Problem Statement: NASA and NOAA Satellites Solar-Wind Dataset

Predict DISTURBANCES In Earth’s Geomagnetic Field

Introduction

This study was carried out to achieve the prediction of the Disturbance storm time index variations based on one-hour interplanetary data and solar wind parameters using artificial neural networks.

The earth's magnetosphere is an intricate input-output system, where the inputs are the solar wind parameters and the output measure are the geomagnetic. The disturbance storm time (DST) index is one such quantity measure for the intensity of disturbance in the geomagnetic field due to the magnetic storm. The geomagnetic activities have catastrophic effects on several communication networks and harmful for biological livings. Therefore, it is significant to develop a model for the prediction of disturbance in the geomagnetic field.

As a key specification of the magnetospheric dynamics, the Dst index is used to drive geomagnetic disturbance models such as NOAA/NCEI’s High-Definition Geomagnetic Model - Real-Time (HDGM-RT). Additionally, magnetic surveyors, government agencies, academic institutions, satellite operators, and power grid operators use the Dst index to analyze the strength and duration of geomagnetic storms.

Empirical models have been proposed as early as in 1975 to forecast Dst solely from solar-wind observations at the Lagrangian (L1) position by satellites such as NOAA’s Deep Space Climate Observatory (DSCOVR) or NASA's Advanced Composition Explorer (ACE). Over the past three decades, several models were proposed for solar wind forecasting of Dst, including empirical, physics-based, and machine learning approaches. While the ML models generally perform better than models based on the other approaches, there is still room to improve, especially when predicting extreme events. More importantly, solutions that work on the raw, real-time data streams and are agnostic to sensor malfunctions and noise. Improved models can provide more advanced warning of geomagnetic storms and reduce errors in magnetic navigation systems.

In this we are going to perform a detailed eda on the dataset and predict the future dst values for 100 days using bidirectional lstm.

Content

The data is composed of solar wind measurements collected from two satellites: NASA's Advanced Composition Explorer (ACE) and NOAA's Deep Space Climate Observatory (DSCOVR).

To ensure similar distributions between the training and test data, the data is separated into three non-contiguous periods. All data are provided with a period and time delta multi-index which indicates the relative timestep for each observation within a period, but not the real timestamp. The period identifiers and time deltas are common across datasets.

The primary feature data are provided in solar_wind.csv. They are composed of solar-wind readings from the ACE and DSCOVR satellites:

- bx_gse - Interplanetary-magnetic-field (IMF) X-component in geocentric solar ecliptic (GSE) coordinate (nanotesla (nT))
- by_gse - Interplanetary-magnetic-field Y-component in GSE coordinate (nT)
- bz_gse - Interplanetary-magnetic-field Z-component in GSE coordinate (nT)
- theta_gse - Interplanetary-magnetic-field latitude in GSE coordinates (defined as the angle between the magnetic vector B and the ecliptic plane, being positive when B points North) (degrees)
- phi_gse - Interplanetary-magnetic-field longitude in GSE coordinates (the angle between the projection of the IMF vector on the ecliptic and the Earth–Sun direction) (degrees)
- bx_gsm - Interplanetary-magnetic-field X-component in geocentric solar magnetospheric (GSM) coordinate (nT)
- by_gsm - Interplanetary-magnetic-field Y-component in GSM coordinate (nT)
- bz_gsm - Interplanetary-magnetic-field Z-component in (GSM) coordinate (nT)
- theta_gsm - Interplanetary-magnetic-field latitude in GSM coordinates (degrees)
- phi_gsm - Interplanetary-magnetic-field longitude in GSM coordinates (degrees)
- bt - Interplanetary-magnetic-field component magnitude (nT)
- density - Solar wind proton density (N/cm^3)
- speed - Solar wind bulk speed (km/s)
- temperature - Solar wind ion temperature (Kelvin)
- source - Starting in 2016, the solar wind data for any given point in time can be sourced from either DSCOVR or ACE satellites depending on the quality. "ac" denotes it was sourced from ACE, and "ds" from DSCOVR.

Satellite Data

ACE and DSCOVR satellites are not stationary. They actually orbit around the L1 point, in a relatively constant position with respect to the Earth as the Earth revolves around the sun.

satellite_pos.csv records the daily positions of the DSCOVR and ACE Spacecrafts in Geocentric Solar Ecliptic (GSE) Coordinates for projections in the XY, XZ, and YZ planes. The columns for each spacecraft are denoted by the suffixes _ace or _dscovr.

- gse_x - Position of the satellite in the X direction of GSE coordinates (km)
- gse_y - Position of the satellite in the Y direction of GSE coordinates (km)
- gse_z - Position of the satellite in the Z direction of GSE coordinates (km)

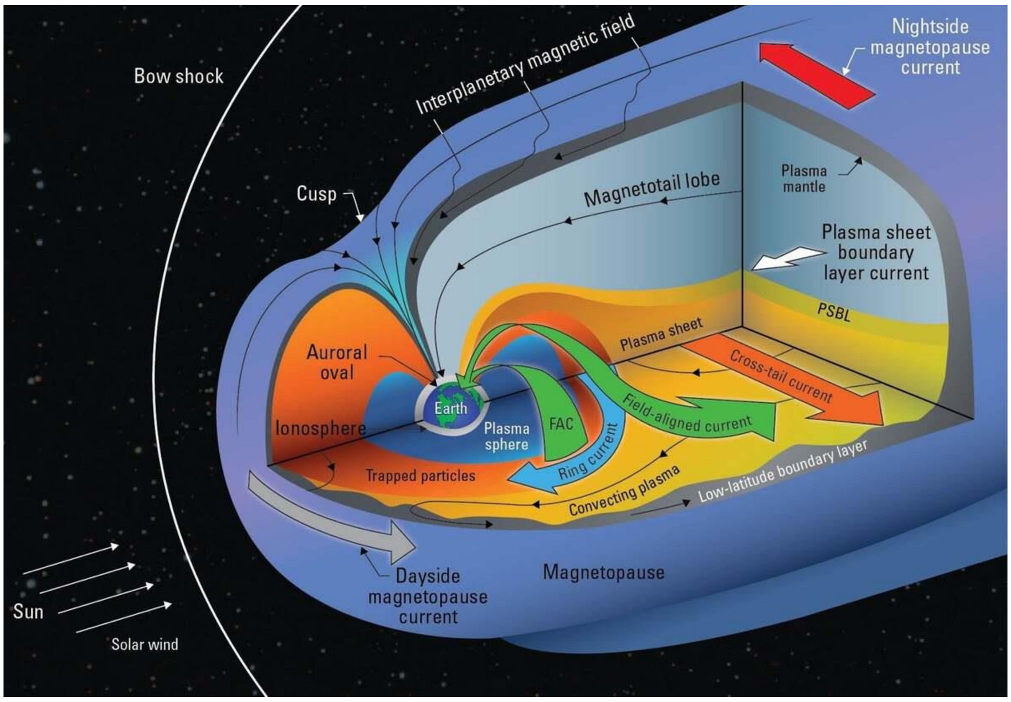
Sunspots Data

The Sun exhibits a well-known, periodic variation in the number of spots on its disk over a period of about 11 years, called a solar cycle. In general, large geomagnetic storms occur more frequently during the peak of these cycles. Sunspot numbers might allow for calibration of models to the solar cycle.

Sunspots are indexed according to the first corresponding day in labels.csv.

Labels

The labels are hourly Dst values, indexed using the same period and timedelta multi-index.

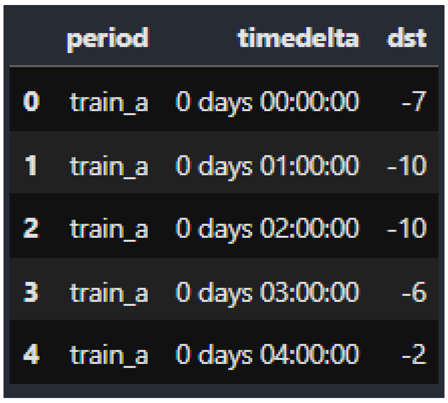


Goal

The goal is to predict dst value for 100 days in each period.

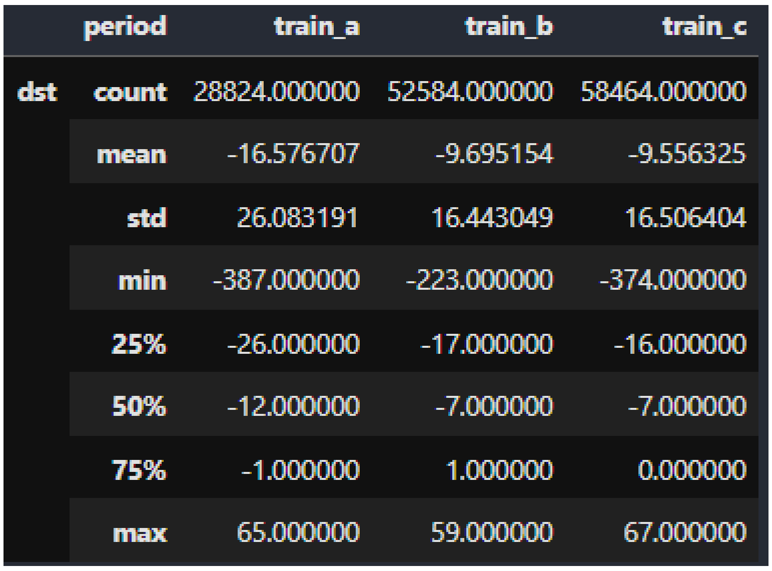
EDA on Dataset

On viewing the labels.csv we can find hourly dst recordings for three periods.



	period	timedelta	dst
0	train_a	0 days 00:00:00	-7
1	train_a	0 days 01:00:00	-10
2	train_a	0 days 02:00:00	-10
3	train_a	0 days 03:00:00	-6
4	train_a	0 days 04:00:00	-2

On applying describe function:

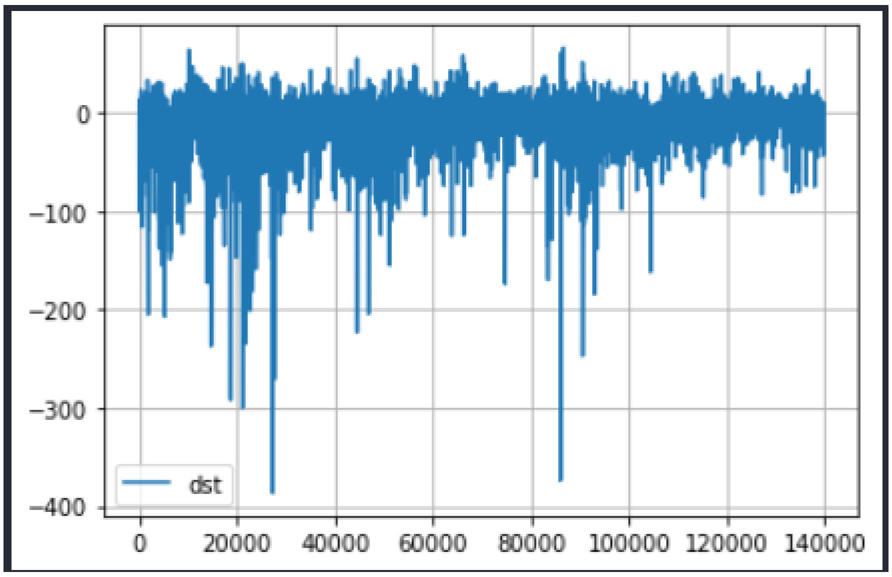


	period	train_a	train_b	train_c
dst	**count**	28824.000000	52584.000000	58464.000000
	mean	-16.576707	-9.695154	-9.556325
	std	26.083191	16.443049	16.506404
	min	-387.000000	-223.000000	-374.000000
	25%	-26.000000	-17.000000	-16.000000
	50%	-12.000000	-7.000000	-7.000000
	75%	-1.000000	1.000000	0.000000
	max	65.000000	59.000000	67.000000

We have nearly 140,000 observations of hourly dst data, representing over 15 years. There are almost twice as many observations in either the train_b or train_c periods than there are train_a. Also note that most of the values are negative. It also seems train_a represents a more intense period, given that it has a lower mean and higher standard deviation. So we may have outliers.

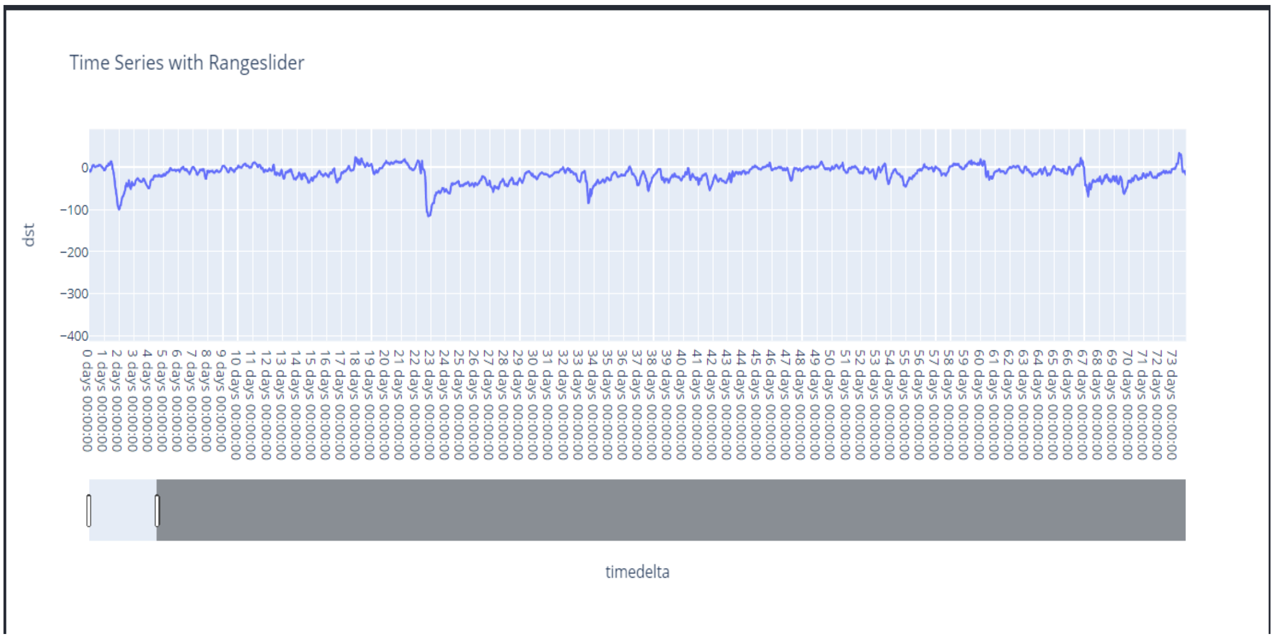
A very strong magnetic field disturbance has a large Dst value, measured in nano-Teslas (nT). Because these disturbances are usually flowing towards the Earth, the values are negative. Sometimes Dst can be highly positive. During calm conditions, Dst values are situated at or just below 0.

On plotting the dst values we observe:



We observe that there are many big spikes which are outliers

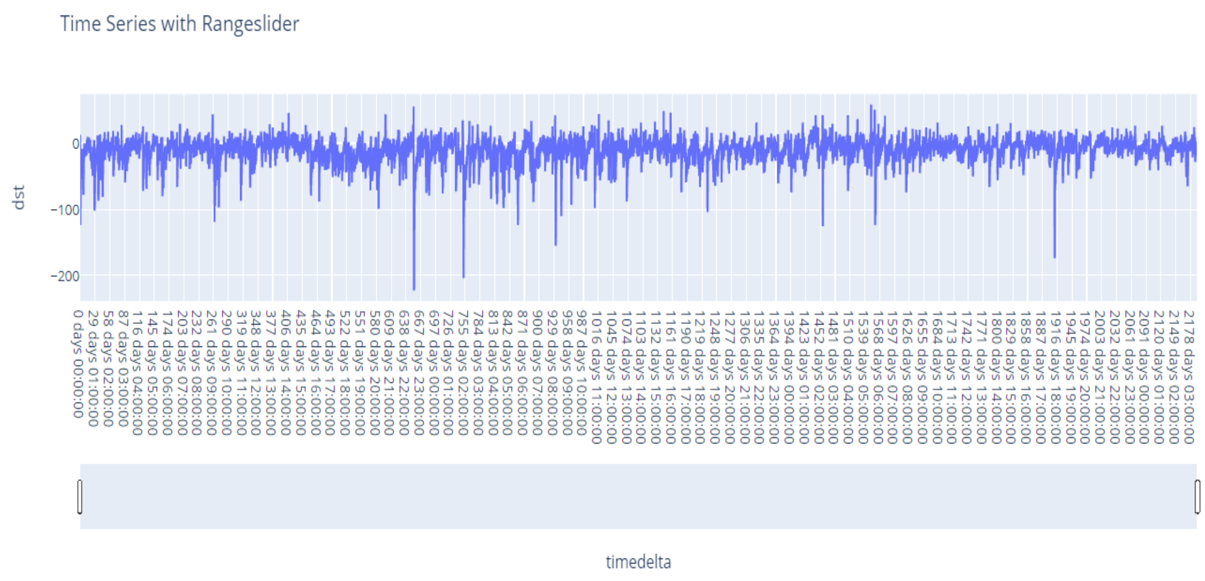
For better visualization lets plot this with the help of plotly



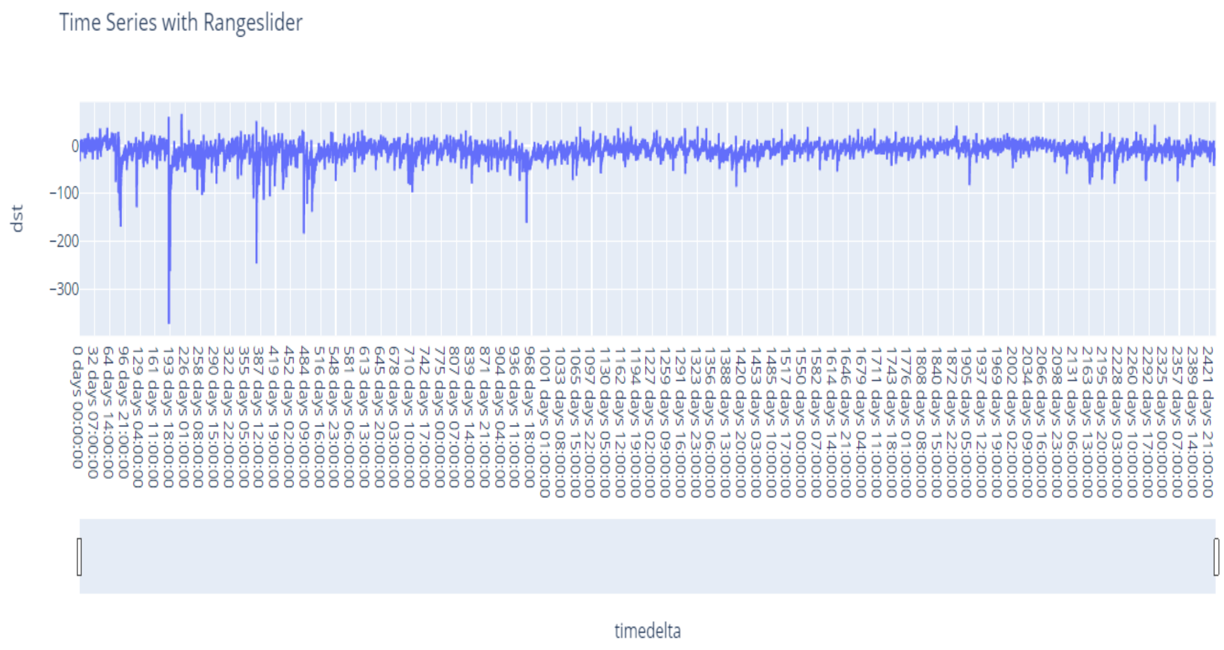
Here we are visualizing the data day wise for period train_a.

We observe that generally the value is lying in range (100,0) and (0,-100).

So lets visualize the data for periods train_b and train_c also.



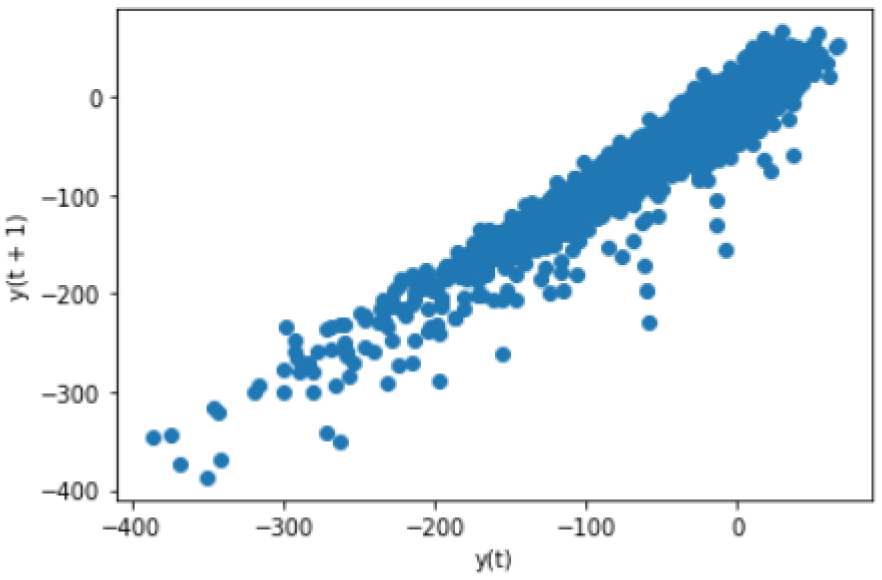
train_b



train_c

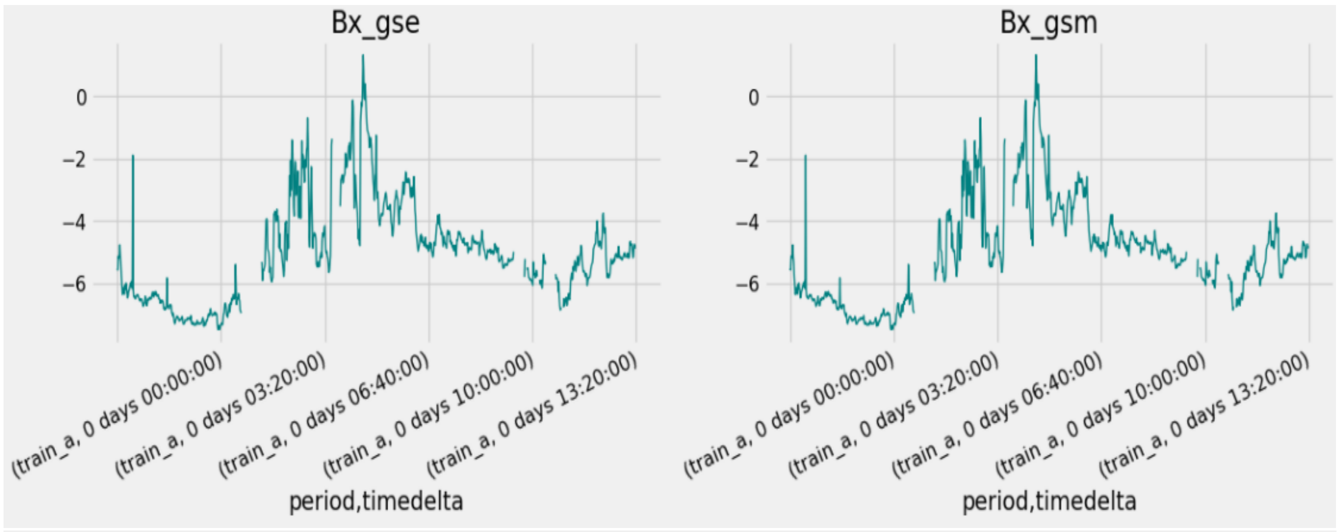
Let's plot an autocorrelation plot

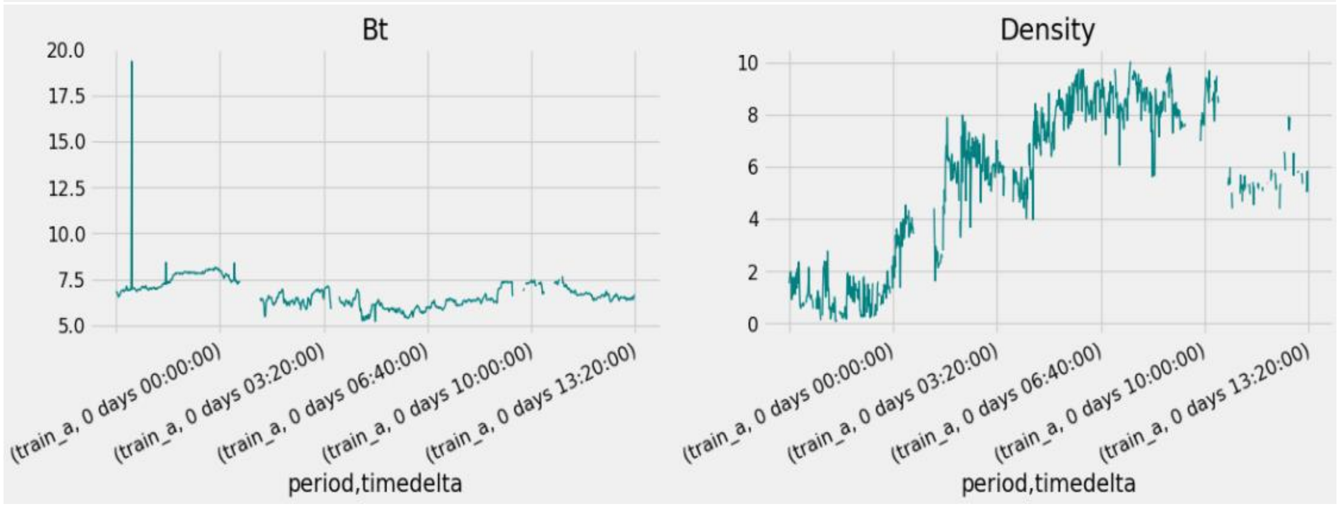
Autocorrelation plots are a commonly-used tool for checking randomness in a data set. It basically shows the relation with itself with some lags.

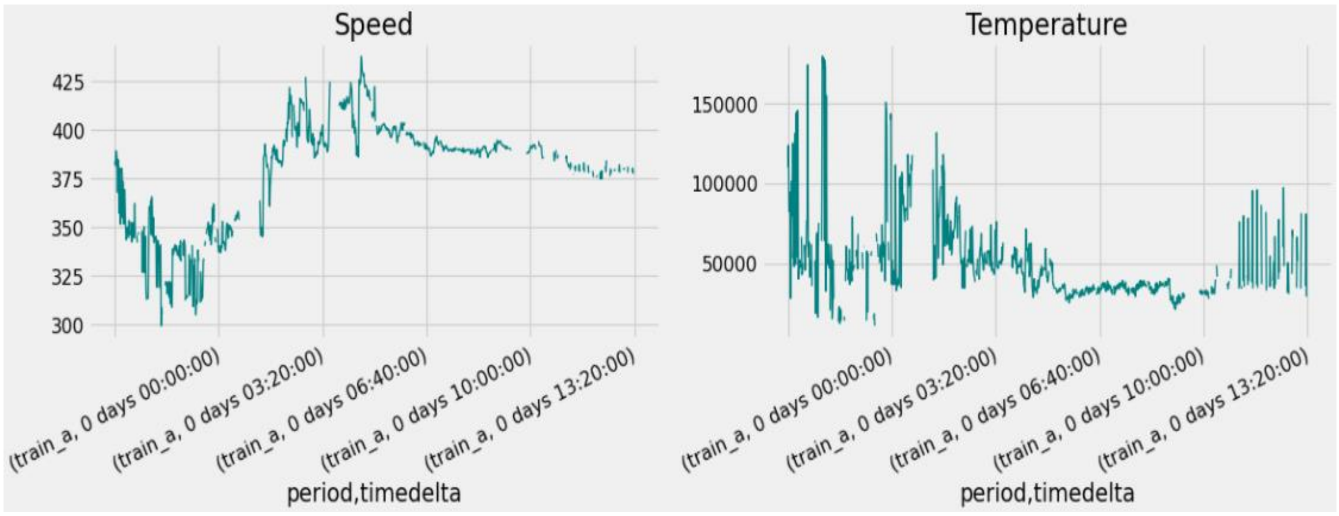


Though most of the datapoints combine to form a straight line we can see some abrupt jumps. These indicate that there are outliers.

Let's do a few visualizations of other datasets. First, we'll just plot the first 1,000 rows for some of our time series data to get a sense of its shape. Instead of plotting all of our features, let's just choose a few IMF features and a few plasma features.



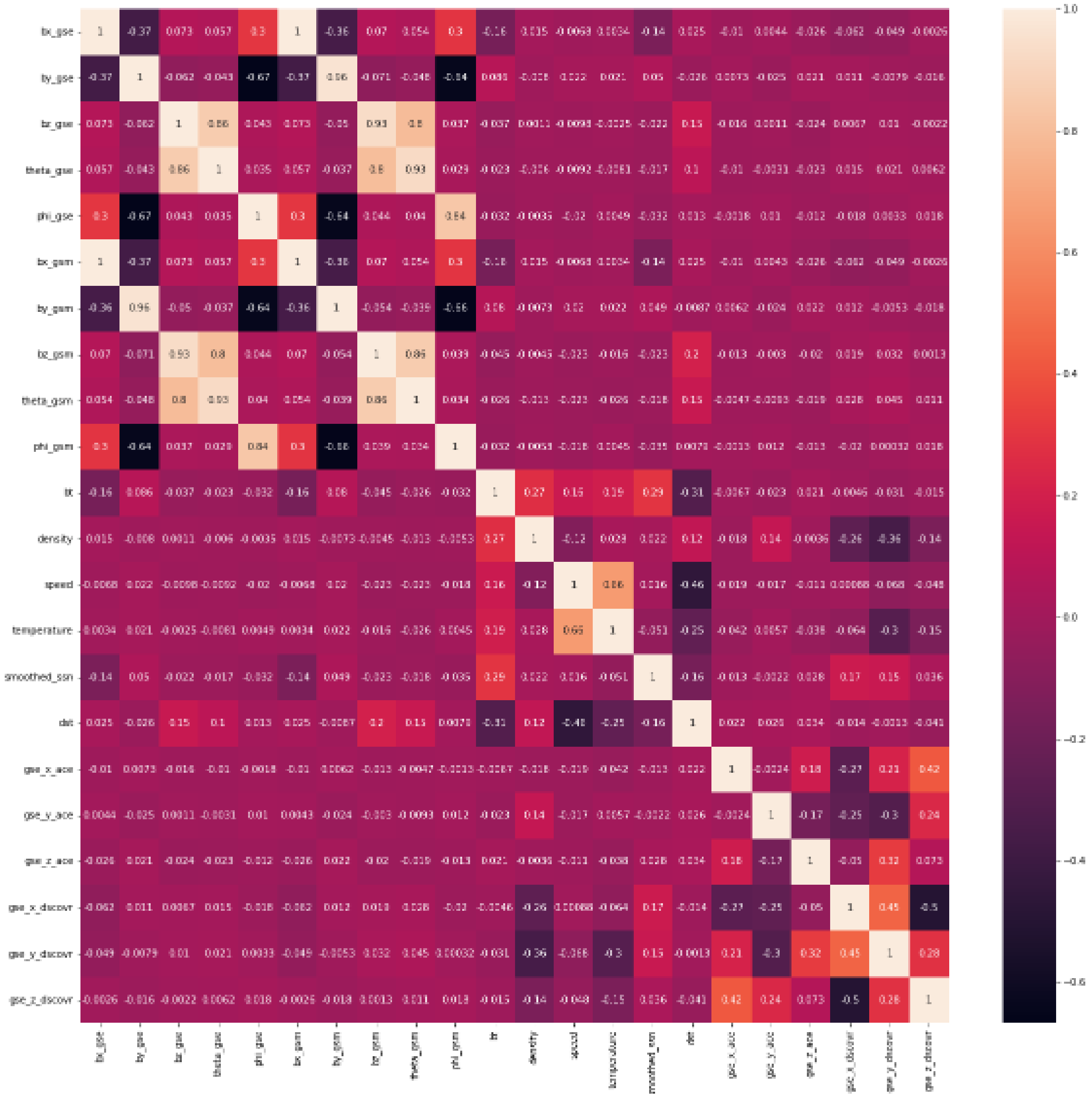




On exploring other datasets, we found that We got an insight that values exist across very different scales. For instance, temperature values are quite high - reaching into the hundreds of thousands Kelvin. Meanwhile, IMF readings are fairly small values, and usually negative.

So, we have to scale it before training.

As our last exploratory step, let's plot a correlation matrix to find other relationships between our features.



The plasma related features like speed and temperature look to be strongly anti-correlated with Dst. The IMF feature bt also exhibits strong anti-correlation.

The gsm and gse variables are strongly correlated. This could present multicollinearity issues if both are present in our model. We probably want to leave some of those out.

If we had to do multivariate time series analysis, we could have selected following features: speed ,temperature ,bt , density, gse variables.

Also we had to convert all the values to daily data.

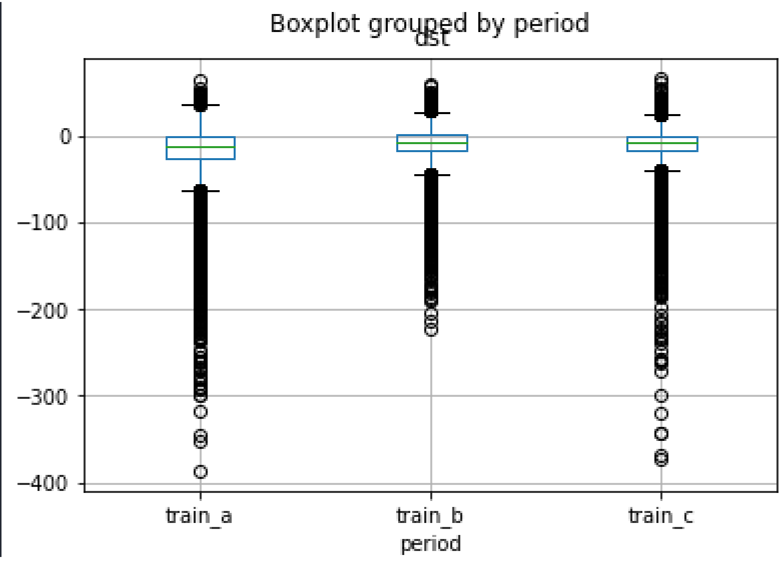
For this project we will be sticking to univariate time series analysis using labels.csv

Time Series Analysis

Time series analysis is a specific way of analyzing a sequence of data points collected over an interval of time. In time series analysis, analysts record data points at consistent intervals over a set period of time rather than just recording the data points intermittently or randomly. What sets time series data apart from other data is that the analysis can show how variables change over time. In other words, time is a crucial variable because it shows how the data adjusts over the course of the data points as well as the final results. It provides an additional source of information and a set order of dependencies between the data. Time series analysis typically requires a large number of data points to ensure consistency and reliability. An extensive data set ensures you have a representative sample size and that analysis can cut through noisy data. It also ensures that any trends or patterns discovered are not outliers and can account for seasonal variance. Additionally, time series data can be used for forecasting—predicting future data based on historical data. Time series analysis helps organizations understand the underlying causes of trends or systemic patterns over time.

Initially we see that the shape of labels.csv is `(139872, 3)`

Let's check for outliers. We will check the outliers by plotting pandas boxplot.



Here we can see that there are some outliers present in our data

We will use **IQR** method to remove outliers.

On checking the shape at this point we see that the shape is `(133431, 3).` Here we see that 6441 rows are dropped.

Now we will use 70% of data as training data and 30% of data as testing data. We have three non-contiguous periods, meaning we have gaps in our data. We don't know how long each gap is or in what order each period occurred. We also know that the three periods are differently distributed. That suggests that observations from each period should be included in our train set, instead of reserving one wholesale as our test set. So, we will use 70% of data from each period for training and 30% of data from each period for testing.

We have to scale the data. For this purpose, we will use minmax scaler. (This estimator scales and translates each feature individually such that it is in the given range on the training set, e.g., between zero and one. It is given by the formula:

$X_{scaled}=X-X_{min}/X_{max}-X_{min}$)

After splitting the dataset into training and testing dataset we will divide them into batches of data according to timestep. For ex: - in our case the time step is 100. So, 1st 100 data will go to X_train and 101th data will go to Y_test. And such process will continue....

We see that there are 27283 batches in X_train and 105946 batches in Y_train.

In the next step we have to reshape X_train and Y_train.

In this project we are going to use bidirectional LSTM.

On checking the RSME, we get the RSME of 25.061532673650408 on training data and 25.231458010859573 on test data.

By using this trained model we predicted the future dst values for next 100 days in each period.

For train_a

```
[[0.44423043727874756], [0.17952172458171844], [0.11787647008895874],
[0.11878626048564911], [0.11142745614051819], [0.10155448317527771],
[0.09499438107013702], [0.09184513241052628], [0.09031254053115845],
[0.08907835930585861], [0.08757199347019196], [0.08566667884588242],
[0.08337543159723282], [0.08063790947198868], [0.07736861705780029],
[0.07307097315788269], [0.06973081827163696], [0.06724272668361664],
[0.06537781655788422], [0.06401615589857101], [0.06301382184028625],
[0.06222725287079811], [0.06158791482448578], [0.06107543781399727],
[0.06064913794398308], [0.060306135565042496], [0.060028161853551865],
[0.05979515612125397], [0.05962014198303223], [0.05948130413889885],
[0.059368111193180084], [0.059275597333390808], [0.059201281517744064],
[0.05914371833205223], [0.05909609794616699], [0.05906355381011963],
[0.05904316902160645], [0.0590297468006610], [0.05901502817869186],
[0.05900442600250244], [0.05898824334144592], [0.05897646397352218],
[0.05895831808447838], [0.05895959585905075], [0.05895461142063141],
[0.05895329639315605], [0.05895401909947395], [0.058957114815711975],
[0.0589621402323246], [0.058966442942619324], [0.058968834578990936],
[0.058972373604774475], [0.058975737541913986], [0.05897940322756767],
[0.0589844770729541], [0.058989714831113815], [0.05899505317211151],
[0.058999087661504745], [0.059002045542001724], [0.059007346630096436],
[0.05901368707418442], [0.05901901423931122], [0.05902332067489624],
[0.05902784317731857], [0.059035126119852066], [0.059046532958745956],
[0.05904994159936905], [0.05905363708734512], [0.0590573213994503],
[0.05906122550368309], [0.059070341289043427], [0.059072110801935196],
[0.059071529656648636], [0.05906977877020836], [0.059069715440273285],
[0.059070855379104614], [0.05907334387302399], [0.05908322334289551],
[0.0590890496969223], [0.05909554660320282], [0.05909876525402069],
[0.05909904092550278], [0.0590985082089901], [0.05909952521324158],
[0.05909937247633934], [0.05910176783800125], [0.05910281464457512],
[0.05910282954573631], [0.05910252034664154], [0.059102002531290054],
[0.05910176783800125], [0.059105247259140015], [0.059118084609508514],
[0.059124305844306946], [0.05912528932094574], [0.059131771326065063],
[0.05913287773728371], [0.059138376265764236], [0.059137992560863495],
[0.05913504958152771]]
```

For train_b

```
101[[0.44423043727874756], [0.17952172458171844], [0.11787647008895874],
[0.11878626048564911], [0.11142745614051819], [0.10155448317527771],
[0.09499438107013702], [0.09184513241052628], [0.09031254053115845],
[0.08907835930585861], [0.08757199347019196], [0.08566667884588242],
[0.08337543159723282], [0.08063790947198868], [0.07736861705780029],
[0.07307097315788269], [0.06973081827163696], [0.06724272668361664],
[0.06537781655788422], [0.06401615589857101], [0.06301382184028625],
[0.06222725287079811], [0.06158791482448578], [0.06107543781399727],
[0.06064913794398308], [0.060306135565042496], [0.060028161853551865],
[0.05979515612125397], [0.05962014198303223], [0.05948130413889885],
[0.059368111193180084], [0.05927559733390808], [0.059201281517744064],
[0.05914371833205223], [0.05909609794616699], [0.05906355381011963],
[0.05904316902160645], [0.0590297468006610], [0.05901502817869186],
[0.05900442600250244], [0.05898824334144592], [0.05897646397352218],
[0.05895831808447838], [0.05895959585905075], [0.05895461142063141],
[0.05895329639315605], [0.05895401909947395], [0.058957114815711975],
[0.0589621402323246], [0.058966442942619324], [0.058968834578990936],
[0.058972373604774475], [0.058975737541913986], [0.05897940322756767],
[0.0589844770729541], [0.058989714831113815], [0.05899505317211151],
[0.058999087661504745], [0.059002045542001724], [0.059007346630096436],
[0.05901368707418442], [0.05901901423931122], [0.05902332067489624],
[0.05902784317731857], [0.059035126119852066], [0.059046532958745956],
[0.05904994159936905], [0.05905363708734512], [0.0590573213994503],
[0.05906122550368309], [0.059070341289043427], [0.059072110801935196],
[0.059071529656648636], [0.05906977877020836], [0.059069715440273285],
[0.059070855379104614], [0.05907334387302399], [0.05908322334289551],
[0.0590890496969223], [0.05909554660320282], [0.05909876525402069],
[0.05909904092550278], [0.0590985082089901], [0.05909952521324158],
[0.05909937247633934], [0.05910176783800125], [0.05910281464457512],
[0.05910282954573631], [0.05910252034664154], [0.05910200253129005 4],
[0.05910176783800125], [0.05910524725914001 5], [0.05911808460950851 4],
[0.05912430584430694 6], [0.05912528932094574], [0.05913177132606506 3],
[0.05913287773728371], [0.05913837626576423 6], [0.05913799256086349 5],
[0.05913504958152771]]
```

For train_c

```
[[0.44423043727874756], [0.17952172458171844], [0.11787647008895874],
[0.11878626048564911], [0.11142745614051819], [0.10155448317527771],
[0.09499438107013702], [0.09184513241052628], [0.09031254053115845],
[0.08907835930585861], [0.08757199347019196], [0.08566667884588242],
[0.08337543159723282], [0.08063790947198868], [0.07736861705780029],
[0.07307097315788269], [0.06973081827163696], [0.06724272668361664],
[0.06537781655788422], [0.06401615589857101], [0.06301382184028625],
[0.06222725287079811], [0.06158791482448578], [0.06107543781399727],
[0.06064913794398308], [0.060306135565042496], [0.060028161853551865],
[0.05979515612125397], [0.05962014198303223], [0.05948130413889885],
[0.059368111193180084], [0.05927559733390808], [0.05920128151774406 4],
[0.05914371833205223], [0.05909609794616699], [0.05906355381011963],
[0.059043169021606445], [0.05902974680066109], [0.059015028178691864],
[0.05900442600250244], [0.05898824334144592], [0.058976463973522186],
[0.05895831808447838], [0.05895959585905075], [0.05895461142063141],
[0.05895329639315605], [0.05895401909947395], [0.058957114815711975],
[0.0589621402323246], [0.05896644294261932 4], [0.05896883457899093 6],
[0.058972373604774475], [0.058975737541913986], [0.05897940322756767],
[0.05898447707295418], [0.058989714831113815], [0.05899505317211151],
[0.058999087661504745], [0.059002045542001724], [0.059007346630096436],
[0.05901368707418442], [0.05901901423931122], [0.05902332067489624],
[0.05902784317731857], [0.059035126119852066], [0.059046532958745956],
[0.05904994159936905], [0.05905363708734512], [0.0590573213994503],
[0.05906122550368309], [0.059070341289043427], [0.059072110801935196],
[0.05907152965664863 6], [0.05906977877020836], [0.059069715440273285],
[0.059070855379104614], [0.05907334387302399], [0.05908322334289551],
[0.0590890496969223], [0.05909554660320282], [0.05909876525402069],
[0.05909904092550278], [0.0590985082089901], [0.05909952521324158],
[0.05909937247633934], [0.05910176783800125], [0.05910281464457512],
[0.05910282954573631], [0.05910252034664154], [0.05910200253129005 4],
[0.05910176783800125], [0.05910524725914001 5], [0.05911808460950851 4],
[0.05912430584430694 6], [0.05912528932094574], [0.05913177132606506 3],
[0.05913287773728371], [0.05913837626576423 6], [0.05913799256086349 5],
[0.05913504958152771]]
```