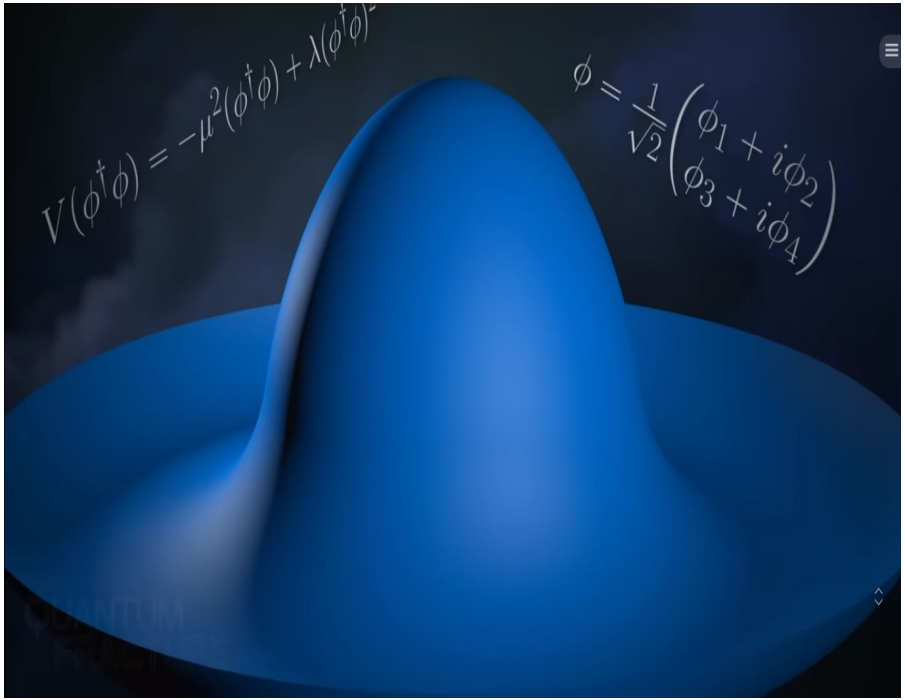
Bosón de Higgs

$$H \rightarrow \tau^{+}\tau^{-}$$

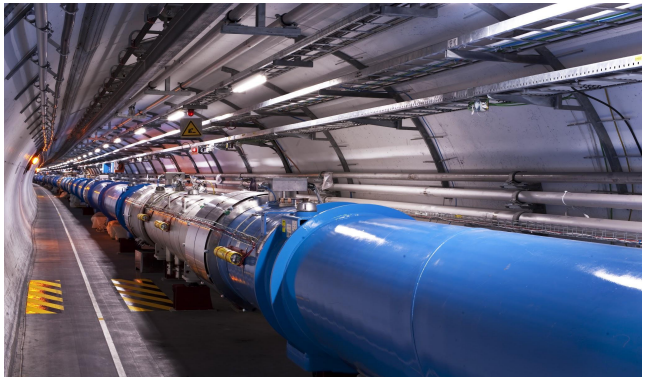
Maximiliano de Jesús Galindo Hernández

Alejandro Mendoza Puig

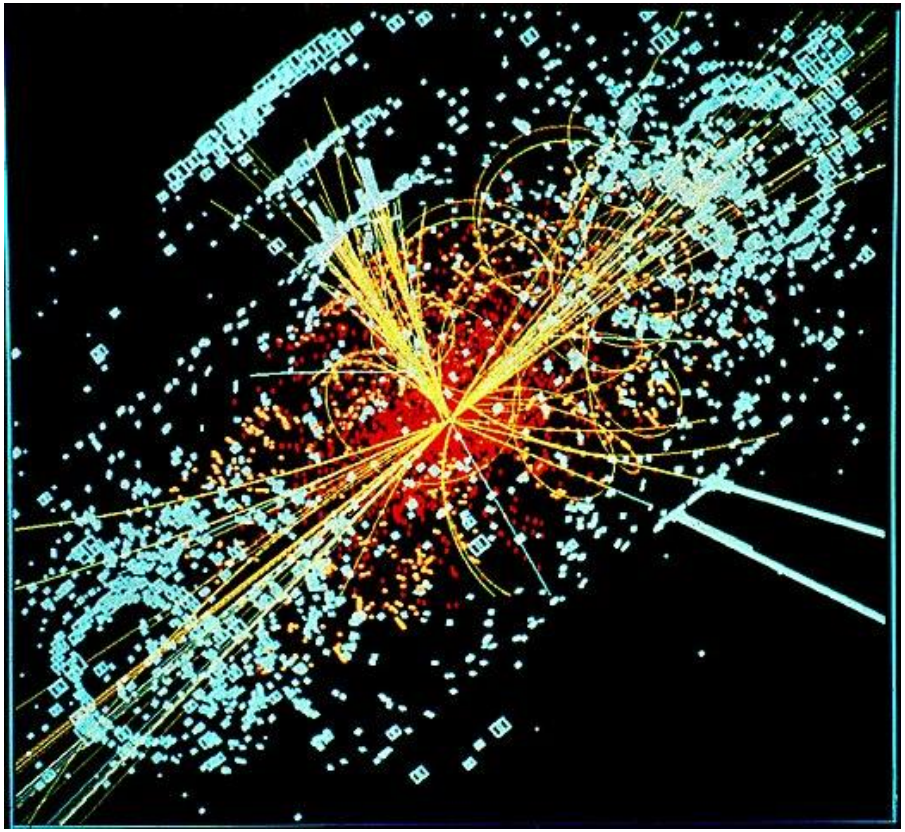


- El bosón de Higgs es una partícula fundamental propuesta que desempeña un papel crucial en la explicación de cómo las demás partículas fundamentales adquieren masa.

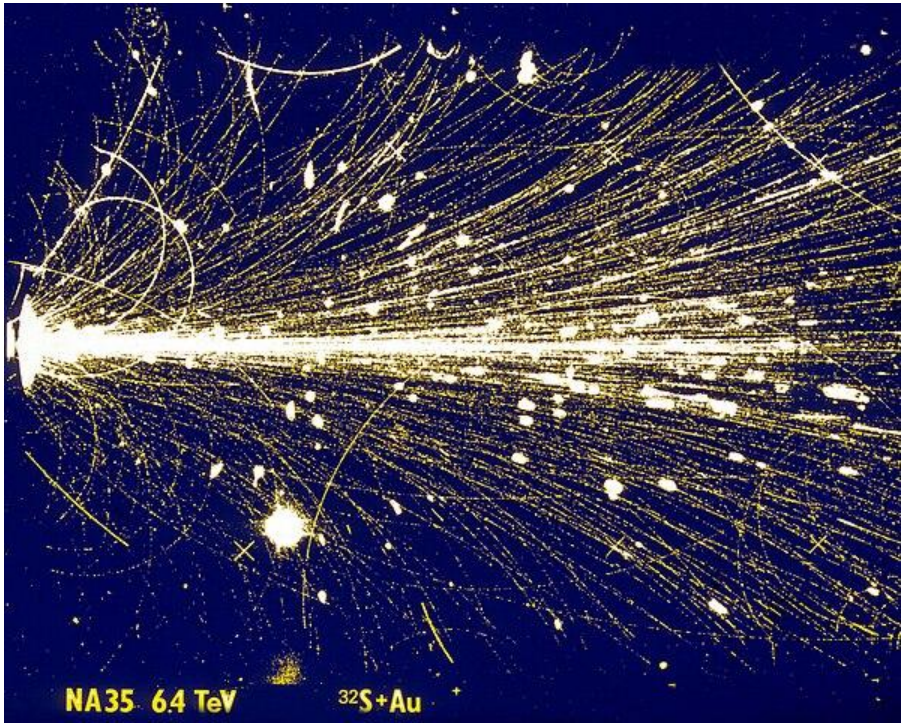
Contexto



- En la búsqueda de nuevas partículas, los físicos emplean una herramienta conocida como acelerador de partículas.
- Estos aceleradores de partículas provocan la colisión entre partículas, generando así otras partículas, como es el caso del bosón de Higgs.



- Una traza hipotética del bosón de Higgs en una colisión simulada protón-protón.

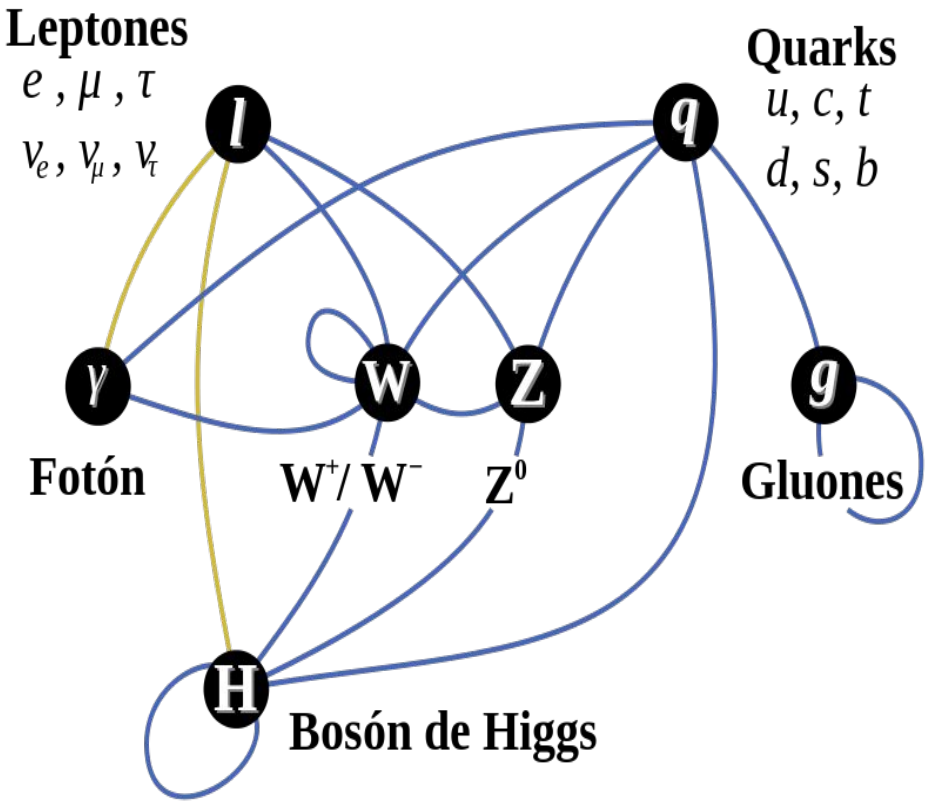


Base de Datos

- Los datos fueron generados mediante simulaciones de Monte Carlo. Con un total de 11M de filas y 28 variables.
- Las primeras 21 características son propiedades cinemáticas, mientras que las últimas siete son funciones de las primeras 21.

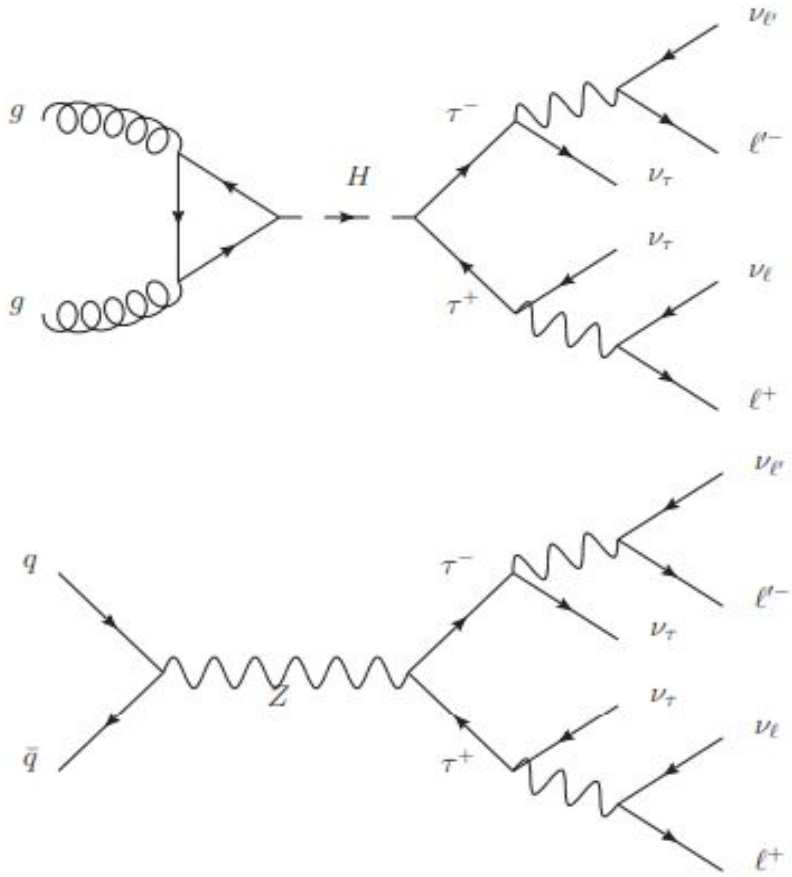
Variables a Analizar

- Los momentos tridimensionales, p, de los leptones cargados;
- Pérdidas de energía en el estado final transversal a la dirección del haz.
- El número y momentos de los 'jets' de partículas debido a la radiación de gluones o quarks.



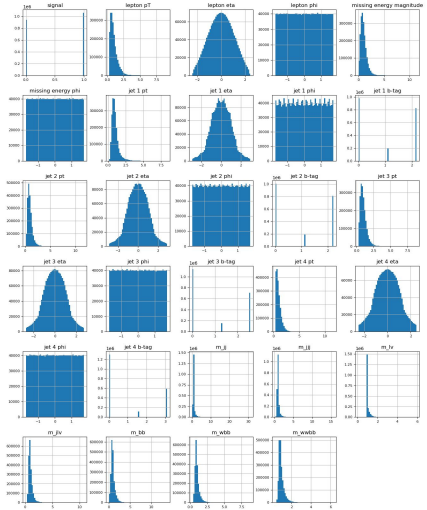
Definición del Problema (Motivación)

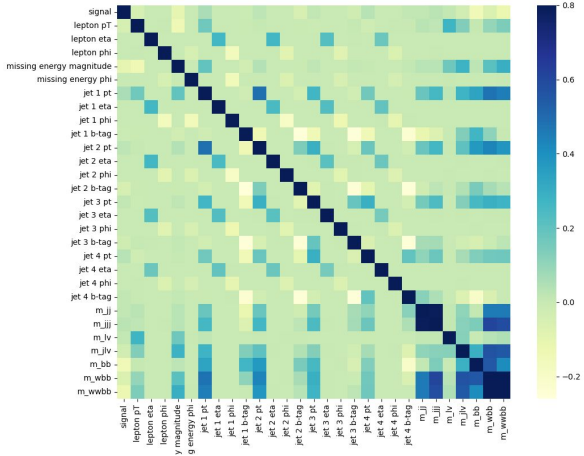
- La medición de un segundo de las colisiones genera 40TB de datos.
- Sets de datos de millones de registros son públicos en competencias de ML
- Estos datos son procesados principalmente con Python en competencias de Kaggle



Objetivos

- Medir y comparar el desempeño de Python con el de Java haciendo utilización de hilos.
- Funciones principalmente usadas en la fase de exploración de datos como filtrados de columnas, renglones y lectura de archivos > 1GB
- Identificar diferencias en las distribuciones de datos cuando se detectó un bosón de Higgs y cuando es otra partícula.





Avances

Funciones en python para explorar los datos posterior a filtrarse con Java.

- Distribuciones
- Correlaciones

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

Abrir, particionar y poner nombre a columnas del set de datos

```
data = pd.read_csv('HIGGS.csv.gz', nrows=2000000, compression='gzip',
header=None, sep=',')

new_column_names = ["signal",
                    "lepton pT",
                    "lepton eta",
                    "lepton phi",
                    "missing energy magnitude",
                    "missing energy phi",
                    "jet 1 pt",
                    "jet 1 eta",
                    "jet 1 phi",
                    "jet 1 b-tag",
                    "jet 2 pt",
                    "jet 2 eta",
                    "jet 2 phi",
                    "jet 2 b-tag",
                    "jet 3 pt",
                    "jet 3 eta",
                    "jet 3 phi",
                    "jet 3 b-tag",
                    "jet 4 pt",
                    "jet 4 eta",
                    "jet 4 phi",
                    "jet 4 b-tag",
                    "m_jj",
                    "m_jjj",
                    "m_lv",
                    "m_jlv",
                    "m_bb",
                    "m_wbb",
                    "m_wwbb"]
data.columns = new_column_names
print(data.columns)

# Guardar los 2 millones de datos en un nuevo archivo csv
data.to_csv('HIGGS_2M.csv', index=True)
```

Revisar si los datos están completos

```
# Revisamos que no haya datos nulos en ninguna de las columnas
vacios = data.isnull().sum()

for i in range(len(vacios)):
    print("{}: {}".format(new_column_names[i], vacios[i]))
```

Distribución de los datos

```
data.shape

data.head(10)

caracteristicas ={}
for i in range(len(new_column_names)):
    caracteristicas[new_column_names[i]] =
data[new_column_names[i]].describe()

for i in caracteristicas:
    print(caracteristicas[i])

# Graficar las distribuciones de datos
for i in range(len(new_column_names)):
    plt.figure(figsize=(9, 8))
    sns.distplot(data[new_column_names[i]], color='b', bins=100)
    plt.savefig('Exploracion/{}.png'.format(new_column_names[i]))

data.hist(figsize=(16, 20), bins=100, xlabelsize=10, ylabelsize=10)
```

Correlaciones

```
correlaciones = data.corr()
f, ax = plt.subplots(figsize=(10, 10))
sns.heatmap(correlaciones)
plt.savefig('Exploracion/Correlacion.png')
```