



DATA SCIENCE AND ANALYTICS DISSERTATION

Classification of Bow shock and Magnetopause event positions by using the Magnetic field properties of Saturn

A thesis submitted in partial fulfillment of the requirement for the degree of M.Sc. in Data Science and Analytics.

Author: Mathews Philip Venattu (20250487)

Supervisor: Dr. Katarina Domijan

Department of Mathematics and Statistics
University: National University of Ireland, Maynooth.

Date Submitted: 10th August, 2021

Abstract

Cassini-Huygens Space Research mission was a joint collaboration of NASA, ESA and Italian Space agency to study about the planet Saturn and its system that includes its Rings and Natural Satellites. During its life span of about 20 years the spacecraft orbited the planet Saturn for 13 years and also frequently changed its shape and tilt. Because of this frequent changes in the orbital trajectory of spacecraft, it passed multiple times through the boundaries of Bow shock and Magnetopause at different latitude, longitude and phases of Solar Cycle. In this Project, I have used the Magnetic Field data collected during the year 2005 from MAG (Magnetometer) instrument attached to the spacecraft to develop a classification model for detecting the Magnetopause and bow shock boundaries at Saturn . Magnetometer (MAG)in Cassini recorded the strength and direction of the Magnetic field around the spacecraft while it was orbiting the planet Saturn. Logistic Regression model was used to find significant predictors in different datasets when classifying Bow Shock events and Magnetopause events. A Random forest and Lasso regression model was developed for classifying the different events and analyzed those models based on it accuracy and sensitivity.

Contents

List of tables	1
List of figures	2
1 Introduction	3
2 Background	5
3 Datasets	6
3.1 Raw Datasets	6
3.1.1 Dataset 1: Magnetopause And Bow shock Crossing List	6
3.1.2 Dataset 2: Magnetometer Dataset	7
3.2 Derived Datasets	7
3.2.1 Dataset 3: Combined Data Of Dataset 1 And Dataset 2	8
3.2.2 Dataset 4: Average and Standard Deviation Dataset	8
4 Data Manipulation Section	10
4.1 Removing Time Dependency	10
4.2 Data Imputation	10
5 Exploratory Data Analysis	11
5.1 Trajectory of Spacecraft	12
5.2 Examining the Data Imbalance	12
5.3 Proportion of Direction of Cross in Different Classes	13
6 Significance of Predictors	14
6.1 Logistic Regression Model 1: Bow Shock vs Other Events	14
6.1.1 Results	14
6.2 Logistic Regression Model 2: Magnetopause vs Other Events	16
6.2.1 Results	16
6.3 Conclusion	18
7 Classification Model Development	19
7.1 Random Forest Model	19
7.1.1 Setting up Random Forest Model	19
7.1.2 Train and Test Dataset	19
7.1.3 Results	19
7.2 Lasso Regression Model	22
7.2.1 Setting up Lasso Regression Model	22
7.2.2 Train and Test Dataset	22
7.2.3 Results	23
7.3 Overall Results	26
8 Conclusion	27
9 References	28
10 Appendix	29
10.1 Supporting code & Supporting Files Link	29

List of tables

1	Number of Datapoints in each Boundary Class	12
2	Logistic Regression : Bow Shock Vs Other Events Results Table	15
3	Logistic Regression - Bow Shock Vs Other Events Results Table (Dataset 3) . . .	15
4	Logistic Regression : Bow Shock Vs Other Events Results Table (Dataset 4) . . .	15
5	Logistic Regression : Bow Shock Vs Other Events Results Table (Dataset 3 without Lead variables)	16
6	Logistic Regression : Magnetopause Vs Other Events Results Table	16
7	Logistic Regression : Magneto Pause Vs Other Events Results Table (Dataset 3)	17
8	Logistic Regression : Magneto Pause Vs Other Events Results Table (Dataset 4)	17
9	Logistic Regression : Magneto Pause Vs Other Events Results Table (Dataset 3 without Lead Variables)	18
10	Overall Results: Significance of Predictors	18
11	Overall Results: Model Selection	26

List of figures

1	Diagram of Planetary Magnetosphere	3
2	Line Graph of Total Magnetic Field Strength between 72 and 74th day of the year 2005	11
3	Line Graph of Total Magnetic Field Strength between 136 and 138th day of the year 2005	11
4	Boundary Crossing points on the Trajectory of Cassini Space craft	12
5	Proportion of different directions on each Class	13

1 Introduction

All planets in our solar system have magnetic field like earth. Some of the planets like Uranus, Saturn, Jupiter, and Neptune have large magnetic field than earth. Magnetosphere of an astronomical object is the area surrounding that object where its Magnetic field is present. Like all other planets Saturn also has the similar magnetosphere structure - it has a Bow shock, Magnetosheath, Magnetopause and Magneto tail. Among this structure bow shock is the point at which the Magnetosphere of the Saturn interacts with the solar wind that in turn suddenly reduces its speed and pressure. Magnetopause is the boundary between Saturn's Magnetic field and Solar wind. The Magnetosheath exists between the Bow shock and the Magnetopause, an area of shocked solar wind that is significantly influenced by the changes that occur within the Bow shock and whose features can impact the interaction across the Magnetopause. The dynamic pressure of Solar wind usually determines boundary and position of Magnetopause and Bow shock. The Cassini Huygens Mission is a joint NASA/ESA/ASI project to make a detailed survey of the ringed planet Saturn and its natural satellites. Cassini spacecraft recorded magnetic field and plasma condition of the environment during its insertion to Saturn's orbit by using the Cassini Magnetometer (MAG), Plasma Spectrometer (CAPS). The magnetic field strength pattern clearly shows some "overshoot" and "foot" when the spacecraft passed through the boundaries of Magnetopause, Bow shock and Magnetosheath (Achilleos et al. (2006)). The Magnetometer (MAG) which continuously acquired magnetic field data that is associated with the Plasma Environment and internal source of Saturn are essential to study about the interaction between solar wind and Magnetosphere of Saturn.

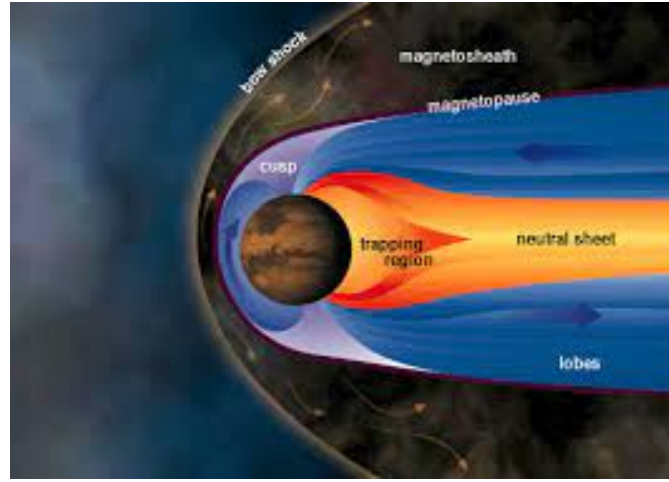


Figure 1: Diagram of Planetary Magnetosphere

Figure 1 is a schematic diagram of how Bow shock and Magnetopause boundaries look around the planet Saturn. Magnetosphere of Saturn acts as an obstacle to the Solar wind far away from the planet and the Magnetopause of the planet Saturn lies in 20 R_s (Radius of Saturn=60330 km). So, the Solar wind which interacts with the magnetosphere is decelerated by the bow shock and the physical properties of the plasma (Sergis et al. (2013)). The magnetospheric pressure, which inflates the equatorial magnetosphere considerably more than the high-latitude magnetosphere, affects the geometry of the Magnetopause boundary itself, resulting in clear polar flattening

(Pilkington et al. (2014)). In the case of Earth, basic pressure balance is due to the interaction between upstream solar wind flow against magnetic pressure from the magnetosphere and this helps to draw the approximate location of Magnetopause boundary but when it comes to Saturn we must consider the influence of the natural satellite Enceladus, which serves as a huge internal plasma source. The pressure related with the super thermal component of this internally produced plasma serves to inflate the magnetosphere considerably beyond what a basic magnetic pressure calculation would predict (Pilkington et al. (2015)). So even in a steady solar wind conditions, due to the internal plasma mentioned above the Magnetopause boundary of Saturn may move significantly.

Lots of studies were done based on the data acquired from the Cassini Spacecraft instruments. (Sergis et al. (2013)) they chose intervals between 2004 and 2011 when the spacecraft was travelling through the magnetic sheath and used the data recorded to explore different properties like plasma, energetic particle, magnetic field density, temperature etc. They concentrated more on the presence of W⁺s ions (water group) and explained the ejection of energetic ions as a function of pitch angle and energy which shows the often flow of heavy energetic ions from bow shock. Analysis of CAPS (Cassini Plasma Spectrometer) by (Burkholder et al. (2017)) shown the significant ion flow differences prenoon and postnoon, and apart from that magnetic field data was used to illustrate the rotation of IMF (Interplanetary Magnetic Field) vector.

In this project we are considering the bow shock and Magnetopause crossing of the spacecraft during the year 2005. This report explains about the entire project with different sections like Background, Dataset explanation, Data Manipulation, Data Visualization, Models trained to classify the type of crossings, R code used, summary of the results and conclusion. Background section explains about the different terms used and current approaches done by others for classification. In Dataset explanation section, it explains about each variable and the different datasets that were used for this project. Data Manipulation section gives a detailed explanation about the transformations and Different models were trained with different derived datasets to achieve better results. Each models that were used for the classification in this project was explained on the Model Developmnet Section of this report. Remaining portions explains about the code and its results.

2 Background

In the early days, scientists have very less information regarding the Planet Saturn and its magnetosphere because as we all know that the magnetic fields are invisible, and it needs to be studied from inside. Cassini Huygens mission was a great opportunity for the science world to explore the planet and its behavior. This mission helped to study the magnetic field and the flow of different gases under the influence of the magnetic field which affects the auroras of Saturn. This mission had given some powerful insights about the atmosphere and the surrounding of Planet Saturn. By comparing Saturn with the similar exoplanets will give information regarding the evolution of the Solar System. Different studies were done based on the data gathered from the Cassini spacecraft. Based on this data Sulaiman et al. (2017) demonstrated that polar flattening of the Magnetopause causes shorter streamline pathways over the poles, resulting in a higher-pressure gradient, which twists the field. This in turn leads to different conditions at the Magnetopause when compared to those predicated based on axisymmetric assumptions. From 2004 day 299 through 2012 day 151, a substantial data was compiled by Thomsen et al. (2018) of Magnetosheath measurements was collected using data from CAPS, MAG, and MIMI. This data collection enables researchers to investigate things like local temporal dependence of Magnetosheath parameters. They also demonstrated a new method for estimating upstream solar wind speed using the same Magnetosheath parameters. Pan et al. (2021) used the MAG data for research which provides a broad picture of low-frequency waves in Saturn's magnetosphere, which has crucial consequences for how magnetospheric energy leaks.

Both the Bow shock and Magnetopause models can be used as a significant tool which gives insights about the solar upstream conditions and its dynamic pressure at which they are associated. All the data associated with Cassini uses KSM coordinate system and this system is Saturn centered where the x -axis is towards the sun (Masters et al. (2013)). Orbital tour of Cassini spacecraft around the planet Saturn was started in the month of July 2004. During that time the spacecraft crossed 100 Bow Shock boundaries. A study done by Pan et al. (2021) On 11th and 12th of April 2005, Cassini magnetometer readings were made during a typical sequence of Cassini bow shock crossings. The spacecraft began and finished the period downstream of the shock in the Magnetosheath solar wind, with two trips into the upstream solar wind, each separated by two shock crossings. The presence of obvious shock ramps and a constant upstream field indicates that these are quasi-perpendicular crossings. During this time, the magnetic field strength values that were recorded by the magnetometer was so high. In this report I analysed the magnetometer and position data of Cassini spacecraft during the year 2005 to classify the Magnetopause and Bow Shock events.

3 Datasets

Cassini Spacecraft orbited around the Saturn for about 9 years. During this period, the spacecraft transmitted valuable information regarding Saturn like the magnetic field strength, position at which it was measured to earth. Magnetometer and CAPS are the main instruments that were used for measuring the magnetic field strength and Kinetic Energy of particles at each point. For this project, I only considered the data that was recorded during the year 2005 by the spacecraft. Mainly two datasets were used in this project to make a final combined useful dataset, first dataset contains a list of Bow Shock and Magnetopause event crossings that occurred during the year 2005 (Jackman, Thomsen, and Dougherty (2019)). The second dataset contains the information regarding the position of spacecraft and the vector data of Magnetic field strength.

3.1 Raw Datasets

3.1.1 Dataset 1: Magnetopause And Bow shock Crossing List

This dataset only contains the data of the year 2005 and that was originally developed by compiling two datasets that are posted in the MAPSView webpage (<http://mapskp.cesr.fr/BSMP/index.php>) which contains the Bow Shock and Magnetopause event crossings between 2004 day 179 and 2007 day 349 (H.J. McAndrews, S.J. Kanani, A. Masters, and J.C. Cutler) through visual identification of CAPS and MAG data. The second list of data has the Magnetopause crossings during the year 2004 to October 2010 and May 2012 to February 2013 (Pilkington et al. (2015)).

This dataset contains seven variables: `year_cross`, `doy_cross`, `doyfrac_cross`, `c_cross`, `hour_cross`, `minute_cross`, `type_cross`, `dirn_cross`, `xcrosslist`, `ycrosslist`, `zcrosslist`

Variable Description

- `year_cross`: It contains a numeric value of the year in which spacecraft crossed the event.
- `doy_cross`: It contains a numeric value of the day on which spacecraft crossed the event.
- `hour_cross`: It contains a numeric value of the hour at which spacecraft crossed the event.
- `minute_cross`: It contains a numeric value of the minute at which spacecraft crossed the event.
- `doyfrac_cross`: $\text{doy_cross} + (\text{hour_cross}60 + \text{minute_cross}) / (2460)$
- `type_cross`: This is a categorical variable contains information about what type of event did the spacecraft crossed.
 - **MP**: Magnetopause
 - **BS**: Bow Shock
 - **DG**: Data gap
 - **SC**: SCAS interval which are unreliable data
- `dirn_cross`: This is also a Categorical variable that contains information regarding in which direction did the spacecraft moved. The direction categories in this variable are:
 - **in**: Inbound means the spacecraft is moving towards the planet.
 - **out**: Outbound means the spacecraft is moving away from the planet.
 - **S_SW**: Starts with the solar wind is the region at which spacecraft recorded values at the start of solar wind.
 - **S_SH**: Starts with Magnetosheath is the region at which spacecraft recorded values at the start of Magnetosheath.
 - **S_SP**: Starts with Magnetosphere is the region at which spacecraft recorded values at the start of magnetosphere.

- **E_SW**: Ends with the solar wind is the region at which spacecraft recorded values at the end of solar wind.
- **E_SH**: Ends with Magnetosheath is the region at which spacecraft recorded values at end start of Magnetosheath.
- **E_SP**: Ends with Magnetosphere is the region at which spacecraft recorded values at the end of magnetosphere.

For an inbound the first event that will occur is a Bow Shock and later followed by Magnetopause. But in the case of an outbound direction the first event that occur will be a Magnetopause and later followed by a Bow Shock. Region of sampling at the start of any data gap will have a S_SW, S_SH and S_SP direction type and Region of sampling at the end of data gap E_SW, E_SH and E_SP. Dimension of this dataset is (480,10).

3.1.2 Dataset 2: Magnetometer Dataset

In this dataset it contains the magnetometer data of Cassini Spacecraft during the year 2005. Time difference between each data points are one minute which means each data point represents the data of a particular minute. The data in this dataset are provided in the KSM (Kronocentric Solar Magnetospheric) Coordinate system which is a kind of Saturn centered Coordinate system where direction of X is from Saturn to the Sun and X-Z plane of the Coordinate system contains the Saturn centered axis of Magnetic Dipole ‘M’.

Some of the relevant variables in the dataset are:

- **X_KSM.km.**: This is the X coordinate point value of the spacecraft in KSM Coordinate System.
- **Y_KSM.km.**: This is the Y coordinate point value of the spacecraft in KSM Coordinate System.
- **Z_KSM.km.**: This is the Z coordinate point value of the spacecraft in KSM Coordinate System.
- **Timestamp.UTC.** : It is the timestamp at which data point was recorded by the Magnetometer.
- **DOY.UTC.** : It tells about the day at which the datapoint was recorded in the year 2005.
- **BX_KSM.nT.**: It is the x component of magnetic field strength in Amperes/meter.
- **BY_KSM.nT.**: It is the y component of magnetic field strength in Amperes/meter.
- **BZ_KSM.nT.**: It is the z component of magnetic field strength in Amperes/meter.
- **BTot.nT.** : It is the resultant vector of Bx, By and Bz

$$B_{Tot} = \sqrt{Bx^2 + By^2 + Bz^2} \quad (1)$$

This dataset contains 494683 rows and 12 columns.

3.2 Derived Datasets

This section contains the information regarding all the datasets that were derived from the Raw Datasets.

3.2.1 Dataset 3: Combined Data Of Dataset 1 And Dataset 2

This is a newly created dataset by merging dataset 1 and dataset 2, so that we can understand the Magnetic field properties during the events like Magnetopause and Bow Shock. For merging the two datasets, the date and time variables were used as the key because that is common variable in both the dataset. Before joining these dataset it is necessary to format the date in Dataset 1. The `doy_cross` variable was selected and added "2004-12-31" date to that variable. By doing so it will generate a date with respect to the reference date. In dataset 2 the variable `Timestamp.UTC.` is in string format inorder to convert it into a data format I used `as.POSIXct()` function with `format="%d/%m/%Y %H:%M"`. Later the `Timestamp.UTC.` variable was converted into a new format which can be used as a key that is similar to the date created in Dataset 1 and which is then stored in the `date` variable. Two new variables were created in the Dataset 2 from the newly created `date` variable known as `hour_cross` and `minute_cross`. The `left_join()` function was used for merging the two datasets by using the variables `date`, `hour_cross` and `minute_cross` which is common on dataset 1 and dataset 2. In the newly created dataset it contains all the variables of dataset 1 and dataset 2. The dimension of the newly created dataset is 494683 rows and 19 columns. Some of the variables are removed from the dataset because we know that dataset 1 has very less number of datapoints when compared to dataset 2 So, it is better remove the variables like `xcrosslist`, `ycrosslist`, `zcrosslist`, `year_cross`, `doy_cross`, `SCET.s.`, `doyfrac_cross`, `hour_cross` and `minute_cross` from the merged dataset. Data manipulations and Visualizations were done on this newly created dataset.

After removing some of the variables, now the modified dataset has 494683 rows and 16 columns.

3.2.2 Dataset 4: Average and Standard Deviation Dataset

This dataset was created after the exploratory data analysis done on Dataset 3. From the results of those analysis Standard deviation and Average value of the Total Magnetic field 15 minutes before and 15 minutes after of an event occurred data point are showing some pattern. `rowMeans()` function from the base package and `rowSds()` function from the `matrixStats` (Bengtsson et al. (2021)) package was used for creating this dataset.

Some of the Variables are

- `Avg_Lag_Bx.:` This is the Average of all the `Bx` values that were recorded 15 minutes before each datapoint.
- `Avg_Lag_By:` This is the Average of all the `By` values that were recorded 15 minutes before each datapoint.
- `Avg_Lag_Bz:` This is the Average of all the `Bz` values that were recorded 15 minutes before each datapoint.
- `Avg_Lag_BTot:` This is the Average of all the `BTot` values that were recorded 15 minutes before each datapoint .
- `SD_Lag_Bx. :` This is the Standard Deviation of all the `Bx` values that were recorded 15 minutes before each datapoint.
- `SD_Lag_By :` This is the Standard Deviation of all the `By` values that were recorded 15 minutes before each datapoint.
- `SD_Lag_Bz:` This is the Standard Deviation of all the `Bz` values that were recorded 15 minutes before each datapoint.
- `SD_Lag_BTot.:` This is the Standard Deviation of all the `BTot` values that were recorded 15 minutes before each datapoint.

- **Avg_Lead_Bx.:** This is the Average of all the **Bx** values that were recorded 15 minutes after each datapoint.
- **Avg_Lead_By:** This is the Average of all the **By** values that were recorded 15 minutes after each datapoint.
- **Avg_Lead_Bz:** This is the Average of all the **Bz** values that were recorded 15 minutes after each datapoint.
- **Avg_Lead_BTot:** This is the Average of all the **BTot** values that were recorded 15 minutes after each datapoint.
- **SD_Lead_Bx. :** This is the Standard Deviation of all the **Bx** values that were recorded 15 minutes after each datapoint.
- **SD_Lead_By :** This is the Standard Deviation of all the **By** values that were recorded 15 minutes after each datapoint.
- **SD_Lead_Bz:** This is the Standard Deviation of all the **Bz** values that were recorded 15 minutes after each datapoint.
- **SD_Lead_BTot.:** This is the Standard Deviation of all the **BTot** values that were recorded 15 minutes after each datapoint.

4 Data Manipulation Section

This section will explain about the data manipulation that was done on the merged dataset (Dataset 3). There were lots of NA values in different predictors, so it was important to impute these values before using it for training the models.

4.1 Removing Time Dependency

Since all the datapoints were recorded by the spacecraft using the instruments over time so, there can be a time dependency. To remove the time dependency, I made the dataset wider which means a thirty-minute window was used for each data point and stored the magnetic field strength values and position of the spacecraft at each minute as a column for each row. Now for each data point there are 219 columns. I have labelled each column in the format (predictor_name{minute_index}). For example, the BX_KSM16 represents the BX_KSM value after one minute of the selected datapoint. All time-dependency variables like `Timestamp.UTC`, `doy_cross`, `time` etc were removed.

4.2 Data Imputation

In this dataset there are many NA values in different predictors like `type_cross` and `dirn_cross`. Since `type_cross` and `dirn_cross` are both categorical variables So, it is critical to impute the NA values with relevant short terms. `type_cross` variable represents the type of event at which the spacecraft crossed. Currently `type_cross` variable has values MP, BS, DG and SCAS which represents Magnetopause, Bow shock, Data gap and Unreliable data. All the other data than the above-mentioned categories in the newly created dataset can be categorized as NE which means No Events Occurred.

For `dirn_cross` variable, which represents the direction at which spacecraft is moving. This variable has the categories E_SH, E_SP, E_SW, I, O, S_SH S_SP and S_SW. So, I have imputed all the datapoints which has NA values in `dirn_cross` as UD (Unknown Direction) which means the direction of the spacecraft when that datapoint was recorded is Unknown.

5 Exploratory Data Analysis

The dataset contains the records of more than 490000 magnetometer readings with labels of type of crossing, Magnetic field strength values of fifteen minutes before and after of a datapoint and position of the spacecraft at which the data was recorded. The orbits of the spacecraft covered almost all local hours and gave sufficient dayside coverage. Before getting into further analysis its important to understand whether the data is imbalanced or not. The dataset contains enough datapoints for training different models. Understanding each variables on data is required before training the Models for classifying the events.

To understand the inbound and outbound boundary crossings we are going to consider a particular timeframe at which spacecraft crossed all the boundaries. From Jackman, Thomsen, and Dougherty (2019) It is stated that from 72th day to 74th day of the year 2005 Cassini Spacecraft was in Outbound on the dawn flank. Figure shown below is the Line graph of Total Magnetic Field recorded during the period of 72 to 74th day of the year 2005. In the Figure below the blue dotted line represents the point at which spacecraft crossed Bow Shock boundary and the red dotted line represents the Magnetopause. The package `ggplot` is used for the data exploration in this project.

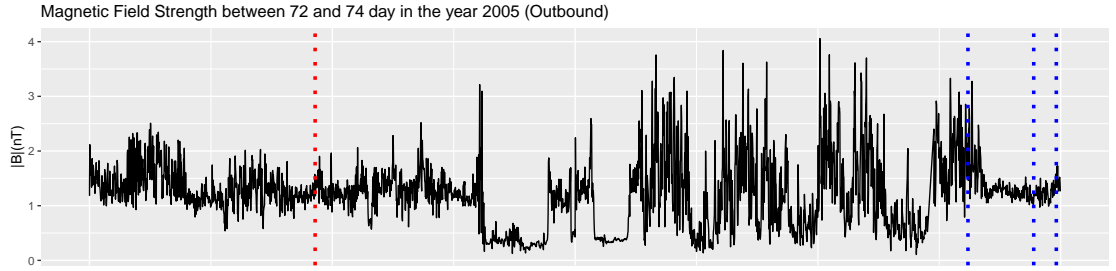


Figure 2: Line Graph of Total Magnetic Field Strength between 72 and 74th day of the year 2005

From Figure 2, we can clearly see that when the spacecraft approached the Bow Shock boundary there is a large value fluctuation of total magnetic field but when it comes to Magnetopause event there are only small value fluctuations. Since this data was recorded during the Outbound, the first boundary that was crossed by the spacecraft was Magnetopause and later followed by the Bow shock Boundaries.

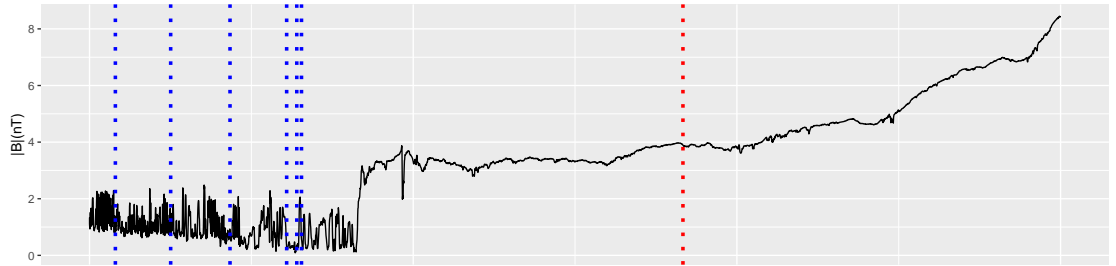


Figure 3: Line Graph of Total Magnetic Field Strength between 136 and 138th day of the year 2005

During the Inbound prenoon of Cassini Spacecraft which is between 136th and 138th day of the year 2005, the spacecraft observed some clean bow shock crossings on 136th day and followed by the Magnetopause Crossings on 137th day of the year 2005. Figure Shown below is the Magneticfield data between those days and the events like Bow Shock and Magnetopause are

marked with blue and red color respectively.

From Figure 3, we can clearly see that first boundary crossed by the spacecraft was Bow shock and later followed by the Magnetopause. There are very large fluctuations during when it crossed the Bow Shock boundary but in the case of Magnetopause region the total magnetic field strength was kind of constant.

5.1 Trajectory of Spacecraft

It is crucial to analyze the effect of Positions of Spacecraft in predicting the events based on this dataset, so that we can understand the significance of that variable.

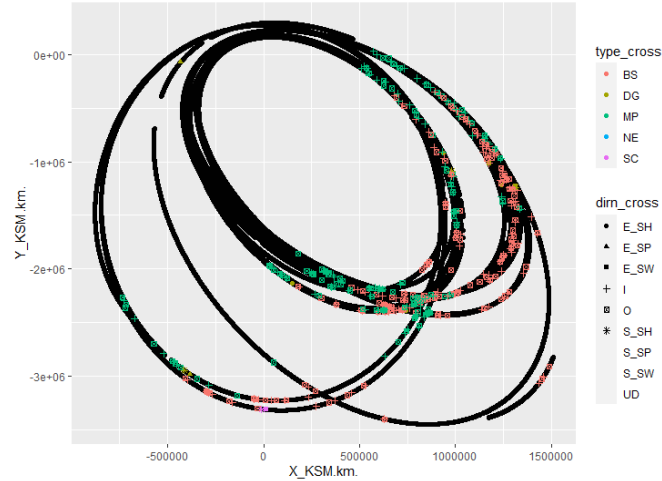


Figure 4: Boundary Crossing points on the Trajectory of Cassini Space craft

From the above plot we can clearly say that the Bow Shock events are in the Outermost orbit on which the spacecraft traveled and all the Magnetopause boundary crossings are in the tail of those orbits. Most of the Bow Shock boundary crossing events has high $X_KSM.km.$ value which is greater than 500000 when compared to other events. The $Y_KSM.km.$ value for Bow Shock events mostly lies in between 0 and -2×10^{-3} . The plot also gives information about the range of $X_KSM.km.$ and $Y_KSM.km.$ when 90% of boundary crossing happened which is $(0, 11 \times 10^5)$ and $(-2 \times 10^6, 0)$.

5.2 Examining the Data Imbalance

It is always important to check whether the given data is balanced or not. If it is not balanced that means there is a large number of particular class and fewer data points for all other classes of data. This imbalance will makes the model that we wants to train biased to that set of data.

Event Class	Count
SCAS (unreliable)	2
Data Gap	15
Magnetopause	203
Bow Shock	245
No Events Occurred	494127
Total	494592

Table 1: Number of Datapoints in each Boundary Class

From Table 1, it is clear that 90% of the data points are in the No Events Occurred Class and there are only very few data in the Bow shock and Magnetopause class. So a method must be adopted to prepare the training dataset for model training. Since there are only a very few data points which are in DG and SCAS class we can remove those data points because both classes gives unreliable data.

5.3 Proportion of Direction of Cross in Different Classes

The `dirn_type` variable explains at which direction the spacecraft was moving when it took the measurements. To get more insights on which direction did the most boundary crossings were recorded.

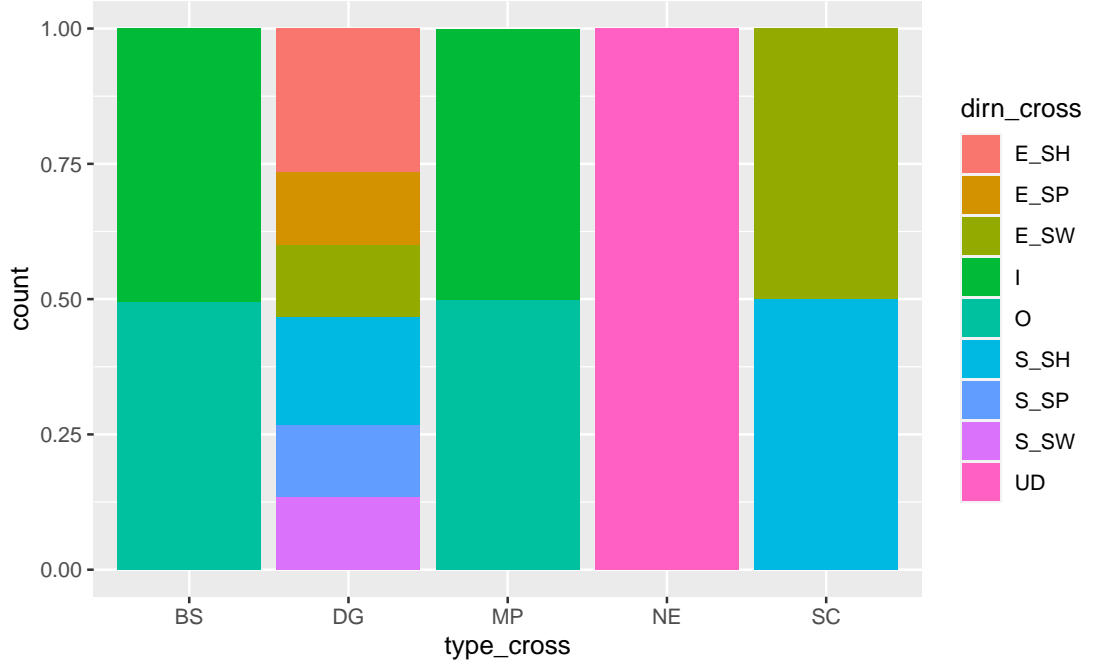


Figure 5: Proportion of different directions on each Class

From Figure 5, we can see that 50% of both the Bow Shock and Magnetopause events are recorded during Inbound and remaining 50% during the Outbound of the Spacecraft. But for all No events occurred class the direction of cross is Unknown. All other directions like starting from Magnetosheath, Starting from solar wind, starting from Magnetosphere etc are in SCAS and Datagap class. All datapoints with SCAS class are unreliable data and SCAS data was recorded during when the spacecraft starts with Magnetosheath and Ends with Solar wind. All the data points that were recorded when spacecraft was in the End of Magnetosheath, Starts with solar wind, End of Magnetosphere, End of Magnetosphere and starts with Magnetosphere are in Data Gap class. Since the direction of cross of all NE data points are Unknown or not available therefore the variable `dirn_cross` should not be used for prediction.

6 Significance of Predictors

Before Model development Significance of each variable must be evaluated, so that we can remove the unnecessary variables which will leads to over fitting of models. To find the significance of variables two Logistic Regression Models were used. Logistic regression Models can be fit to the data by using the maximum likelihood technique. The **family** of the models used was set to **binomial** because we are trying to classify only two classes.

Train and Test Dataset

Since the Dataset is a highly imbalanced one with 90% of the data has the class **No Events Occurred**. It is crucial to train the model with a balanced dataset, so for that we sampled 100 datapoints of each class by using **sample_n()** function for Dataset 4 and 300 NE Points were chosen instead of 100 for the model with Dataset 3 because if predictor count is nearly as big as total data points then the linear regression is too flexible and overfits the data. All other datapoints were chosen as Test Dataset for this model.

Standardizing the Train and Test Dataset

Standardizing is a techniques used in the Machine Learning. The main aim of Normalization is to make all the numeric columns in the dataset to a common scale. All the numeric variables in the train dataset was Normalized by using the **scale()** function. The standard deviation and mean of each variable of train dataset was stored separately. The test dataset was then Standardized by using the standard deviation and Means that were stored before for each variable. For Standardizing I have used the below formula:

$$X' = \frac{(X - \mu)}{\sigma} \quad (2)$$

In this formula mu represents the mean and sigma represents the standard deviation of that variable in the train dataset. By standardizing features we are centering the datapoints to zero and making the standard deviation of value 1.

6.1 Logistic Regression Model 1: Bow Shock vs Other Events

In this model I tried to predict the Bow shock events by using different derived datasets. Since the datasets are highly imbalanced, it is important to sample equal number of classes from datasets to train the Logistic Regression model. Before splitting into training and Test Dataset, a new variable called **event_occured** was created and all Bow Shock events were stored as 1 and all the rest of the events as 0 in the **events_occured** variable.

6.1.1 Results

Models were trained with different derived datasets and later compared the results. Table 2 Shown below contains the information regarding the datasets used and count of variables that are significant which is extracted from the summary of the model. From the results of Models that were trained with different datasets, Table 2 indicates that all the predictor variables are significant.

Dataset Used	Significant Predictors	Total Predictors
Dataset 3	218	218
Dataset 4	25	25
Dataset 3 without Lead Variables	107	107

Table 2: Logistic Regression : Bow Shock Vs Other Events Results Table

Summary of Dataset 3

From Dataset 3, we have sampled 300 NE points, 100 Bow shock data points and 100 Magnetopause data points for the train data. Since there are 218 predictors in this dataset it is important to have more than 300 datapoints in the training dataset because if predictor count is nearly as big as total data points then the linear regression is too flexible and it will overfits the data. Table 3 shown below is the confusion matrix of Logistic Regression Model with Dataset 3.

##	Reference	
## Prediction	0	1
##	0 339058	83
##	1 154856	62

Accuracy	Sensitivity	Specificity	Pos Pred Rate	Neg Pred Rate
68.6	42.8	68.6	0.04	100

Table 3: Logistic Regression - Bow Shock Vs Other Events Results Table (Dataset 3)

From the Table 3, we can clearly see that the accuracy is about 68.6% . Eventhough it gives better accuracy, the sensitivity of the Model is very low. Accuracy is the percentage of datapoints that are classified correctly but in the case of the Sensitivity which is number of true positive predictions divided by the total number of positive predictions in this model. The sensitivity gives a value that explains about Bow Shock events that are correctly classified. 42.8 is the sensitivity of the above model,so, 42.8 out of 100 Bow Shock events were classified correctly through this model. So, we can say that this Logistic Regression model is not a good one.

Summary of Dataset 4 : Average and Standard Deviation Data

In the case of this dataset there are only 24 predictors, so the train dataset contains 100 NE points, 100 BS Points and 100 MP points which was sampled from the Dataset 4. The Table 4 shown below is the Confusion matrix of the model with this dataset.

##	Reference	
## Prediction	0	1
##	0 492767	98
##	1 1347	47

Accuracy	Sensitivity	Specificity	Pos Pred Rate	Neg Pred Rate
99.7	32.4	99.7	3.372	100

Table 4: Logistic Regression : Bow Shock Vs Other Events Results Table (Dataset 4)

From Table 4, we can clearly see that the accuracy is about 99.7% . Eventhough it gives better accuracy than the Model with Dataset 3 but still the sensitivity of the Model is similar to that of the Model with Dataset 3. 32.4% of the Bow Shock boundaries were classified correctly. When compared to model that was trained with Dataset 3, the NE points were classified more accurately in this model.The specificity of this model is 99.7% which means the 99.7% of the NE Points

were classified correctly through this model.

Summary of Dataset 3 : Without Lead Variables

In this dataset there are only 107 predictors which means half of the variables were removed so, the train dataset contains 100 NE points, 100 BS Points and 100 MP points which was sampled from the Dataset 4. Table 5 shown below is the Confusion matrix of the model that is trained with Dataset 3(without lead variables). A dataset without Lead variables means all the columns which contains only the magnetic field information before 15 minutes of each datapoint.

##	Reference		
## Prediction	0	1	
##	0 447781	60	
##	1 46349	85	

Accuracy	Sensitivity	Specificity	Pos Pred Rate	Neg Pred Rate
99.7	58.6	90.6	0.183	100

Table 5: Logistic Regression : Bow Shock Vs Other Events Results Table (Dataset 3 without Lead variables)

From Table 5, we can clearly see that the accuracy is about 90.6% . It gives better accuracy and sensitivity than the Model trained with the Dataset 3. 58.6% of the Bow Shock boundaries were classified correctly. But when compared to Model that was trained with Dataset 3, non Bow shock Classes were classified more accurately in this model.The specificity of this model is 90.6% which means the 90.6% of the NE Points were classified correctly through this model.

6.2 Logistic Regression Model 2: Magnetopause vs Other Events

Unlike the previous model which classifies Bow Shock events from all other Events, This model classify Magnetopause event from rest of the events by using different derived datasets. Since the datasets are highly imbalanced It is important to sample equal number of classes from each dataset to train different Logistic Regression model. Before splitting into training and Test Dataset, A new variable called `event_occured` was created and all Magnetopause events were stored as 1 and all the rest of the events as 0 in that newly created variable.

6.2.1 Results

Logistic Regression models were trained with different derived datasets and later the results of each model was compared .Table 6 contains the information regarding the datasets used and count of Variables that are significant. A brief summary of models trained with different dataset was explained in this section.

Dataset Used	Significant Predictors	Total Predictors
Dataset 3	218	218
Dataset 4	25	25
Dataset 3 without Lead Variables	107	107

Table 6: Logistic Regression : Magnetopause Vs Other Events Results Table

Summary of Dataset 3

From Dataset 3, I have sampled 300 NE points, 100 Bow shock points and 100 Magnetopause Data points as the train data.Since there are 218 predictors in this dataset it is crucial to have

more than 300 dtapoints in the train dataset. If predictor count is nearly as big as total data points then the linear regression is too flexible and overfits the data. Table 7 shown below is the confusion matrix of this model.

##	Reference			
## Prediction	0	1		
##	0 352864	58		
##	1 141092	45		
	Accuracy	Sensitivity	Specificity	Pos Pred Rate
	71.4	43.7	71.4	0.032
				Neg Pred Rate
				100

Table 7: Logistic Regression : Magneto Pause Vs Other Events Results Table (Dataset 3)

From Table 7, we can clearly see that the accuracy is about 71.4% . Eventhough it gives a better accuracy value the sensitivity of the Model is very low. Accuracy is the percentage of datapoints that are classified correctly but in the case of the Sensitivity which is number of exact positive predictions divided by the total number of positive in this model the sensitivity gives a value that explains about Bow Shock events that are correctly classified . 43.7% is the sensitivity of the above model So, 44 out of 100 Bow Shock events were classified correctly through this model. When Comparing with the Logistic Regression Model that was used for Classifying the Bow Shock and Rest of the Events, this Logistic regression model was able to classify the Magnetopause events more efficiently by using Dataset 3.

Summary of Dataset 4 : Average and Standard Deviation Data

In the case of this dataset there are only 24 predictors so, 100 NE points, 100 BS Points and 100 MP points were sampled from the Dataset 4. The table Shown below is the Confusion matrix of the model with this dataset.

##	Reference			
## Prediction	0	1		
##	0 479437	7		
##	1 14719	96		
	Accuracy	Sensitivity	Specificity	Pos Pred Rate
	97	93.2	97	0.648
				Neg Pred Rate
				100

Table 8: Logistic Regression : Magneto Pause Vs Other Events Results Table (Dataset 4)

From Table 8, we can clearly see that the accuracy is about 97% . When comparing with the model that was trained with Dataset 3, this model did a vary good job in classifying the Magneto pause events. 93.2% of the Magnetopause boundaries were classified correctly. The NE points were also classified more accurately in this model. The specificity of this model is 97% which means the 97% of the NE Points were classified correctly through this model.

Summary of Dataset 3 : Without Lead Variables

In this dataset there are only 107 predictors which means half of the variables were removed so, the train dataset contains 100 NE points, 100 BS Points and 100 MP points from the Dataset 4. Table 9 Shown below is the Confusion matrix of the model with this dataset.

##	Reference			
## Prediction	0	1		
##	0 423862	59		

1 70310 44

Accuracy	Sensitivity	Specificity	Pos Pred Rate	Neg Pred Rate
85.8	42.7	85.8	0.063	100

Table 9: Logistic Regression : Magneto Pause Vs Other Events Results Table (Dataset 3 without Lead Variables)

From Table 9, we can clearly see that the accuracy is about 85.8% . It gives better accuracy and sensitivity than the Model with the Dataset 3 but less when compared to model with Dataset 4. 42.7% of the Magnetopause boundaries were classified correctly. The specificity of this model is 85.8% which means the 85.8% of the NE Points were classified correctly through this model.

6.3 Conclusion

Based on summary table of each model that was trained with different datasets we can say that all the variables are significant. Table 10 shown below explains the overall results obtained from different models.

Model Used	Dataset Used	Significant Predictors	Accuracy	Sensitivity
LR Model 1	Dataset 3	218	68.6	42.8
LR Model 1	Dataset 4	25	99.7	32.4
LR Model 1	Dataset 3 without Lead Variables	107	90.6	58.6
LR Model 2	Dataset 3	218	71.4	43.7
LR Model 2	Dataset 4	25	97	93.2
LR Model 2	Dataset 3 without Lead Variables	107	85.8	42.7

Table 10: Overall Results: Significance of Predictors

In Table 10 , LR Model 1 represents the Logistic Regression Model that classifies Bow Shock Events from all other events and LR Model 2 represents the Logistic Regression Model that classifies Magneto Pause events from all other events. From the table we can conclude that the Models that were trained with Dataset 4 and Dataset 3 (without Lead Variables) performed better than the model that is trained with Dataset 3 .

7 Classification Model Development

This section explains about the model that was developed for the classification of different boundaries based on the Magnetometer data. In the Predictor Significance section, we found that Dataset 4(Average Dataset) and Dataset 3(without lead variables) performed better when compared to Dataset 3. The results of classification models that were trained with different datasets was explained in this section and based on results a final model will be selected that can be used to address the problem statement. The model with high sensitivity values and better accuracy will be considered as the best Model. Because sensitivity value of each class will tell us the percentage of each correctly classified category . Several methods can be used for classifying the boundaries by using these kind of datasets, but in this project we have used Random forest model and Lasso regression Model.

7.1 Random Forest Model

This section explains about the three Random forest Models that were trained by using three different datasets. Random Forest Model is a Supervised Learning algorithm where it builds an ensemble of decision trees. Random forest produces great results by handling large datasets with higher dimensionality. Additional randomness will be added to the model by the random Forest algorithm, while growing the trees. It looks for the best predictor among a random set of predictors which will generally results in a better model.

7.1.1 Setting up Random Forest Model

In this project i have used the **ranger** (Wright et al. (2018)) package because it provides a faster implementation of random forest and also easy to tune the hyper parameters. Here the model uses following configuration:

- **num.trees** was set to 500. **num_trees** will determine how many trees the algorithm builds before it takes the voting or averages of the prediction. Higher the number of trees will usually gives better performance but makes the computation a lot slower.
- **verbose** which shows the computation status and Estimated runtime was set to TRUE
- **importance** was set to **impurity** which will give the variable importance. impurity measures the Gini Index for the classification.

7.1.2 Train and Test Dataset

Like we discussed in Section 5 the dataset is a highly imbalanced one with 90% of the data has the class **No Events Occurred**. It is important to train the model with a balanced dataset so, for that we sampled 100 datapoints of each class by using **sample_n()** function for Dataset 4. Both the train and test dataset was standardized by using the same technique that was discussed in the section 6.

7.1.3 Results

Summary of Dataset 3

From Dataset 3 I have sampled 300 NE points, 100 Bow shock data points and 100 Magnetopause Data points for the train data because there are 218 predictors in this dataset and If predictor count is nearly as big as total data points then the model becomes too flexible and overfits the data. Confusion Matrix of the model is shown below.

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    BS    MP    NE
##           BS    27     4  38199
##           MP     7    35  36829
##           NE   111    64 418783
##
## Overall Statistics
##
##           Accuracy : 0.8478
##           95% CI : (0.8468, 0.8488)
##           No Information Rate : 0.9995
##           P-Value [Acc > NIR] : 1
##
##           Kappa : 0.001
##
## Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##           Class: BS Class: MP Class: NE
## Sensitivity      1.862e-01 3.398e-01 0.848063
## Specificity      9.227e-01 9.254e-01 0.294355
## Pos Pred Value   7.063e-04 9.493e-04 0.999582
## Neg Pred Value   9.997e-01 9.999e-01 0.000972
## Prevalence       2.935e-04 2.085e-04 0.999498
## Detection Rate   5.465e-05 7.084e-05 0.847638
## Detection Prevalence 7.738e-02 7.463e-02 0.847992
## Balanced Accuracy 5.544e-01 6.326e-01 0.571209

```

The accuracy of the Random Forest Model with Dataset 3 is about 84.8% . The sensitivity value for each class which is 18.621% , 33.981% and 84.806% for the Bow Shock, Magnetopause and No Events occurred Class respectively. Among these values the sensitivity of the Bow Shock and Magnetopause class is very low.

Summary of Dataset 4 : Average and Standard Deviation Data

This dataset contains only 24 predictors so, 100 NE points, 100 BS Points and 100 MP points were sampled from the Dataset 4. The table shown below is the Confusion matrix of the model with this dataset.

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    BS    MP    NE
##           BS    56     8  60015
##           MP    12    48  50607
##           NE    77    47 383189

```

```

##
## Overall Statistics
##
##           Accuracy : 0.7758
##           95% CI : (0.7746, 0.777)
##       No Information Rate : 0.9995
##       P-Value [Acc > NIR] : 1
##
##           Kappa : 0.0013
##
## Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##           Class: BS Class: MP Class: NE
## Sensitivity      0.3862069 4.660e-01  0.77598
## Specificity      0.8784748 8.975e-01  0.50000
## Pos Pred Value   0.0009321 9.474e-04  0.99968
## Neg Pred Value   0.9997949 9.999e-01  0.00112
## Prevalence       0.0002935 2.085e-04  0.99950
## Detection Rate   0.0001133 9.715e-05  0.77559
## Detection Prevalence 0.1216029 1.026e-01  0.77584
## Balanced Accuracy 0.6323408 6.818e-01  0.63799

```

From the matrix we can clearly see that the accuracy is about 77.58% .38.621% of the Bow Shock boundaries and 46.602 % of Magnetopause boundaries were classified correctly through this model.

Summary of Dataset 3 : Without Lead Variables

In this dataset there are only 107 predictors which means half of the variables were removed so, 100 NE points, 100 BS Points and 100 MP points were sampled from the Dataset 3 as the training data. The table shown below is the Confusion matrix of the model with this dataset.

```

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    BS      MP      NE
##           BS      99      17 172553
##           MP      16      69 111779
##           NE      30      17 209695
##
## Overall Statistics
##
##           Accuracy : 0.4246
##           95% CI : (0.4232, 0.426)
##       No Information Rate : 0.9995
##       P-Value [Acc > NIR] : 1
##

```



```

##                      Kappa : 5e-04
##
##  McNemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##                      Class: BS Class: MP Class: NE
## Sensitivity           0.6827586 0.6699029 0.4244606
## Specificity           0.6507599 0.7737731 0.8104839
## Pos Pred Value        0.0005734 0.0006168 0.9997759
## Neg Pred Value        0.9998570 0.9999111 0.0007064
## Prevalence            0.0002934 0.0002084 0.9994983
## Detection Rate        0.0002003 0.0001396 0.4242476
## Detection Prevalence  0.3493379 0.2263194 0.4243427
## Balanced Accuracy     0.6667593 0.7218380 0.6174722

```

This model performed better in classifying Magnetopause and Bow shock events when compared to other random forest models that were trained with Dataset 3 and Dataset 4. The sensitivity of bow shock and magnetopause events are 68.276 % and 66.99 % respectively. But the overall accuracy is still very low which is only 42.459%.

7.2 Lasso Regression Model

This section explains about the three Lasso Regression Models that were trained by using three different datasets. Lasso Regression model is actually a modification of Linear Regression which performs L1 regularization. L1 regularization adds a penalty which is equal to the absolute values of regression coefficients and tries to decrease the value. When we compare it with ridge regression, Lasso regression will make coefficients that are responsible for large variance to zero. Lasso Model shrinks some of the coefficients to zero for each category those are very less significant on predicting that particular category.

7.2.1 Setting up Lasso Regression Model

In this project i have used the `glmnet()` (Friedman et al. (2021)) package for implementing the Lasso Regression Model because the algorithm is extremely fast and fits a generalize linear model with maximum likelihood. Here the model uses following configuration:

- `family` was set to `multinomial` it extends the binomial when there is more than two classes. Since in our response variable has more than two levels.
- `alpha` is set to the value 1 for lasso regression. `alpha` is for the elastic net mixing parameter.
- `lambda` was set to `impurity` which will give the variable importance. `impurity` measures the Gini Index for the classification.

7.2.2 Train and Test Dataset

Like we discussed in Section 5 the dataset is a highly imbalanced one with 90% of the data has the class `No Events Occurred`. It is important to train the model with a balanced dataset so, for that we sampled 100 datapoints of each class by using `sample_n()` function for Dataset 4. Both the train and test dataset was standardized by using the same technique that was discussed in section 6. A new variable `event_occured` is introduced in the dataset which is a numerical

representation of classes. All the Bow Shock, Magneto pause and No Events occurred classes were encoded by the numbers 1,2 and 0 respectively.

7.2.3 Results

Summary of Dataset 3

From Dataset 3 I have sampled 300 NE points, 100 Bow shock data points and 100 Magnetopause Data points for the train data because there are 218 predictors in this dataset and If predictor count is nearly as big as total data points then the linear regression is too flexible and overfits the data. Confusion Matrix of the Model is shown below.

```
## Confusion Matrix and Statistics
##
##              y_te
## predictions_test  0      1      2
##              0 319857    85    70
##              1 108587    43     7
##              2  65517    17    26
##
## Overall Statistics
##
##              Accuracy : 0.6473
##              95% CI : (0.646, 0.6487)
##      No Information Rate : 0.9995
##      P-Value [Acc > NIR] : 1
##
##              Kappa : 2e-04
##
##      McNemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##              Class: 0  Class: 1  Class: 2
## Sensitivity          0.6475349 2.966e-01 2.524e-01
## Specificity          0.3750000 7.802e-01 8.674e-01
## Pos Pred Value       0.9995156 3.958e-04 3.966e-04
## Neg Pred Value       0.0005339 9.997e-01 9.998e-01
## Prevalence           0.9994982 2.934e-04 2.084e-04
## Detection Rate       0.6472100 8.701e-05 5.261e-05
## Detection Prevalence 0.6475236 2.198e-01 1.327e-01
## Balanced Accuracy    0.5112675 5.384e-01 5.599e-01
```

The accuracy of the Lasso Regression Model with Dataset 3 is about 64.7% . The sensitivity values are 29.655% , 25.243% and 64.753% for the Bow Shock, Magnetopause and No Events classes respectively. Among these values the sensitivity of the Bow Shock and Magnetopause class is very low which means this model failed to classify majority of the events correctly. The Classes 0,1 and 2 in the confusion matrix is actually representing the NE,BS and MP events.

Summary of Dataset 4 : Average and Standard Deviation Data

In the case of this dataset there are only 24 predictors so, 100 NE points, 100 BS Points and 100 MP points were sampled from the Dataset 4. The table Shown below is the Confusion matrix of the model with this dataset.

```
## Confusion Matrix and Statistics
##
##               y_te
## predictions_test    0      1      2
##               0 291559    70    51
##               1 122396    61    16
##               2  80006    14    36
##
## Overall Statistics
##
##               Accuracy : 0.5901
##               95% CI : (0.5888, 0.5915)
##       No Information Rate : 0.9995
##       P-Value [Acc > NIR] : 1
##
##               Kappa : 3e-04
##
## Mcnemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##               Class: 0  Class: 1  Class: 2
## Sensitivity          0.5902470 0.4206897 3.495e-01
## Specificity          0.5120968 0.7522345 8.381e-01
## Pos Pred Value       0.9995852 0.0004981 4.497e-04
## Neg Pred Value       0.0006271 0.9997740 9.998e-01
## Prevalence           0.9994982 0.0002934 2.084e-04
## Detection Rate       0.5899508 0.0001234 7.284e-05
## Detection Prevalence 0.5901956 0.2478162 1.620e-01
## Balanced Accuracy    0.5511719 0.5864621 5.938e-01
```

The accuracy of this Lasso Regression Model which was trained with Dataset 4 is about 59% . The sensitivity values are 42.069% , 34.951% and 59.025% for the classes Bow Shock, Magnetopause and No Events occurred respectively. When compared to model trained with Dataset 3, this model misclassified more NE points.

Summary of Dataset 3 : Without Lead Variables

In this dataset there are only 107 predictors which means half of the variables were removed so, the train dataset contains 100 NE points, 100 BS Points and 100 MP points which were sampled from the Dataset 3. The table shown below is the Confusion matrix of the model with this dataset.

```
## Confusion Matrix and Statistics
##
##               y_te
## predictions_test  0      1      2
##               0 288342    67    58
##               1 148253    57    22
##               2  57382    21    23
##
## Overall Statistics
##
##               Accuracy : 0.5836
##               95% CI : (0.5822, 0.585)
##       No Information Rate : 0.9995
##       P-Value [Acc > NIR] : 1
##
##               Kappa : 2e-04
##
## McNemar's Test P-Value : <2e-16
##
## Statistics by Class:
##
##               Class: 0  Class: 1  Class: 2
## Sensitivity          0.5837154 0.3931034 2.233e-01
## Specificity          0.4959677 0.6998968 8.838e-01
## Pos Pred Value       0.9995667 0.0003843 4.005e-04
## Neg Pred Value       0.0005978 0.9997456 9.998e-01
## Prevalence           0.9994982 0.0002934 2.084e-04
## Detection Rate       0.5834225 0.0001153 4.654e-05
## Detection Prevalence 0.5836755 0.3001305 1.162e-01
## Balanced Accuracy    0.5398416 0.5465001 5.536e-01
```

This model failed to classify majority of Magnet pause events when compared to other Lasso Regression models mentioned in this section. The accuracy of this Lasso Regression Model which was trained with Dataset 3 without lead variables is about 58.4% .

7.3 Overall Results

Different models were fitted by using the datasets that were derived from the raw datasets. All the models were trained by using the balanced datasets that were sampled from the derived dataset that was intended for the model. Rest of the data points in that dataset were used as the test Dataset to find the accuracy and Sensitivity of the model.

Model Used	Dataset Used	Accuracy	Sensitivity BS	Sensitivity MP
RF Model	Dataset 3	84.8	18.6	34
RF Model	Dataset 4	77.6	38.6	46.6
RF Model	Dataset 3 without Lead Variables	42.5	68.3	67
LSR Model	Dataset 3	64.7	29.7	25.2
LSR Model	Dataset 4	59	42.1	35
LSR Model	Dataset 3 without Lead Variables	58.4	39.3	22.3

Table 11: Overall Results: Model Selection

- RF :- Random Forest Model
- LSR:- Lasso Regression Model

From Table 11, we can say that some of the models have higher accuracy but there are lot of misclassification when it comes to the Magnetopause and Bow shock classes. It may be due to very less number of datapoints in that particular classes. Eventhough we took the dataset of the entire year 2005 the Cassini spacecraft crossed the boundaries of Magnetopause and Bow shock regions very few times. When compared to Lasso Regression Model we can say that Random Forest model performed better in terms of accuracy and Classification of NE Points. Random Forest model with Dataset 3 has better accuracy but they misclassified most datapoints that are of class Bow Shock and Magnetopause events.

8 Conclusion

This project aimed to classify the Bow shock and Magnetopause boundaries based on the Magnetometer data collected by Cassini spacecraft. New datasets were derived from the raw datasets which was explained in the Dataset explanation section. Data visualizations were done on all the derived datasets and those analysis shown that the dataset was highly imbalanced. In section 6 of this report, we have explained about the results of different Logistic Regression models that were used to find the significance of predictors in each datasets. Based on these results, **Dataset 4** performed better for both classifying the Bow Shock against other events and Magnetopause versus other events. But in the random forest models which were fitted with all the derived datasets for classifying three categories BS,MP and NE were not performed well especially in classifying the BS and MP events. The performance of Lasso regression models with **multinomial** family were very poor when compared to the random forest models.

Eventhough there was some magnetic field strength pattern before or after each type of events, it is hard to classify the boundary crossings with the dataset that is currently provided because it contains only very less datapoints of Bow Shock and magnetopause events in the dataset. It may be because we have only considered the magnetometer data of the year 2005 which itself consist about 494000 data points. Additional studies are required by using an extended dataset which consist of entire cassini space mission to classify the boundary crossings by using this dataset. Recurrent Neural Networks can be also adopted to classify these boundary crossings but it was outside the scope of this thesis.

9 References

- Achilleos, N, C Bertucci, CT Russell, GB Hospodarsky, AM Rymer, CS Arridge, ME Burton, et al. 2006. “Orientation, Location, and Velocity of Saturn’s Bow Shock: Initial Results from the Cassini Spacecraft.” *Journal of Geophysical Research: Space Physics* 111 (A3).
- Bengtsson, Henrik, Hector Corrada Bravo, Maintainer Henrik Bengtsson, TRUE LazyLoad, TRUE ByteCompile, and Statistics biocViews Infrastructure. 2021. “Package ‘matrixStats.’”
- Burkholder, B, PA Delamere, X Ma, MF Thomsen, RJ Wilson, and F Bagenal. 2017. “Local Time Asymmetry of Saturn’s Magnetosheath Flows.” *Geophysical Research Letters* 44 (12): 5877–83. <https://doi.org/10.1002/2017GL073031>.
- Friedman, Jerome, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Kenneth Tay, Noah Simon, and Junyang Qian. 2021. “Package ‘Glmnet.’” *CRAN R Repository*.
- Jackman, Caitriona M, Michelle F Thomsen, and Michele Karen Dougherty. 2019. “Survey of Saturn’s Magnetopause and Bow Shock Positions over the Entire Cassini Mission: Boundary Statistical Properties and Exploration of Associated Upstream Conditions.” *Journal of Geophysical Research: Space Physics* 124 (11): 8865–83.
- Masters, Adam, JA Slavin, GA DiBraccio, T Sundberg, RM Winslow, CL Johnson, BJ Anderson, and H Korth. 2013. “A Comparison of Magnetic Overshoots at the Bow Shocks of Mercury and Saturn.” *Journal of Geophysical Research: Space Physics* 118 (7): 4381–90.
- Pan, Dong-Xiao, Zhong-Hua Yao, Rui-Long Guo, Bertrand Bonfond, Yong Wei, William Dunn, Bin-Zheng Zhang, et al. 2021. “A Statistical Survey of Low-Frequency Magnetic Fluctuations at Saturn.” *Journal of Geophysical Research: Space Physics* 126 (2): e2020JA028387.
- Pilkington, Nathan M, Nick Achilleos, Chris S Arridge, Adam Masters, Nick Sergis, Andrew J Coates, and Michele K Dougherty. 2014. “Polar Confinement of Saturn’s Magnetosphere Revealed by in Situ Cassini Observations.” *Journal of Geophysical Research: Space Physics* 119 (4): 2858–75. <https://doi.org/10.1002/2014JA019774>.
- Pilkington, NM, N Achilleos, CS Arridge, P Guio, A Masters, LC Ray, N Sergis, MF Thomsen, AJ Coates, and MK Dougherty. 2015. “Asymmetries Observed in Saturn’s Magnetopause Geometry.” *Geophysical Research Letters* 42 (17): 6890–8. <https://doi.org/10.1002/2015JA021290>.
- Sergis, N, CM Jackman, A Masters, SM Krimigis, MF Thomsen, DC Hamilton, DG Mitchell, MK Dougherty, and Andrew J Coates. 2013. “Particle and Magnetic Field Properties of the Saturnian Magnetosheath: Presence and Upstream Escape of Hot Magnetospheric Plasma.” *Journal of Geophysical Research: Space Physics* 118 (4): 1620–34. <https://doi.org/10.1002/jgra.50164>.
- Sulaiman, AH, X Jia, N Achilleos, N Sergis, DA Gurnett, and WS Kurth. 2017. “Large-Scale Solar Wind Flow Around Saturn’s Nonaxisymmetric Magnetosphere.” *Journal of Geophysical Research: Space Physics* 122 (9): 9198–9206. <https://doi.org/10.1002/2017JA024595>.
- Thomsen, Michelle F, AJ Coates, CM Jackman, N Sergis, X Jia, and KC Hansen. 2018. “Survey of Magnetosheath Plasma Properties at Saturn and Inference of Upstream Flow Conditions.” *Journal of Geophysical Research: Space Physics* 123 (3): 2034–53.
- Wright, Marvin N, Stefan Wager, Philipp Probst, and Maintainer Marvin N Wright. 2018. “Package ‘Ranger.’”

10 Appendix

10.1 Supporting code & Supporting Files Link

GitHub Repository: <https://github.com/matty1203/BowShockClassifier>

Dataset OneDrive Link: [https://maynoothuniversity-my.sharepoint.com/:f/r/personal/mathews_philipvenattu_2021_mumail_ie/Documents/Thesis%20\(20250487\)%20Dataset?csf=1&web=1&e=AV79IL](https://maynoothuniversity-my.sharepoint.com/:f/r/personal/mathews_philipvenattu_2021_mumail_ie/Documents/Thesis%20(20250487)%20Dataset?csf=1&web=1&e=AV79IL)

SUPPORTING FILES: https://maynoothuniversity-my.sharepoint.com/:u/g/personal/mathews_philipvenattu_2021_mumail_ie/EX-kFO26xqxDtPuKHDgyW-cB7Ma5-g4OHuwQYag9Xmo0jw?e=e9rV5l

Including Libraries and Reading the Datasets

```
suppressMessages(library(ggplot2))
suppressMessages(library(dplyr))
suppressMessages(library(tidyverse))
suppressMessages(library(caret))
suppressMessages(library(glmnet))
suppressMessages(library(ranger))
suppressMessages(library(MASS))
### Reading the Raw Datasets from local
raw_data_mag<-readRDS('./Data/dataset_version_0
                    /Full_Cassini_Master_MP_BS_CMJ_revised2005.rds'
                    ,refhook = NULL)
raw_data_all<-readRDS('./Data/dataset_version_0/Cass_data2005.rds'
                    , refhook = NULL)
```

Joining two Datasets by using the Timestamp

```
## Joining the two datasets
copy_data1<-raw_data_mag
copy_data2<-raw_data_all
##### Formatting Date in raw_data_mag
copy_data1$date<-as.Date(copy_data1$doy_cross,origin="2005-01-01")
head(copy_data1)
copy_data1$time<-paste(copy_data1$hour_cross,
                      copy_data1$minute_cross,"00",sep=":")
#####Extracting Date, hour cross and minute cross from the timestamp
copy_data2$TimeStamp<-as.POSIXct(copy_data2$Timestamp.UTC.,
                                format="%d/%m/%Y %H:%M")
copy_data2$date<-as.Date(format(copy_data2$TimeStamp, "%Y-%m-%d" ))
copy_data2$hour_cross<-as.integer(format(copy_data2$TimeStamp, "%H" ))
copy_data2$minute_cross<-as.integer(format(copy_data2$TimeStamp, "%M" ))
##### Merging with time variables and saving the
leftJoinDf <- left_join(copy_data2,copy_data1,by=c('date','hour_cross'
```



```

, 'minute_cross'))
leftJoinDf<-leftJoinDf%>%select(-c(xcrosslist,ycrosslist,zcrosslist));
saveRDS(leftJoinDf,"joined_data.rds")

#### Reading the saved dataset
raw_data_3<-readRDS('./Data/dataset_version_0/joined_data.rds', refhook = NULL)

```

Creating Dataset 3 (Full Dataset)

```

copy_data<-raw_data_3%>%select(-c(X,DOY.UTC.,SCET.s.,deltaTime.s.,
                                date,hour_cross,minute_cross,year_cross,doy_cross,
                                time,xcrosslist,ycrosslist,zcrosslist,doyfrac_cross));

## A new function created which takes dataset as input and saves the final
## processed dataset (It takes 1-2 days to completely process the dataset)
wideData_creator<-function(data_ip){
  data_created<-data.frame();
  i<-0;
  total<-(dim(data_ip)[1]-16)
  ####Starting from the 16th Row of the dataset
  for(data_ind in c(16:total )){
    #count restarts for each column in new row
    diff_min<-1;
    out <- paste0("Completed ",data_ind, "/",total)
    print(out)
    data_created<-dplyr::bind_rows(data_created,data_ip[data_ind,])
    #####Running a loop from 15 rows before to 15 rows after of a particular row
    for(time_ind in c((data_ind-15):(data_ind+15))){
      if(time_ind!=data_ind){
        #### Creating
        data_created[i,paste("X_KSM",diff_min,sep = "")]<-data_ip[time_ind,'X_KSM.km. ']
        data_created[i,paste("Y_KSM",diff_min,sep = "")]<-data_ip[time_ind,'Y_KSM.km. ']
        data_created[i,paste("Z_KSM",diff_min,sep = "")]<-data_ip[time_ind,'Z_KSM.km. ']
        data_created[i,paste("BX_KSM",diff_min,sep = "")]<-data_ip[time_ind,'BX_KSM.nT. ']
        data_created[i,paste("BY_KSM",diff_min,sep = "")]<-data_ip[time_ind,'BY_KSM.nT. ']
        data_created[i,paste("BZ_KSM",diff_min,sep = "")]<-data_ip[time_ind,'BZ_KSM.nT. ']
        data_created[i,paste("BTot_KSM",diff_min,sep = "")]<-data_ip[time_ind,'BTot.nT. ']
      }
      diff_min=diff_min+1;
      i=i+1; # New Data Index}
    print("Saving Data as a RDS File.....")
    saveRDS(data_created,"Full_Data_Wider.rds")
    print("Function Completed.....")
    return(data_created)
  }
}

wideData_creator(copy_data)
raw_widerDataset <-readRDS('Full_Data_Wider.rds', refhook = NULL)

### Removing all na values in the dataset and imputing the NA values in

```

```

##Direction of cross and Type of cross by 'UD' and 'NE' respectively.
raw_widerDataset$type_cross<-replace_na(raw_widerDataset$type_cross,"NE")
raw_widerDataset$dirn_cross<-replace_na(raw_widerDataset$dirn_cross,"UD")
raw_widerDataset<-na.omit(raw_widerDataset)
df <- subset(raw_widerDataset, select = -c(TimeStamp,TimeStamp.UTC.) )
df$type_cross=as.factor(df$type_cross)
saveRDS(df, "input_dataset.rds")

```

Creating Dataset 4 (Average and Standard Deviation Dataset)

```

dat<-readRDS('input_datasetFinal.rds', refhook = NULL)
cpy_data<-dat
###Using Row means and rowSDs for finding the mean and standard deviation
#### Below are the indexes of each columns which we use to take
##average and SD in the dataset
lag_col_index<-c(10,17,24,31,38,45,52,59,63,70,77,84,91,98,105)
lead_col_index<-c(112,119,126,133,140,147,154,161,168,175,183,190,197,204)
shortSD_Data<-data.frame(X_KSM=cpy_data[,1],Y_KSM=cpy_data[,2],
                        Z_KSM=cpy_data[,3],B_Tot=cpy_data[,4],
                        BX=cpy_data[,5],BY=cpy_data[,6],
                        BZ=cpy_data[,7],type_cross=cpy_data[,8],
                        dirn_cross=cpy_data[,9],
                        Avg_Lag_Bx=rowMeans(cpy_data[,lag_col_index]),
                        Avg_Lag_By=rowMeans(cpy_data[,lag_col_index+1]),
                        Avg_Lag_Bz=rowMeans(cpy_data[,lag_col_index+2]),
                        Avg_Lag_BTot=rowMeans(cpy_data[,lag_col_index+3]),
                        SD_Lag_Bx=rowSds(as.matrix(cpy_data[,lag_col_index])),
                        SD_Lag_By=rowSds(as.matrix(cpy_data[,lag_col_index+1])),
                        SD_Lag_Bz=rowSds(as.matrix(cpy_data[,lag_col_index+2])),
                        SD_Lag_BTot=rowSds(as.matrix(cpy_data[,lag_col_index+3])),
                        Avg_Lead_Bx=rowMeans(cpy_data[,lead_col_index]),
                        Avg_Lead_By=rowMeans(cpy_data[,lead_col_index+1]),
                        Avg_Lead_Bz=rowMeans(cpy_data[,lead_col_index+2]),
                        Avg_Lead_BTot=rowMeans(cpy_data[,lead_col_index+3]),
                        SD_Lead_Bx=rowSds(as.matrix(cpy_data[,lead_col_index])),
                        SD_Lead_By=rowSds(as.matrix(cpy_data[,lead_col_index+1])),
                        SD_Lead_Bz=rowSds(as.matrix(cpy_data[,lead_col_index+2])),
                        SD_Lead_BTot=rowSds(as.matrix(cpy_data[,lead_col_index+3]))))
saveRDS(shortSD_Data,"Average_SD_Data.rds")

```

Data Visualization Part

```

####Reading all the created datasets for future use
raw_data_4<-readRDS('./Data/dataset_version_2/input_datasetFinal.rds'
                    , refhook = NULL)
raw_data_5<-readRDS('./Data/dataset_version_2/Average_SD_Data.rds'
                    , refhook = NULL)
####Line Graph of Total Magnetic Field Strength between 72 and 74th day of

```

```

##the year 2005
d1<-as.Date(72,origin="2005-01-01")
d2<-as.Date(73,origin="2005-01-01")
temp<-filter(raw_data_3,between(date,d1,d2))
bowShock<-filter(temp,type_cross=="BS")$TimeStamp
MagP<-filter(temp,type_cross=="MP")$TimeStamp
ggplot(data=temp, aes(x=TimeStamp , y=abs(BTotal.nT.))) +
  geom_line()+
  ylab("|B| (nT)")+
  xlab("Timestamp")+
  geom_vline(xintercept=bowShock, linetype="dotted",colour="blue",size=1.3)+
  geom_vline(xintercept=MagP, linetype="dotted",colour="red",size=1.3)+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) + theme(legend.position = "none") +
  ggtitle("Magnetic Field Strength between 72 and 74 day in the year 2005 (Outbound)") +
  xlab("BTot")

###Line Graph of Total Magnetic Field Strength between 136 and 138th day of
##the year 2005
d1<-as.Date(136,origin="2005-01-01")
d2<-as.Date(138,origin="2005-01-01")
temp<-filter(raw_data_3,between(date,d1,d2))
bowShock<-filter(temp,type_cross=="BS")$TimeStamp
MagP<-filter(temp,type_cross=="MP")$TimeStamp
ggplot(data=temp, aes(x=TimeStamp , y=abs(BTotal.nT.))) +
  geom_line()+
  ylab("|B| (nT)")+
  xlab("Timestamp")+
  geom_vline(xintercept=bowShock, linetype="dotted",colour="blue",size=1.3)+
  geom_vline(xintercept=MagP, linetype="dotted",colour="red",size=1.3)+
  theme(axis.title.x=element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank())

###Boundary Crossing points on the Trajectory of Cassini Space craft
ggplot(raw_widerDataset, aes(x=X_KSM.km. , y=Y_KSM.km.))+
  geom_point(alpha=0.3)+
  geom_point(events_data, mapping =aes(x=X_KSM.km. ,
                                       y=Y_KSM.km. ,color=type_cross,shape=dirn_cross))

### Proportion of Direction of Cross in Different Classes
ggplot(data = raw_data_4) + geom_bar(mapping = aes(x = type_cross,
                                                  fill=dirn_cross)
                                     ,position = "fill")

```

Logistic Regression: Bow Shock vs Other Events- Summary of Dataset 3

```

copy_data<-raw_data_4
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
##Creating a Balanced Dataset for training the Model
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(300)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the Train dataset
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 217, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 217, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
### BS as 1 and all other events as 0
## Removing dirn_cross and creating new variable `new_trainData`.
new_trainData<-dplyr::select(cassini.train_scaled,-dirn_cross)
new_testData<-dplyr::select(cassini.test_scaled,-c(dirn_cross))
#### Training Data
cassini.logTrain<-new_trainData%>%mutate(event_occured=
  ifelse(type_cross=='NE'|type_cross=='MP',0,1))
cassini.logTrain$event_occured<-as.factor(cassini.logTrain$event_occured)
cassini.logTrain <- dplyr::select(cassini.logTrain,-c(type_cross))
#### Test Data
cassini.logTest<-new_testData%>%mutate(event_occured=
  ifelse(type_cross=='NE'|type_cross=='MP',0,1))
cassini.logTest$event_occured<-as.factor(cassini.logTest$event_occured)
cassini.logTest <- dplyr::select(cassini.logTest,-c(type_cross))
#### Model 1 Training and Confusion Matrix
LOGModel1<- glm(event_occured ~ ., family = "binomial", data = cassini.logTrain)
cassini.logTest$prob=predict(LOGModel1,cassini.logTest,type="response")
cassini.logTest$pred <- factor(ifelse(cassini.logTest$prob < .8, 0,1))
conf_mat_log1<-confusionMatrix(cassini.logTest$pred, cassini.logTest$event_occured,positive='

```

conf_mat_log1

Summary of Dataset 4 : Average and Standard Deviation Data

```
data_avg<-readRDS("./Data/dataset_version_2/Average_SD_Data.rds", refhook = NULL)
copy_data<-data_avg
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
##Creating a Balanced Dataset for training the Model
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(100)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
#### Scaling the Train dataset
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 23, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 23, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
## Logistic Regression Model 2 (Average)
new_trainData<-cassini.train_scaled
new_testData<-cassini.test_scaled
### BS as 1 and all other events as 0
#### Training Data
cassini.logTrain<-new_trainData%>%mutate(event_occured=
  ifelse(type_cross=='NE'|type_cross=='MP',0,1))
cassini.logTrain$event_occured<-as.factor(cassini.logTrain$event_occured)
cassini.logTrain <- dplyr::select(cassini.logTrain,-c(type_cross))
#### Test Data
cassini.logTest<-new_testData%>%mutate(event_occured=
  ifelse(type_cross=='NE'|type_cross=='MP',0,1))
cassini.logTest$event_occured<-as.factor(cassini.logTest$event_occured)
cassini.logTest <- dplyr::select(cassini.logTest,-c(type_cross))
#### Model Training and Confusion Matrix
```

```
LOGModel2<- glm(event_occured ~ ., family = "binomial", data = cassini.logTrain)
cassini.logTest$prob=predict(LOGModel2,cassini.logTest,type="response")
cassini.logTest$pred <- factor(ifelse(cassini.logTest$prob < .8, 0,1))
conf_mat_log2<-confusionMatrix(cassini.logTest$pred,
                               cassini.logTest$event_occured,positive='1')
conf_mat_log2
```

Summary of Dataset 3 : Without Lead Variables

```
copy_data<-raw_data_4[,1:107]
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
##Creating a Balanced Dataset for training the Model
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(100)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the Train dataset
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 107, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 107, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
## Logistic Regression Model 1 (Average)
new_trainData<-cassini.train_scaled
new_testData<-cassini.test_scaled
### BS as 1 and all other events as 0
#### Training Data
cassini.logTrain<-new_trainData%>%mutate(event_occured=
                                         ifelse(type_cross=='NE' | type_cross=='MP',0,1))
cassini.logTrain$event_occured<-as.factor(cassini.logTrain$event_occured)
cassini.logTrain <- dplyr::select(cassini.logTrain,-c(type_cross))
#### Test Data
cassini.logTest<-new_testData%>%mutate(event_occured=
```

```

                                ifelse(type_cross=='NE'|type_cross=='MP',0,1))
cassini.logTest$event_occured<-as.factor(cassini.logTest$event_occured)
cassini.logTest <- dplyr::select(cassini.logTest,-c(type_cross))
#### Model Training and Confusion Matrix
LOGModel2<- glm(event_occured ~ ., family = "binomial", data = cassini.logTrain)
cassini.logTest$prob=predict(LOGModel2,cassini.logTest,type="response")
cassini.logTest$pred <- factor(ifelse(cassini.logTest$prob < .8, 0,1))
conf_mat_log3<-confusionMatrix(cassini.logTest$pred
                                ,cassini.logTest$event_occured,positive='1')
conf_mat_log3$table

```

Logistic Regression Model 2: Magnetopause vs Other Events

Summary of Dataset 3

```

copy_data<-raw_data_4
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
##Creating a Balanced Dataset for training the Model
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(300)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 217, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 217, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
#### MP as 1 and all other events as 0
## Removing dirn_cross and creating new variable `new_trainData`.
new_trainData<-dplyr::select(cassini.train_scaled,-dirn_cross)
new_testData<-dplyr::select(cassini.test_scaled,-c(dirn_cross))
#### Training Data
cassini.logTrain<-new_trainData%>%mutate(event_occured=

```



```

                                ifelse(type_cross=='NE' | type_cross=='BS',0,1))
cassini.logTrain$event_occured<-as.factor(cassini.logTrain$event_occured)
cassini.logTrain <- dplyr::select(cassini.logTrain,-c(type_cross))
#### Test Data
cassini.logTest<-new_testData%>%mutate(event_occured=
                                ifelse(type_cross=='NE' | type_cross=='BS',0,1))
cassini.logTest$event_occured<-as.factor(cassini.logTest$event_occured)
cassini.logTest <- dplyr::select(cassini.logTest,-c(type_cross))
#### Model Training and Confusion Matrix
LOGModel1<- glm(event_occured ~ ., family = "binomial", data = cassini.logTrain)
cassini.logTest$prob=predict(LOGModel1,cassini.logTest,type="response")
cassini.logTest$pred <- factor(ifelse(cassini.logTest$prob < .8, 0,1))
conf_mat_log4<-confusionMatrix(cassini.logTest$pred
                                ,cassini.logTest$event_occured,positive='1')
conf_mat_log4$table

```

Summary of Dataset 4 : Average and Standard Deviation Data

```

data_avg<-readRDS("./Data/dataset_version_2/Average_SD_Data.rds", refhook = NULL)
copy_data<-data_avg
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
##Creating a Balanced Dataset for training the Model
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(100)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 23, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 23, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
## Logistic Regression Model 1 (Average)

```



```

new_trainData<-cassini.train_scaled
new_testData<-cassini.test_scaled
### MP as 1 and all other events as 0.
#### Training Data
cassini.logTrain<-new_trainData%>%mutate(event_occured=
                                     ifelse(type_cross=='NE'|type_cross=='BS',0,1))
cassini.logTrain$event_occured<-as.factor(cassini.logTrain$event_occured)
cassini.logTrain <- dplyr::select(cassini.logTrain,-c(type_cross))
#### Test Data
cassini.logTest<-new_testData%>%mutate(event_occured=
                                     ifelse(type_cross=='NE'|type_cross=='BS',0,1))
cassini.logTest$event_occured<-as.factor(cassini.logTest$event_occured)
cassini.logTest <- dplyr::select(cassini.logTest,-c(type_cross))
#### Model 1 Training and Confusion Matrix
LOGModel2<- glm(event_occured ~ ., family = "binomial", data = cassini.logTrain)
cassini.logTest$prob=predict(LOGModel2,cassini.logTest,type="response")
cassini.logTest$pred <- factor(ifelse(cassini.logTest$prob < .8, 0,1))
conf_mat_log5<-confusionMatrix(cassini.logTest$pred,
                               cassini.logTest$event_occured,positive='1')

conf_mat_log5$table

```

Summary of Dataset 3 : Without Lead Variables

```

copy_data<-raw_data_4[,1:107]
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
##Creating a Balanced Dataset for training the Model
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(100)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 107, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 107, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
}

```

```

temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
## Logistic Regression Model 1 (Average)
new_trainData<-cassini.train_scaled
new_testData<-cassini.test_scaled
#### Training Data
### MP as 1 and all other events as 0
cassini.logTrain<-new_trainData%>%mutate(event_occured=
                                     ifelse(type_cross=='NE'|type_cross=='BS',0,1))
cassini.logTrain$event_occured<-as.factor(cassini.logTrain$event_occured)
cassini.logTrain <- dplyr::select(cassini.logTrain,-c(type_cross))
#### Test Data
cassini.logTest<-new_testData%>%mutate(event_occured=
                                     ifelse(type_cross=='NE'|type_cross=='BS',0,1))
cassini.logTest$event_occured<-as.factor(cassini.logTest$event_occured)
cassini.logTest <- dplyr::select(cassini.logTest,-c(type_cross))
#### Model Training and Confusion Matrix
LOGModel2<- glm(event_occured ~ ., family = "binomial", data = cassini.logTrain)
cassini.logTest$prob=predict(LOGModel2,cassini.logTest,type="response")
cassini.logTest$pred <- factor(ifelse(cassini.logTest$prob < .8, 0,1))
conf_mat_log6<-confusionMatrix(cassini.logTest$pred,
                               cassini.logTest$event_occured,positive='1')
conf_mat_log6$table

```

Model Development(Random Forest): Summary of Dataset 3

```

copy_data<-raw_data_4
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t[!is.na(t$X_KSM16), ]
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
### Sampling a balanced data from dataset 4
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(300)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross))
               , mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 217, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 217, byrow = FALSE)

```

```

## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
## Removing dirn_cross and creating new variable `new_trainData`.
new_trainData<-dplyr::select(cassini.train_scaled,-dirn_cross)
new_testData<-cassini.test_scaled
#### Random forest Model Training and Confusion Matrix
cassini.rfmodel1 <- ranger(type_cross ~ ., data = new_trainData)
new_testData$pred<-predict(cassini.rfmodel1,data = new_testData)$predictions
conf_mat.rfmodel1<-confusionMatrix(new_testData$pred, new_testData$type_cross)
conf_mat.rfmodel1

```

Summary of Dataset 4(Average and Standard Deviation)

```

copy_data<-raw_data_5
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t[!is.na(t$Avg_Lead_Bz), ]
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
### Sampling a balanced data from dataset 4
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(300)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross))
  , mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 23, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 23, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
new_trainData<-dplyr::select(cassini.train_scaled,-dirn_cross)
new_testData<-cassini.test_scaled

```

```
#### Model 1 Training and Confusion Matrix
cassini.rfmodel2 <- ranger(type_cross ~ ., data = new_trainData)
new_testData$pred<-predict(cassini.rfmodel2,data = new_testData)$predictions
conf_mat.rfmodel2<-confusionMatrix(new_testData$pred, new_testData$type_cross)
conf_mat.rfmodel2
```

Summary of Dataset 3 : Without Lead Variables

```
copy_data<-raw_data_4[,1:107]
copy_data<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(100)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 107, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 107, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
new_trainData<-dplyr::select(cassini.train_scaled,-dirn_cross)
new_testData<-dplyr::select(cassini.test_scaled,-dirn_cross)
#### Model Training and Confusion Matrix
cassini.rfmodel3 <- ranger(type_cross ~ ., data = new_trainData,num.trees = 500
  ,importance = 'impurity')
new_testData$pred<-predict(cassini.rfmodel3,data = new_testData)$predictions
conf_mat.rfmodel3<-confusionMatrix(new_testData$pred, new_testData$type_cross)
conf_mat.rfmodel3
```

Lasso Regression Model Summary of Dataset 3

```
copy_data<-raw_data_4
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t[!is.na(t$X_KSM16), ]
copy_data$type_cross<-factor(copy_data$type_cross)
```

```

copy_data$id<-seq.int(nrow(copy_data))
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(150)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 218, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 218, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
cassini.train_scaled<-cassini.train_scaled%>%mutate(event_occured=ifelse(type_cross=='NE',
  0,ifelse(type_cross=='BS',1,
    ifelse(type_cross=='MP',2,0))))
cassini.train_scaled$event_occured<-as.factor(cassini.train_scaled$event_occured)
cassini.train_scaled <- dplyr::select(cassini.train_scaled,-c(type_cross,dirn_cross))

cassini.test_scaled<-cassini.test_scaled%>%mutate(event_occured=ifelse(type_cross=='NE',
  0,ifelse(type_cross=='BS',1
    ,ifelse(type_cross=='MP',2,0))))
cassini.test_scaled$event_occured<-as.factor(cassini.test_scaled$event_occured)
cassini.test_scaled <- dplyr::select(cassini.test_scaled,-c(type_cross,dirn_cross))
# Setting alpha = 1 implements lasso regression
x_tr<-as.matrix(dplyr::select(cassini.train_scaled,-event_occured))
y_tr<-as.matrix(dplyr::select(cassini.train_scaled,event_occured))
x_te<-as.matrix(dplyr::select(cassini.test_scaled,-event_occured))
y_te<-as.matrix(dplyr::select(cassini.test_scaled,event_occured))
lambdas <- 10^seq(2, -3, by = -.1)
lasso_reg <- lasso_reg <- cv.glmnet(x_tr, y_tr, alpha = 1, lambda = lambdas
  , nfolds = 5,family="multinomial")
lambda_best <- lasso_reg$lambda.min # Best Lambda Selection
###Lasso
lasso_model <- glmnet(x_tr, y_tr, alpha = 1, lambda = lambda_best, family="multinomial")
predictions_train <- predict(lasso_model, s = lambda_best, newx = x_tr,type="class")
predictions_test <- predict(lasso_model, s = lambda_best, newx = x_te,type="class")

```

```
conf_mat.lsmode11<-confusionMatrix(table(predictions_test,y_te))
conf_mat.lsmode11
```

Summary of Dataset 4 : Average and Standard Deviation Data

```
data_avg<-readRDS("./Data/dataset_version_2/Average_SD_Data.rds", refhook = NULL)
copy_data<-data_avg
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t[!is.na(t$Avg_Lead_Bz), ]
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(150)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross))
  , mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 218, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 218, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
### Converting Each categories in Numerical Variables
cassini.train_scaled<-cassini.train_scaled%>%mutate(event_occured=ifelse
  (type_cross=='NE',0,ifelse(type_cross=='BS', 1
    ,ifelse(type_cross=='MP',2,0))))
cassini.train_scaled$event_occured<-as.factor(cassini.train_scaled$event_occured)
cassini.train_scaled <- dplyr::select(cassini.train_scaled,-c(type_cross,dirn_cross))
cassini.test_scaled<-cassini.test_scaled%>%mutate(event_occured=ifelse(type_cross=='NE'
  ,0,ifelse(type_cross=='BS',1
    ,ifelse(type_cross=='MP',2,0))))
cassini.test_scaled$event_occured<-as.factor(cassini.test_scaled$event_occured)
cassini.test_scaled <- dplyr::select(cassini.test_scaled,-c(type_cross,dirn_cross))
x_tr<-as.matrix(dplyr::select(cassini.train_scaled,-event_occured))
y_tr<-as.matrix(dplyr::select(cassini.train_scaled,event_occured))
x_te<-as.matrix(dplyr::select(cassini.test_scaled,-event_occured))
```



```

y_te<-as.matrix(dplyr::select(cassini.test_scaled,event_occured))
lambdas <- 10^seq(2, -3, by = -.1)
lasso_reg <- lasso_reg <- cv.glmnet(x_tr, y_tr, alpha = 1, lambda = lambdas
                                , nfolds = 5, family="multinomial")
lambda_best <- lasso_reg$lambda.min # Best Lambda Selection
###Lasso
lasso_model <- glmnet(x_tr, y_tr, alpha = 1, lambda = lambda_best, family="multinomial")
predictions_train <- predict(lasso_model, s = lambda_best, newx = x_tr,type="class")
#eval_results(y_train, predictions_train, train)
predictions_test <- predict(lasso_model, s = lambda_best, newx = x_te,type="class")
conf_mat.lsmodel2<-confusionMatrix(table(predictions_test,y_te))
conf_mat.lsmodel2

```

Summary of Dataset 3 : Without Lead Variables

```

copy_data<-raw_data_4[,1:107]
t<-copy_data[!(copy_data$type_cross=="DG" | copy_data$type_cross=="SC"),]
copy_data<-t
copy_data$type_cross<-factor(copy_data$type_cross)
copy_data$id<-seq.int(nrow(copy_data))
cassini.train_unscaled<-copy_data%>% filter(type_cross=='NE')%>%sample_n(150)%>%
  rbind(copy_data%>%filter(type_cross=='BS')%>%sample_n(100))%>%
  rbind(copy_data%>%filter(type_cross=='MP')%>%sample_n(100))
cassini.train_unscaled$type_cross<-as.factor(cassini.train_unscaled$type_cross)
cassini.test_unscaled<-copy_data[-cassini.train_unscaled$id,]
cassini.test_unscaled<-dplyr::select(cassini.test_unscaled,-id)
cassini.train_unscaled<-dplyr::select(cassini.train_unscaled,-id)
## Scaling the data
cassini.train_scaled<-cassini.train_unscaled%>%mutate_if(is.numeric,scale)
means <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross))
               , mean)
sds <- lapply(dplyr::select(cassini.train_unscaled,-c(type_cross,dirn_cross)), sd)
means <- matrix(unlist(means), ncol = 107, byrow = FALSE)
sds <- matrix(unlist(sds), ncol = 107, byrow = FALSE)
## Normalizing the TEST DATASET
cassini.test_scaled<-dplyr::select(cassini.test_unscaled,-c(type_cross,dirn_cross))
for(col in c(1:dim(cassini.test_scaled)[2])){
  temp<-(cassini.test_scaled[,col]-means[1,col])/sds[1,col]
  cassini.test_scaled[,col]<-as.matrix(temp)
  temp<-NULL}
cassini.test_scaled$type_cross<-cassini.test_unscaled$type_cross
cassini.test_scaled$dirn_cross<-cassini.test_unscaled$dirn_cross
cassini.train_scaled<-cassini.train_scaled%>%mutate(event_occured=ifelse(type_cross=='NE'
                                ,0,ifelse(type_cross=='BS',1
                                ,ifelse(type_cross=='MP',2,0))))
cassini.train_scaled$event_occured<-as.factor(cassini.train_scaled$event_occured)
cassini.train_scaled <- dplyr::select(cassini.train_scaled,-c(type_cross,dirn_cross))

```

```

cassini.test_scaled<-cassini.test_scaled%>%mutate(event_occured=ifelse(type_cross=='NE'
                                ,0,ifelse(type_cross=='BS',1
                                ,ifelse(type_cross=='MP',2,0))))
cassini.test_scaled$event_occured<-as.factor(cassini.test_scaled$event_occured)
cassini.test_scaled <- dplyr::select(cassini.test_scaled,-c(type_cross,dirn_cross))
# Setting alpha = 1 implements lasso regression
x_tr<-as.matrix(dplyr::select(cassini.train_scaled,-event_occured))
y_tr<-as.matrix(dplyr::select(cassini.train_scaled,event_occured))
x_te<-as.matrix(dplyr::select(cassini.test_scaled,-event_occured))
y_te<-as.matrix(dplyr::select(cassini.test_scaled,event_occured))
lambdas <- 10^seq(2, -3, by = -.1)
lasso_reg <- lasso_reg <- cv.glmnet(x_tr, y_tr, alpha = 1, lambda = lambdas
                                , nolds = 5,family="multinomial")
lambda_best <- lasso_reg$lambda.min # Best Lambda
###Lasso
lasso_model <- glmnet(x_tr, y_tr, alpha = 1, lambda = lambda_best, family="multinomial")
predictions_train <- predict(lasso_model, s = lambda_best, newx = x_tr,type="class")
predictions_test <- predict(lasso_model, s = lambda_best, newx = x_te,type="class")
conf_mat.lmodel3<-confusionMatrix(table(predictions_test,y_te))
conf_mat.lmodel3

```