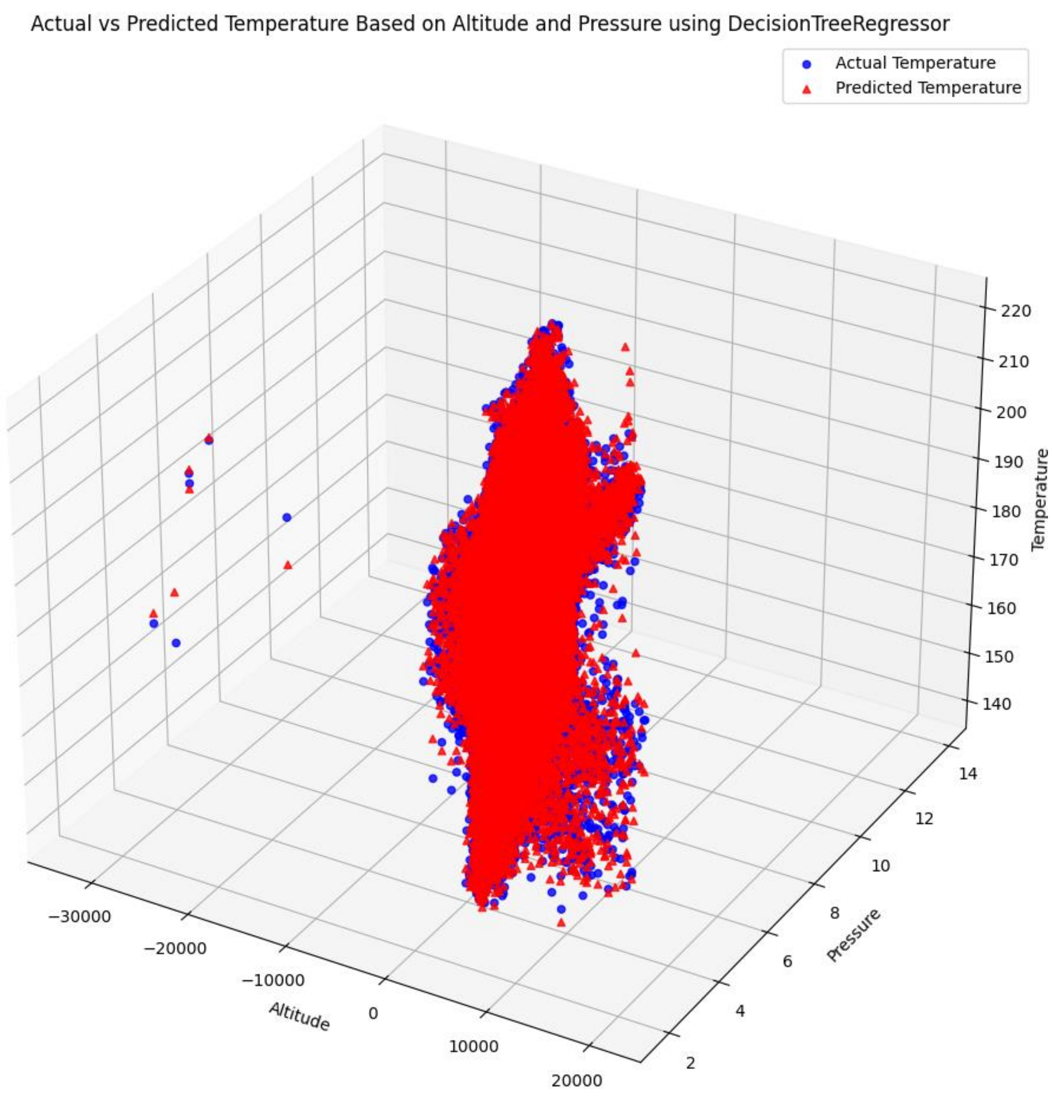


Fig Appendix A.4 Comparison of the actual and predicted temperature based on Altitude and Pressure using Decision Tree Regressor

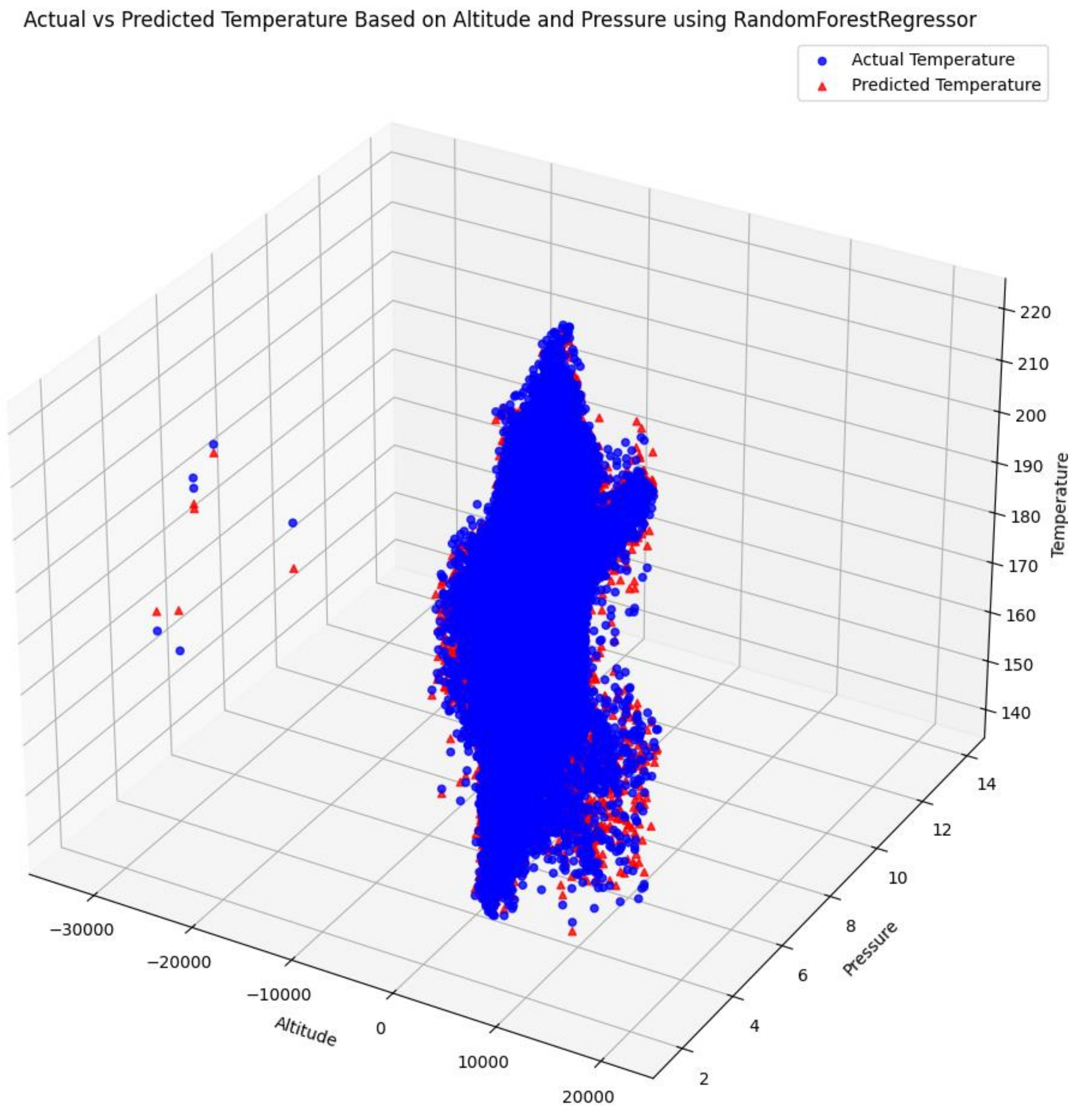


Fig Appendix A.5 Comparison of the actual and predicted temperature based on Altitude and Pressure using Random Forest Regressor