We have a Datetime, Magnetic field in x,y, and z directions, and other 50 physically dimensionless colum ns in the given source files so we name the dimensionless columns that affect flux as "SW_Flux".
A lot of data is missing and needs to be adjusted and replaced with a statistical figure. Data here is provid ed per minute.
The Kp index file is provided on a 3-hourly basis.
Introduction:
The quest 2 of the project "Aurora: A stellar Odessey", we have accomplished the task of predicting solar radiation and geomagnetic storms for the near future with the given data from 2015-23. Here, we intend th e give a brief overview of our approach to tackle this problem, the steps involved with an explanation of w hy we chose the particular methods, approach, model, etc.

Data Visualisation:
A lot of data is missing and needs to be adjusted and replaced with a statistical figure. Data here is provid ed per mintue.
The Kp index file is provided on a 3-hourly basis.
Data Preprocessing:

1. The 0 values were replaced with NaN
2. Sort the datetime so that we get all the data in the dataframe sorted chronologically.
3. Conversion of all the data to numerical data.
4. The dataframe is very large so it was convenient to use ffill and imputation (with mean and median)
5. Extraction of the kp index values from the given files by dropping all the other unnecessary columns.
6. Resize and sample the shape of the data. Since the kp index is measured every 3 hours we need to ad just our raw feature data accordingly to train a model.
7. Concatenate all the files i.e. all the feature dataset files from all years and the kp index files of the corre sponding years to make a single large data set.
8. Use graphs, histograms, and scatterplots to understand the target variable(here kp index) and its math ematical relations with the columns of the dataset.
9. Plot the graphs and use correlation as a medium to judge and select the best columns and features.
   Set a threshold to correlation(for example top 10 features since the threshold correlation coefficient for s electing features, we had set equal to 0.3. Thus we chose the top 10 columns.) and scale the features acc ordingly.
   Normalize the features using minmax scaler. (It is useful when the features have varying ranges and the algorithm used (e.g., neural networks) performs better with features within a specific range.)

Splitting the data into training validation and test sets
Analyze the size of the data set and split it accordingly (here 80 percent training, 10 percent validation, 10 percent train)

Selection of the model:
We use a Recurrent neural network called LSTM(Long Short term Memory) model for training because th e dataset we have a time variant and for such projects where the signal varies with time LSTMs are best s uited.

Training of the Model:
Analyze the size of features and training data and set the batch sizes, timesteps, epochs, and verbose ac cordingly(here epochs:64 batchsize:32 verbose:1)
Leave the model for training and let it complete the training.

Results:
Analyze the loss function(here mse: mean squared error) and then how it decreases with the epochs and save the predictions made.
Plot graphs of actual values and predictions and check how the model is performing. You can use scatter

plots, confusion matrices, and error analysis through statistical functions for checking results.
As mentioned in the problem statement calculate the RMSE value (for example it is close to 1.000628921 1972983).