Predicting Invariant Mass of an Electron

Richard M. Flores[1]

Abstract
The field of Data Science has evolved from statistical calculations to complex algorithms that grant insight into a variety of natural phenomena. Data Science can be accessed to provide information on everything from predicting the future price of corn to the classification of stars in distant galaxies. One application in which Data Science can be utilized is in the analysis of the Petabytes of information generated by the CERN Particle Accelerators in Geneva, Switzerland. In this research report we will analyze the data provided by the Compact Muon Solenoid as the particle accelerator attempts to trigger, reconstruct, and identify events with final state electrons.

Keywords
CERN — Particle Physics — Invariant Mass

[1] *Master of Science, Data Analytics, Western Governors University, United States*
*__Contact Author__: rflo147@wgu.edu

Contents

Introduction

This Research Project Assessment has been separated into 6 subsections which correlate to the sections listed under the grading rubric for Task 2: Data Analytics Report and Executive Summary.

1. Research Proposal

1.1 Research Question

This research project seeks to answer the question, can we predict the mass of an electron by analyzing dielectron events in the mass range 2-110 GeV and will our machine learning algorithm prove to be statistically significant in predicting mass in subatomic particle collisions?

1.2 Context

The concept of subatomic particles has been theorized ever since Democritus had the idea of atomos, or the indivisible blocks of matter, in early 5th century BCE. The concept of atomos, or atoms, has been further elaborated on by great thinkers including Galileo Galilei and Sir Isaac Newton. A modern study of subatomic particles really began in 1808 when John Dalton developed a relationship between the individual building blocks that compose all elements in the universe. With the advent of transistors, circuit chips, and computers the ability to calculate and observe natural phenomena has skyrocketed to previously unimaginable levels. One such place of intense subatomic study is the Conseil Européen pour la Recherche Nucléaire or as we call it, CERN.

The CERN research organization is home to the largest particle physics laboratory in the world. The CERN laboratory is also home to the Large Hadron Collider and the Compact Muon Solenoid (CMS) particle physics detectors. CMS is a

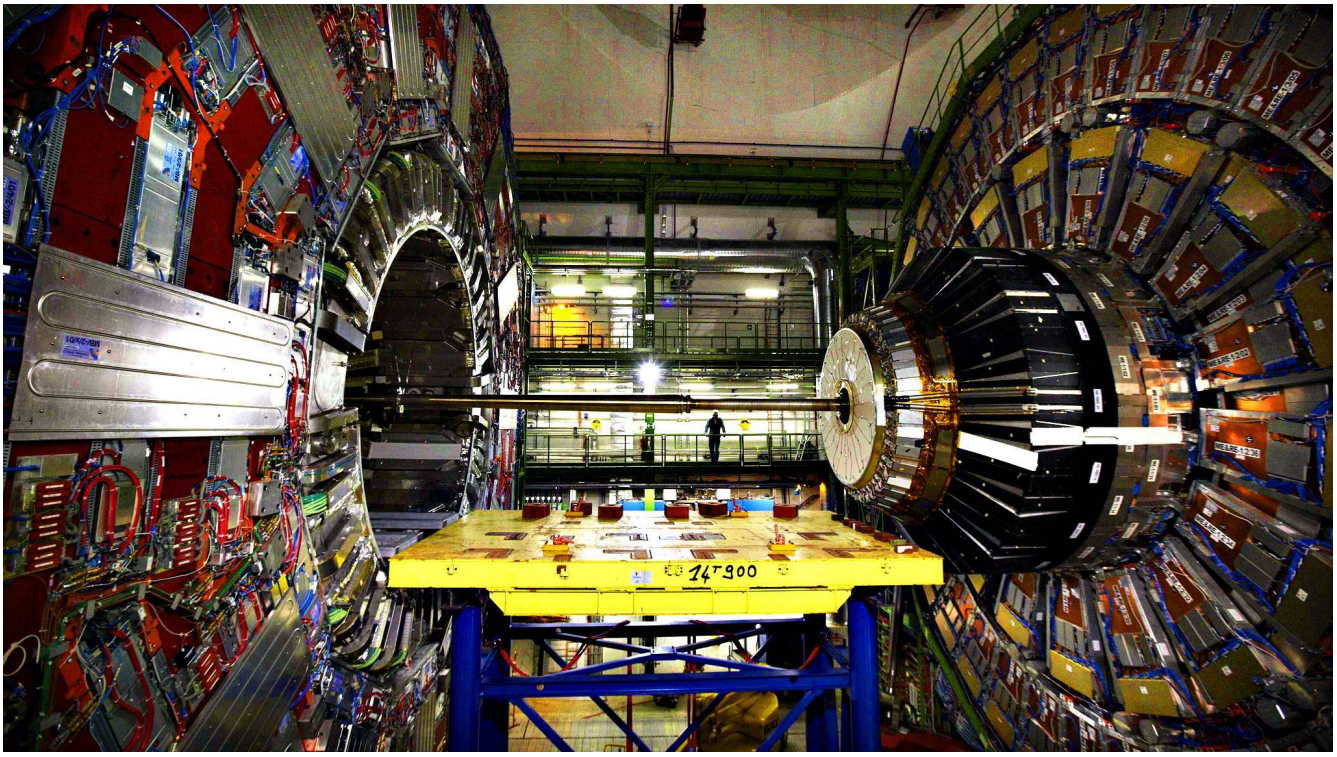

Figure 1. CMS at CERN

particle detector capable of generating enormous amounts of data on the observation of particle collisions at 0.9 - 13 TeV. With the relatively new field of Data Science, we are able to leverage our computing powers alongside sophisticated algorithms to quickly and efficiently observe and deride insights from mountains of data. In this analysis we will leverage the Data Science field, specifically the CatBoost open-source machine learning algorithm, to predict the invariant mass of an electron based on observations from particle collisions in the Compact Muon Solenoid detector.

1.3 Hypothesis

The proposed hypothesis of this research paper is that the observed components of momentum in electron collisions are statistically significant in determining the mass of subatomic particles.

1.4 Justification

A successful outcome of this data-analysis will approximately predict the invariant mass of an electron equal to the theoretically calculated mass. The prediction of the electron's invariant mass is based on observations including total energy, components of momentum, transverse momentum, pseudorapidity, phi angle, and charge. A successful outcome using the CatBoost machine learning algorithm will prove the reliability of the constructed model and is statistically significant in demonstrating the model's use in similar calculations including but not limited to proton collisions and further study of the Higgs Boson.

2. Data Collection

2.1 Description of Data

The dataset provided for investigation of this research project is provided courtesy of the CERN open data organization. The dataset titled "Events with two electrons from 2010" contains observations from the Compact Muon Solenoid (CMS) particle detector which recorded 100,000 dielectron events in the particle accelerator. The data is organized in a CSV spreadsheet and contains multiple recorded observations on the particle collisions including:

1. Run number of the event
2. Event number
3. Total energy of the electron (GeV)
4. Components of the momentum of the electron (GeV)
5. Transverse momentum of the electron (GeV)
6. Pseudorapidity of the electron
7. Phi angle of the electron
8. Charge of the electron
9. Invariant mass of two electrons (GeV)

The events in the dataset were derived and selected based on the presence of two electrons with invariant mass between 2 – 110 GeV. The shape of the dataset contains 19 columns/features and 100,000 rows/records totaling 14 MB of data.

2.2 Advantages and Disadvantages

The selected dataset has several advantages for the research project. One advantage is the dataset has been released under

the Creative Commons CC0 waiver which grants use under the public domain and is not subject to copyright law including related and neighboring rights. The CC0 waiver also allows the ability to copy, modify, distribute, and perform work even for commercial applications without the need to request permission. Another advantage is the integrity of the authoring organization CERN. As the CERN research organization is comprised of 23 member states and 2,600 of the brightest scientific and technical staff from more than 70 countries, we are ensured that our data is accurate, reliable, and the data released has been peer reviewed to be scientifically sound. Before the dataset was released, the information has been thoroughly analyzed and tested in simulation events to verify its accuracy. Thus, we can safely ensure our calculations provided in this research report and true and accurate. There are several disadvantages that come with any publicly available dataset. The biggest concern is the inability to administer control over observations and variables contained in the data. This disadvantage will require careful and thorough wrangling or data cleaning to prepare the information for analysis. Another disadvantage is that due to the age of the dataset and the complexity of the CERN organization, any questions about the data contained may prove to be difficult or answered in a timely manner for this analysis. Although there is a thriving scientific particle physics community online, this research project will rely on credible sources for the sake of authenticity of derived insights.

2.3 Overcoming Challenges

The biggest challenge of this research project is the selection of data analysis techniques and machine learning models to properly evaluate our research goal of calculating the invariant mass of an electron from the CERN dataset. The Western Governors University Master of Science Data Analytics program has provided us data analysts with a plethora of techniques for data analysis such as linear regression and classification as well as machine learning models such as predictive analysis with decision trees and time series. However, it is ultimately our responsibility to take the data analytics training and carefully select the appropriate techniques for analysis as well as the initiative to research and select new and cutting-edge algorithms and models to enable us to provide a complete and insightful analytic report. To overcome this challenge, we will need to carefully and methodically select from our available toolset the most applicable and meaningful analysis tools to provide an insightful outcome that aligns with the research question.

3. Data Extraction and Preparation

3.1 Data Cleaning and Preparation Code

This section will describe the process of examining, cleaning, and preparing the CMS dataset for further analysis. The data has been prepared in Jupyter notebook and extracted for review in the research report. The process for each step in the preparation process has been listed below with a breakdown detailing an Explanation, Justification, and one Advantage and Disadvantage of each selected method.

```
// Import Standard Data Science Libraries

# This is a comment
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Explanation We begin the process by importing the necessary and critical standard Python libraries for Data Analysis; Pandas and NumPy. We also import Seaborn and Matplotlib for visualization.

Justification These libraries allow us to import, analyze, and manipulate the dataset as a Python dataframe.

Advantage Pandas and NumPy's greatest advantage is the simple yet effective language of the libraries which allow for an easy statistical overview of the dataset.

Disadvantage A disadvantage of using Python libraries is the use of these modules have a high learning curve to use effectively. In lieu of using Python for data cleaning and preparation we can use software like Tableau Prep to visualize and quickly edit items such as Null or NaN values.

```
// Data Preparation

# Import Dataset
CERN_DF = pd.read_csv("dielectron.csv")

# Rename and drop unnecessary features
CERN_DF = CERN.drop(["Run","Event"],axis=1)
CERN_DF.rename(columns={'px1 ':'px1'},
    inplace=True)

# Categorize Electron Charges
dataV["Q1"] = pd.Categorical(dataV["Q1"])
dataV["Q2"] = pd.Categorical(dataV["Q2"])
```

Explanation After loading the necessary libraries for this section of analysis, we can use Pandas to import the CERN dataset into a Pandas Dataframe. The 'Run' and 'Event' features are not necessary for analysis and can be removed as well as removing a space from the 'px1' feature for clarity.

Justification It is of paramount importance to import the CERN dataset into a Pandas dataframe for later analysis techniques. We have also categorized our 'Q1' and 'Q2' features.

Advantage There are numerous advantages to examination and manipulation of the dataset in a Pandas dataframe, such as the ease of exploratory analysis as well as the ability to visualize the data.

Disadvantage A disadvantage is that a similar process using database software such as PostgreSQL allows a user familiar with an database IDE such as PGAdmin may be easier depending on the amount of knowledge and experience with said software.

```
// Exploratory Analysis

# Print fundamental analytical data
print(data.shape)
print(data.info())
print(data.describe().T)

# Detect Duplicate Data
print(data.duplicated().value_counts())

# Check for Null and NaN values
print(data.isnull().sum())
print(data.isnull().all())

# Display heatmap of missing data
msno.bar(data,figsize=(8,5))
plt.show()

# Replace missing data with median value
data["M"].fillna(data["M"].median(),
inplace=True)
print(data.isnull().sum())
```

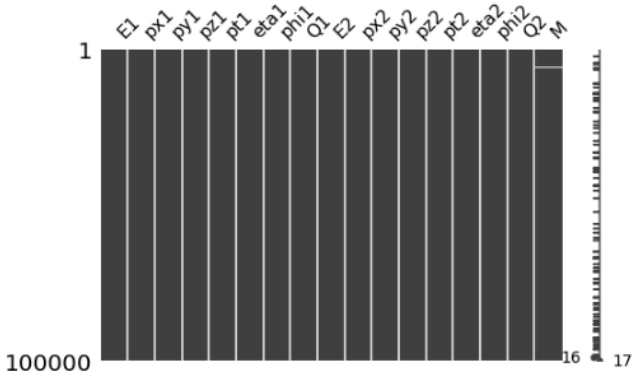

Figure 2. Graph of Missing Data

Explanation We begin our data exploration by examining the fundamental statistics of the CERN dataset. From the .shape function we have verified the dataframe contains 17 features and 100,000 records as expected. The .info function allows us to verify the data types which is essential for correctly inputting the proper values for later data analysis. From the .describe function we can view analytics such as count, mean, min, and max values of each feature.

Justification It is of critical important that we establish certain criteria in order for the analysis to function properly. For instance, in order for our machine learning algorithm to function properly we need to ensure we are inputting the correct variable types which have been verified using the .info function. We have also explored and corrected potential data issues such as missing values by imputing median values.

Advantage Using Pandas and NumPy are highly advantageous in preparing and cleaning the dataset. Using the mentioned libraries afforded us the opportunity to quickly and efficiently review and prepare the data for further analysis.

Disadvantage While Pandas and NumPy are excellent libraries there exist more sophisticated algorithms for analysis which may improve the final outcome, however it is likely that these advantages would show marginal improvement.

```
// Outliers

# Initialize a dataframe to detect outliers
cl = LocalOutlierFactor()
cl.fit_predict(DataForA)

# Representation of the opposite of LOF
    data samples
score = cl.negative_outlier_factor_

# Calculate largest jump
scoresorted = np.sort(score)
print(scoresorted[0:20])

# Print detected outlier Point and Row
point = scoresorted[4]
print(f"Outlier location: {point} ")
print("---"*20)
print("Row Detected:
    \n",DataForA[score==point])

# Clean outliers without record removal
CleanData = DataForA > point
print(data[CleanData])
```

Explanation The final section of data preparation includes the detection and removal of outliers. Results from the LocalOutlierFactor function has reported observer outliers in the px, py, pz, eta, and phi features. We were successful in cleaning outliers by reducing the detected values to the previously viewed max value in each category.

	E1	px1	py1	pz1	pt1	eta1	\
18749	850.60200	53.77760	115.639000	-840.9870	127.53200	-2.58506	
21447	838.83000	-250.58700	-87.965200	-795.6790	265.57800	-1.81719	
21789	3.40368	-1.87945	-0.143547	2.8341	1.88492	1.19674	
44885	770.58300	-22.68970	124.649000	760.0960	126.69700	2.49167	

	phi1	Q1	E2	px2	py2	pz2	pt2
18749	1.13550	1	2.90757	-2.08829	-1.129120	1.67872	2.37400
21447	-2.80399	-1	39.72040	-11.83550	4.618670	-37.63370	12.70480
21789	-3.06536	-1	948.37500	138.74500	69.961800	935.55800	155.38600
44885	1.75085	-1	18.39200	-4.32788	0.947612	17.85040	4.43041

	eta2	phi2	Q2	M
18749	0.658496	-2.645930	1	90.8602
21447	-1.806410	2.769530	1	40.3669
21789	2.495210	0.467041	-1	41.1658
44885	2.101740	2.926040	-1	27.8643

Figure 3. Sample Outlier Results

Justification We have selected to change the value of outliers as opposed to removing the record as there is a sufficiently low amount of outliers detected in order to justify this solution.

Advantage While there are a multitude of algorithms available for detecting and correcting outliers, the selected libraries are able to quickly and efficiently iterate over the 100,000 records to detect outlying values. It is fortunate that the CERN dataset contains a relatively low number of outliers and thus we are able to manipulate the value rather than remove the record from the dataset saving crucial information.

Disadvantage A disadvantage of the selected libraries for outliers is the threshold for detection, with a more sophisticated algorithm we may have opted to remove the record rather than modify the value however the trade-off comes at the cost of compute time which would drastically increase the amount of time needed for detection.

4. Analysis

4.1 Description of Analysis Techniques

For this research report, we will employ two major analysis techniques for the observation and calculation of the CERN dataset. The first analysis technique will involve visualization of the dataset to gain insights from the raw data. The second analysis technique will be our Machine Learning model to achieve the desired outcome of predicting the invariant mass of an electron through the observed dielectron events.

4.1.1 Data Visualization

```
// Correlation

# Create Pearson Correlation plot

correlatePearson =
    data.corr(method="pearson")
figure = plt.figure(figsize=(20,8))
sns.heatmap(corrPearson,annot=True,vmin=-1
,center=0, vmax=1)
plt.show()
```

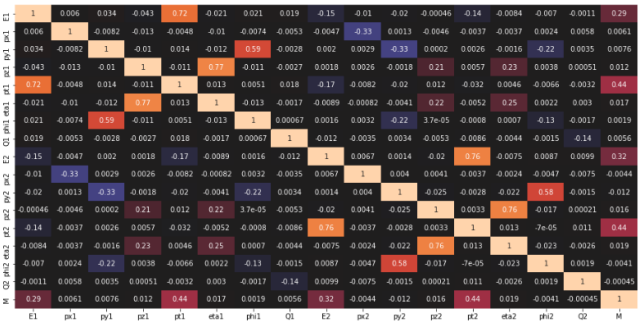

Figure 4. Pearson Correlation Heatmap

Figure 4 displays a Pearson Correlation Heatmap representing relationships and standardized values amongst variables.

```
// Covariance

# Create Covariance plot

figure = plt.figure(figsize=(20,8))
sns.heatmap(covv,annot=True,vmin=-1,center=0,vmax=1)
plt.show()
```

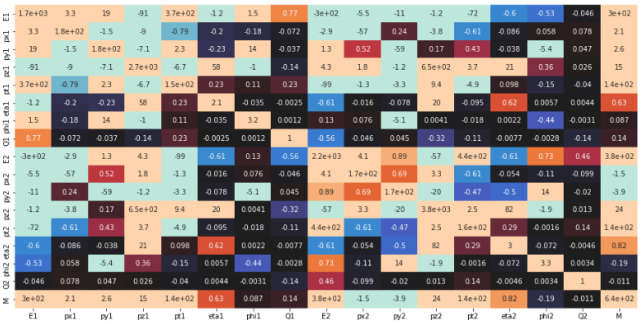

Figure 5. Pearson Covariance Heatmap

Figure 5 displays a Pearson Covariance Heatmap representing relationships and non-standardized values amongst variables.

```
// Histogram

# Create Histogram of Dataset

data.iloc[:,0:16].hist(figsize=(20,20))
plt.show()
```

Histogram in Fig. 6 contains visualizations of the 16 features being calculated in this analysis. The plots displayed show interesting observations in the ranges of each variable and helps assess which prediction model best suits the data.

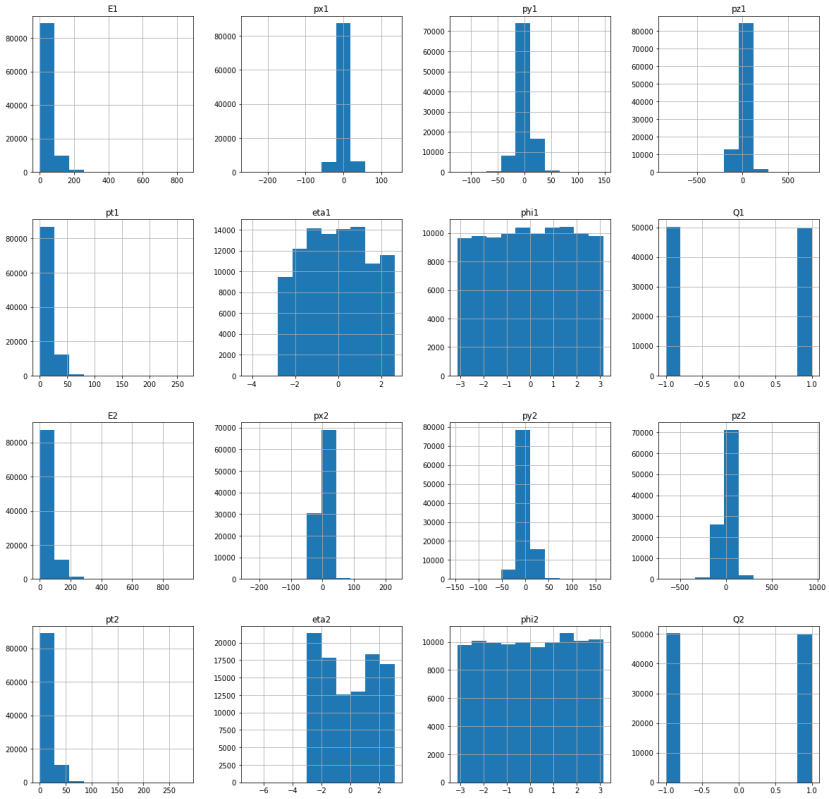

Figure 6. Histogram

```
// Display Scatterplot

# Example of scatterplot comparing E1 and M
    values
figure = plt.figure(figsize=(20,8))
sns.scatterplot(x="E1",y="M",data=data)
plt.show()
```

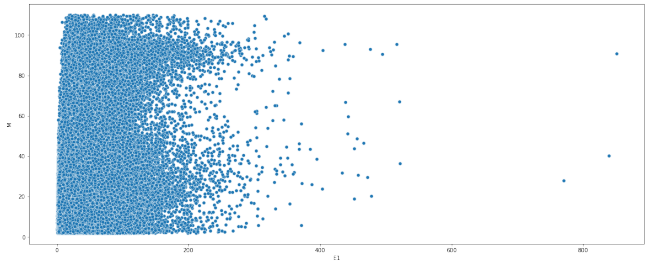

Figure 7. Scatterplot of E1 and M values

Figure 7. visualizes a scatterplot of E1 and M values. Similar scatterplots created display relationships between variables such as eta1 and M, phi1 and M, and the p-vector and M.

```
// Display Histoplot

# Example of Histplot generatored for E2
    values
figure = plt.figure(figsize=(20,5))
sns.histplot(
    data,
    x="E2",
    multiple="stack",
    edgecolor=".3",
    linewidth=.5,
    log_scale=True,)
plt.show()
```

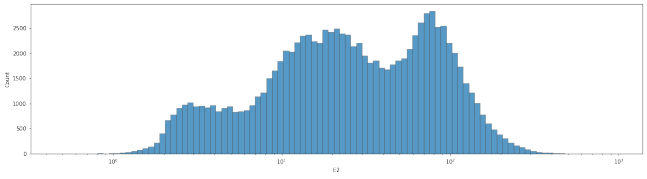

Figure 8. Histplot of E2 values

Figure 8. displays a Histplot of E2 values across the range of values for the variable. Similar plots have been created for eta2, phi2, and the p-vector.

```
// Display Jointplot

# Display Jointplot for E1 and ETA1 values
figure = plt.figure(figsize=(20,8))
sns.jointplot(x="E1",y="eta1",color="#4CB391",
data=data) plt.show()
```

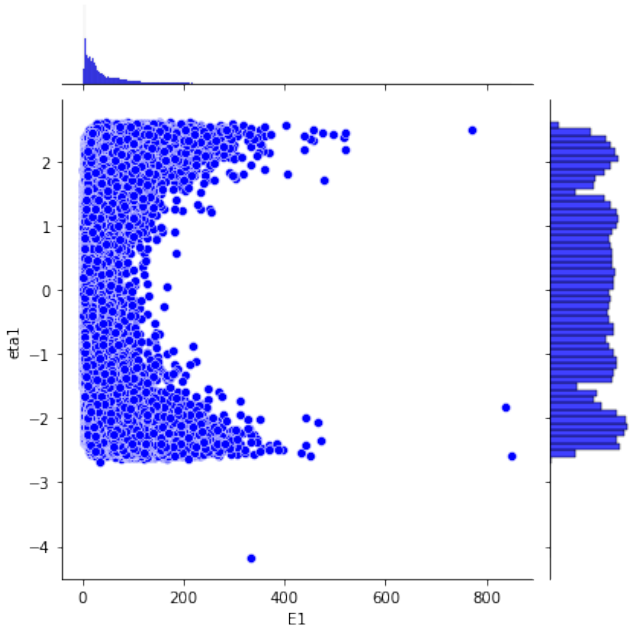

Figure 9. Jointplot of E1 and ETA1 values

Figure 9. displays a graph of a Jointplot for E1 and ETA1 variables. Similar plots have been created to represent the relationship between E1 and the p-vector, E1 and eta1, and E1 and phi1.

```
// Display Distplot

# Distplot representing E1 values

figure = plt.figure(figsize=(20,8))

sns.distplot(data[data['E1'] >
    data['E1'].mean()]["M"],
    color='blue',label='LESS')

sns.distplot(data[data['E1'] <
    data['E1'].mean()]["M"],
    color='yellow',label='UPPER')
plt.legend()
```

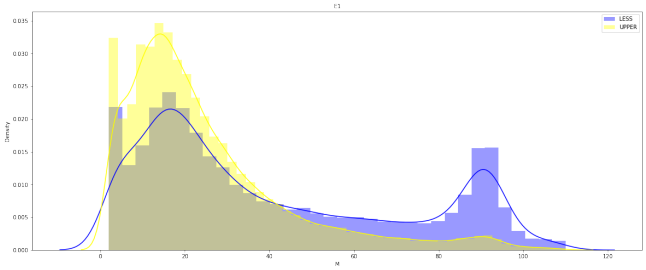

Figure 10. Distplot of E1 values

Figure 10 creates a visualization of a Distplot for the range of E1 values. Similar graphs created show the range for variables such as eta1, phi1, and the p-vector.

```
// Display 3D Plot

# 3D plot representing px1, py1, pz1 values
fig = plt.figure(figsize=(20,8))
ax = Axes3D(fig)
ax.scatter(data["px1"], data["py1"],
    data["pz1"], c="green", s=20, alpha=0.2)
plt.show()
```

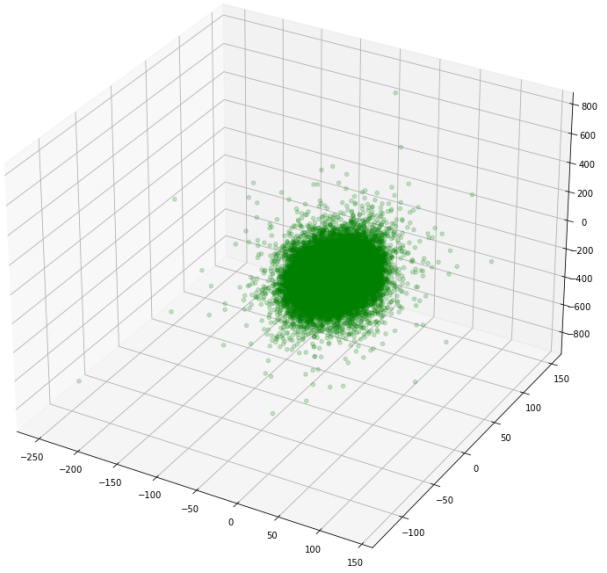

Figure 11. 3D plot of P1 vector

Figure 11. shows a 3D model of the the P-vector (px1, py1, and pz1). A three dimensional representation helps examine the relationship among 3 variables along the x, y, and z planes.

4.1.2 Machine Learning Model

```
// Create test and training sets for
    Prediction Model

x = data.drop("M",axis=1)
y = data["M"]

xTrain,xTest,yTrain,yTest =
    train_test_split(x,y,test_size=0.2,
random_state=42)

# Create Parameters for CatBoost Machine
    Learning Model
params = {

    "depth": [2, 3, 4, 5, 6],
    "learning_rate": [0.1, 0.01, 0.5]
}
```

```
# Display Depth and Learning Rates using
    GridSearchCV module
GSC = GridSearchCV(catbr, params, cv=10,
    verbose=False).fit(xTrain, yTrain)
print(GSC.best_params_)
```

Table 1. GridSearchCV

Results	
Depth	Learning Rate
6	0.5

```
# Create CatBoost ML Model with input
    parameters of previously discovered
    depth and learning rate and fit test and
    training sets

CBR =
    CatBoostRegressor(depth=6,learning_rate
=0.5, verbose=False).fit(xTrain,yTrain)

# View cross validation score
XVAL =
    cross_val_score(catbrtuned,xTest,yTest,
cv=10, scoring="r2").mean()
print(XVAL)

# View Mean Squared Error Score
errorscore =
    -cross_val_score(catbrtuned,xTest,yTest,
cv=10,
    scoring="neg_mean_squared_error").mean()
print(np.sqrt(errorscore))
```

Table 2. CatBoost Model

Results	
R2	MSE
0.9877533212829965	2.7941971481306256

4.2 Justification of Techniques

In this research project, we have utilized a number of analysis techniques to reach our desired outcome. We have reached an excellent Prediction Model R2 score through the use of exploratory analysis and our chosen CatBoost Machine learning model. We have supplied an example of each exploratory analysis tool and provided an example of each without displaying the full results for brevity.

Through the course of exploratory analysis we began with an overview of the data with the use of Pearson Correlation and Covariance Heatmaps. The Heatmaps demonstrate how two variables are linearly correlated, and the possibility of a direct relationship. We then viewed the pertinent variables through a histogram visualization. From the histogram we are able to observe interesting similarities between variables. Further on in our exploration, we visualize Scatterplots between meaningful relationships such as E1 and M, eta and M, phi1 and M, and the p-vector and M. Next we create visualizations of Histoplots where we examine the range of variables such as E2, eta2, phi2 and the p-vector. Next we have Jointplot visualizations comparing relationships between variables such as E1 and the p-vector, and E1 and phi1. Our next visualizations include Distplots representing the mean of a selected variable as well as the Lesser and Upper bounds which we use as another form of range examination for variables such as E1, eta1 and the p-vector. Finally, we examine relationships through the use of 3D plots to examine data along the x, y, and z axis'. 3D plots allow us to explore relationships amongst three variables as opposed to the two variables in previous visualizations.

After completing the exploratory analysis we can then use the insights discovered to create the CatBoost Machine Learning Model. This prediction model was selected for data analysis based on the gradient boosting and decision tree algorithms on which the model is built. CatBoost excels in analysis of numerical, categorical, and text data. We have prepared the CatBoost Prediction model by first determining the proper depth and and learning rate for the model. CatBoost was also selected for its optimized algorithm which is able to accurately calculate datasets with relatively less records. Overall, we achieved our desired outcome with the CatBoost Prediction Model returning an excellent R2 score.

4.3 Advantages and Disadvantages

There are, inevitably, advantages and disadvantages with every data analysis technique. Exploratory analysis techniques excel in providing a broad overview of the statistical information contained within the data. From these observations we can more wisely choose the proper regression, classification, or machine learning technique to achieve our desired outcome. The CatBoost model was selected for this data experiment in part for its excellence in analyzing tabular heterogenous data. CatBoost is the open-source successor to te MatrixNet algorithm and can be universally applied to a wide range of problems and areas. An advantage of CatBoost is the model's ability to examine categorical features and the algorithm's speed of calculation.

A disadvantage of exploratory analysis is that visualizing the data leads to a dependence on the observer to identify and choose a data analysis technique, of which this method may be biased by the observer. A pair of Data Scientists may view the visualizations from exploratory analysis and choose radically different approaches to reaching their final outcomes. Exploratory Analysis relies heavily on the analyst's knowledge and experience with data to interpret and manipulate the data. On that note, there are a plethora of machine learning

models available for use and personal bias towards CatBoost may have led to incorrectly choosing the most effective prediction model for analysis. The biggest disadvantage of CatBoost is the prediction model's black box nature. Although we are able to review the final results, the calculation and tabulation information is not readily available and for a more thorough accounting of the calculations we are left to speculation of how the data was exactly processed.

5. Summary and Implications

5.1 Summary

The subject of this research project was stated as, can we predict the mass of an electron by analyzing dielectron events in the mass range 2-110 GeV and will our machine learning algorithm prove to be statistically significant in predicting mass in subatomic particle collisions? In this data analysis, we reviewed the interaction of electron collisions through examination of variables including energy of the electron, the momentum components vector represented as px, py, pz, tranverse momentum, pseudorapidity, phi angle, charge, and our target variable mass. Through various exploratory analysis techniques we were able to confirm the relationship and accuracy of the provided CERN dataset. We exmained our data with visualizations and graphs including Heatmaps, Histograms, Scatterplots, Histplots, Jointplots, Distplots, and 3D models.

Through the application of the CatBoost open-source Machine Learning Prediction Model we can confirm that our research goal was successfully achieved and the output generated aligns with our research goal. The CatBoost Prediction Model generated an amazing R2 score of 0.98844793 showing that, even with a relatively low amount of 100,000 records, we were able to successfully train and test our dataset and return meaningful results from a model that does not suffer from either under or overfitting. The results align with our research goal and proves that the analysis techniques and the CatBoost Prediction Model specifically are excellent tools for analyzing and creating statistically significant predictions in the study of subatomic particle collisions.

5.2 Limitations

The greatest limitation of the research project is the limited availability of records in the Electron Dataset from the CERN organization for analysis. The 100,000 records provided allow for meaningful and insightful results which align with our research goal, however, more data would allow for increased accuracy in the final calculated outcome. It is for this particular reason that the CatBoost Machine Learning Prediction model was selected as the algorithm allows for accurate and efficient calculations on datasets with limited record availability. CatBoost however was able to overcome this limitation and produce a model that is neither underfit or over fit.

Another limitation is the resources available for computation. CatBoost is an extremely efficient algorithm able to calculate complex mathematics quickly. However, even with a sophisticated machine learning model the time to compute the necessary calculations resulted in long and arduous waiting periods. There are many options available for machine learning, but they require the use of complex and expensive equipment. The creation of more sophisticated models and larger datasets will require access to higher performance systems.

5.3 Recommendations

This research project explored the statistical significance of machine learning prediction models in the field of subatomic particle physics and the meaningful results produced show the viability of Data Science and more specifically the use of machine learning for scientific inquiry. It is highly recommended that the data analysis techniques and the CatBoost machine learning algorithm be applied to similar fields in particle physics including the study of photon collisions and aspects of investigating the Higgs Boson Field. Further study and use of these tools and techniques could provide significant insight into both the scientific branches of Physics and Data Science. CERN researchers and the scientific community could benefit from the insights gained in this research report and like the open-source CERN dataset are free and welcome to use in their own projects and analysis.

5.4 Two Directions for Future Study

In the final section of this research report we will examine the viability of similar machine learning models for possible future investigation. We will examine regression techniques and grade their results on the accuracy of the recorded R2 result.

```
# Instantiate Regression Models
baggr =
    BaggingRegressor(random_state=42,yTrain)
cartr =
    DecisionTreeRegressor(random_state=42)
.fit(xTrain,yTrain)
catbr = CatBoostRegressor(verbose=False)
.fit(xTrain,yTrain)
elasticnet = ElasticNet().fit(xTrain,yTrain)
gbmr =
    GradientBoostingRegressor(verbose=False)
.fit(xTrain,yTrain)
knnr =
    KNeighborsRegressor().fit(xTrain,yTrain)
lasso = Lasso().fit(xTrain,yTrain)
lgbmr = LGBMRegressor().fit(xTrain,yTrain)
pls = PLSRegression().fit(xTrain,yTrain)
rfr =
    RandomForestRegressor(random_state=42,yTrain)
ridge = Ridge().fit(xTrain,yTrain)
xgbr = XGBRegressor().fit(xTrain

# Calculate and Display R2 results of each
    regression technique
for model in models:
```

```
name = model.__class__.__name__
R2CV = cross_val_score(model,xTest,yTest,
cv=10,scoring="r2").mean()
error =
    -cross_val_score(model,xTest,yTest,
cv=10,scoring="neg_mean_squared_error")
.mean()
```

Table 3. Regression Comparison

Results	
Test	R2
BaggingRegressor	0.8622002567153235
DecisionTreeRegressor	0.6071672063829866
CatBoostRegressor	0.9895776897205797
ElasticNet	0.4054388007788111
GradientBoostingRegressor	0.7629687918727849
KNeighborsRegressor	0.886945474307922
Lasso	0.40552788633427134
LGBMRegressor	0.9538695831740363
PLSRegression	0.4003063950414911
RandomForestRegressor	0.8973652785217026
Ridge	0.4052704354267057
XGBRegressor	0.9614366434549941

Table 3. regression results show that CatBoost is the best model for this research project. With a R2 result of 0.9857768 it has received the best results from the regression analysis in comparison to similar models.

One direction for future study of this subject and dataset, we would recommend using similar and well-achieving regression algorithms such as CNN and OLS for a more thorough analysis and significant comparison of Prediction Models that best align with our research goal.

A second direction for future study would be to incorporate a more mathematical approach to the research goal such as Patsy for statistical models and SymPy for displaying complex mathematical formulas.

Finally, another direction for future study would be to apply the insights and knowledge gained in this research project to observations from lateral CERN research experiments including photon particle physics or to the study of similar topics in the fields of chemistry and biology.

Continued study in Machine Learning Prediction Models will serve well to enhance the field of Data Science and concurrently deliver increased insights to the scientific community which can lead to enhancements in countless various fields of knowledge such as economics, bio-sciences, medicine, and improving quality of life.

6. References

McCauley, T. (1970, January 1). Events with two electrons from 2010. CERN Open Data Portal. Retrieved May 10, 2022, from https://opendata.cern.ch/record/304

CatBoost – a new game of machine learning. Affine. (n.d.). Retrieved May 10, 2022, from https://affine.ai/catboost-a-new-game-of-machine-learning/

How CatBoost algorithm works in Machine Learning. Dataaspirant. (2021, January 4). Retrieved May 10, 2022, from https://dataaspirant.com/catboost-algorithm/

Thiesen, S. (2021, February 19). CatBoost regression in 6 minutes. Medium. Retrieved May 10, 2022, from https://towardsdatascience.com/catboost-regression-in-6-minutes-3487f3e5b329

Exploratory Data Analysis in python course. DataCamp. (n.d.). Retrieved May 10, 2022, from https://www.datacamp.com/courses/exploratory-data-analysis-in-python

Supervised learning with scikit-learn course. DataCamp. (n.d.). Retrieved May 10, 2022, from https://www.datacamp.com/courses/supervised-learning-with-scikit-learn

Real Python. (2021, March 19). Linear regression in python. Real Python. Retrieved May 10, 2022, from https://realpython.com/linear-regression-in-python/

Katari, K. (2020, October 9). Simple linear regression model using Python: Machine Learning. Medium. Retrieved May 10, 2022, from https://towardsdatascience.com/ simple-linear-regression-model-using-python-machine-learning

Acknowledgments

I would like to first and foremost thank the CERN Open Data organization for providing the data to make this research project possible.

I am equally grateful to the faculty and staff of Western Governors University and their tireless efforts to deliver the best education possible.

I would also like to thank my Mentor Emil Stoica for his unwavering support and help throughout the MSDA Journey.

I am eternally thankful to my Fiance for her understanding and sacrifices as I continue my education.

And finally, I am most thankful to my Mother for her unconditional love and support through each endeavour I have ever embarked upon.