SOEN 6611 – Software Measurement

PROJECT STEP 5

Source *- SEI implementing Goal-Driven Measurement course material (adapted).*

Submitted to - Prof. Dr. Olga Ormandjieva

Team – 7

Rutwikkumar Sunilkumar Patel – (40160646)
Charit Pareshbhai Patel – (40160658)
Deep Pareshkumar Patel – (40185585)
Bhoomi Shah – (40169655)

Table of Content		
Index	**Contents**	**Page**

1	Dataset Description	3
2	Base measures data collection procedure	12
3	Attach a detailed view of the collected data values	19
4	For each of the V's (Validity, Vincularity and Veracity) indicators	20
5	Conclusion	28
6	Project Code Link	28

1. Dataset Description

Dataset name: IBM HR analytics Employee Attrition & Performance

Source: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Description: The dataset is about factors that accelerate employee attrition. The dataset contains analytical data on employees' Education, Environment Satisfaction, Job Involvement and Satisfaction, Performance Rating, Relationship Satisfaction, and Work-Life Balance. We aim to apply the measures of each of 3 V's to this dataset and analyze the results. Also, perform the data extraction, data preprocessing, and data processing techniques and check whether the dataset can be used for machine learning models.

Size of Dataset: 227.98 kB

Structure: 35 Columns and 1468 rows

Number of unique records: 1468 records

Columns descriptions:

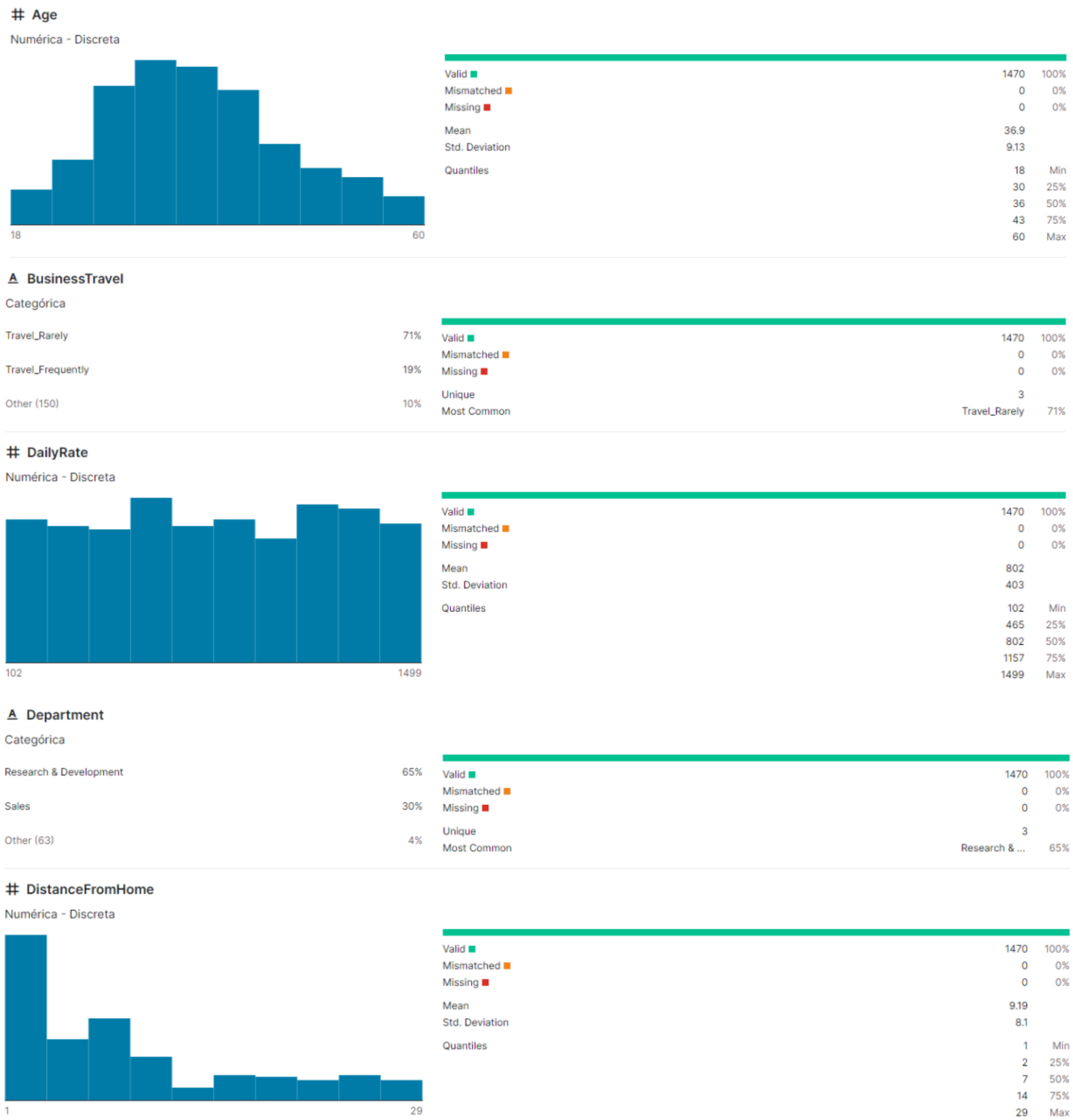


Figure 1.1: Statistical information of the dataset (*Source: screenshot from* *https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset*)

Education

Categórica

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	2.91	
Std. Deviation	1.02	
Quantiles	1	Min
	2	25%
	3	50%
	4	75%
	5	Max

EducationField

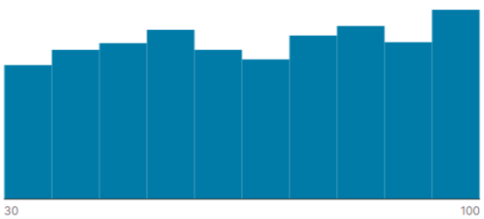
Categórica

Life Sciences	41%
Medical	32%
Other (400)	27%

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Unique	6	
Most Common	Life Sciences	41%

HourlyRate

Numérica - Discreta



Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	65.9	
Std. Deviation	20.3	
Quantiles	30	Min
	48	25%
	66	50%
	84	75%
	100	Max

JobInvolvement

Categórica

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	2.73	
Std. Deviation	0.71	
Quantiles	1	Min
	2	25%
	3	50%
	3	75%
	4	Max

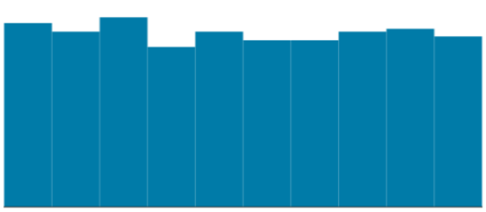
EmployeeCount

Numérica - Discreta

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	1	
Std. Deviation	0	
Quantiles	1	Min
	1	25%
	1	50%
	1	75%
	1	Max

EmployeeNumber

Numérica - Discreta



Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	1.02k	
Std. Deviation	602	
Quantiles	1	Min
	491	25%
	1022	50%
	1556	75%
	2068	Max

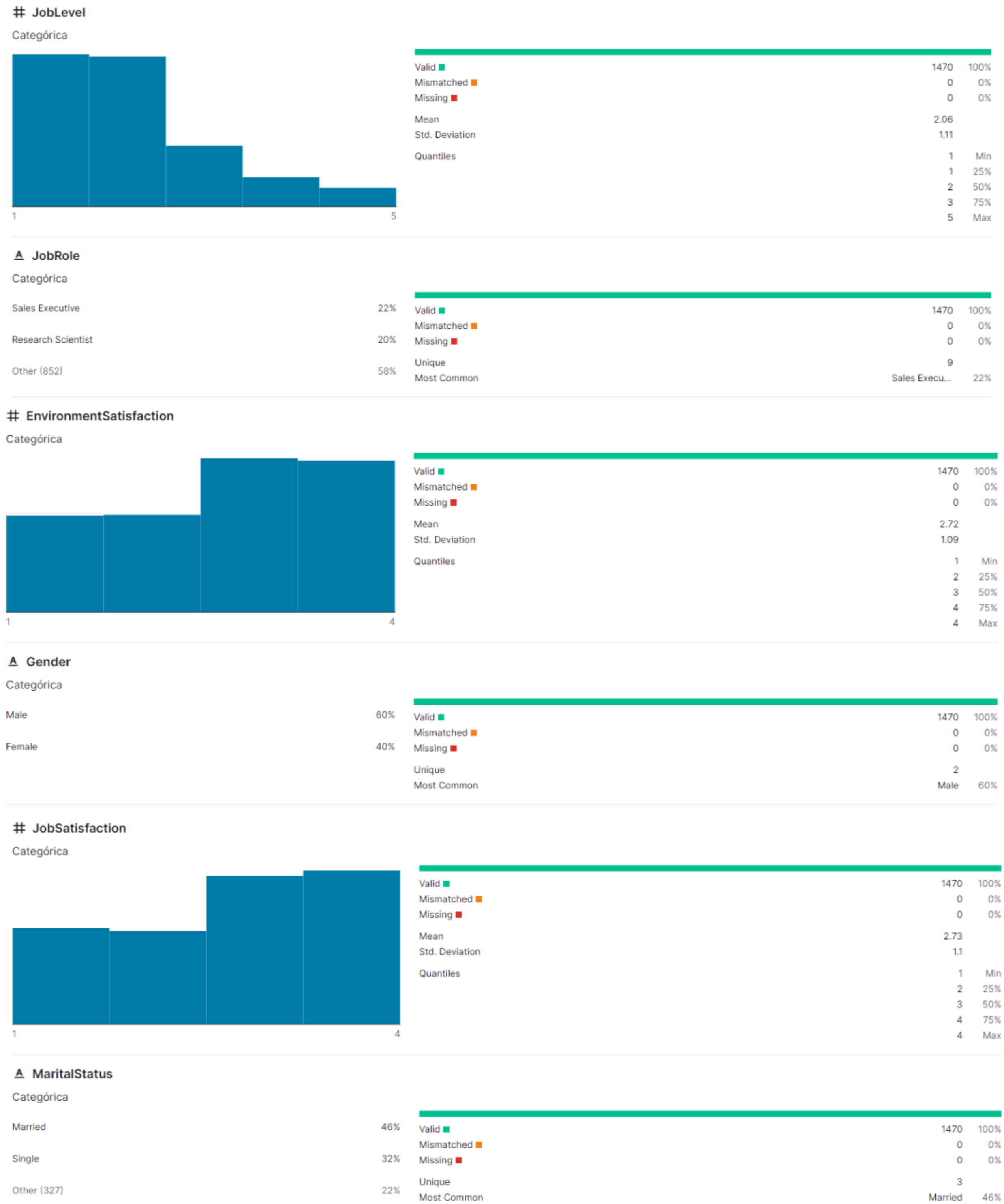


Figure 1.2: Statistical information of the dataset (*Source: screenshot from https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset*)

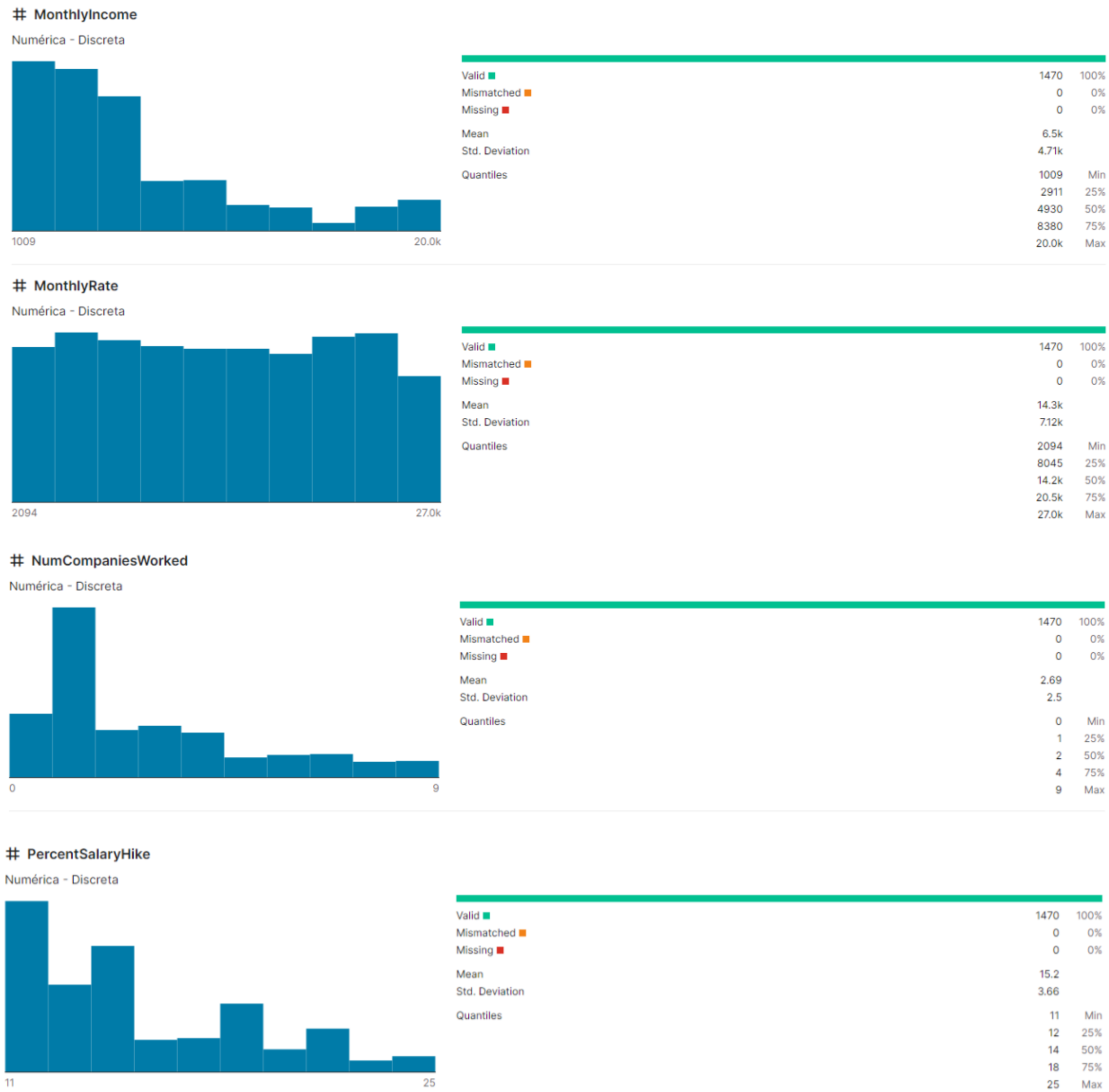


Figure 1.3: Statistical information of the dataset (*Source: screenshot from https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset*)

PerformanceRating

Categórica

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	3.15	
Std. Deviation	0.36	
Quantiles	3	Min
	3	25%
	3	50%
	3	75%
	4	Max

RelationshipSatisfaction

Categórica

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	2.71	
Std. Deviation	1.08	
Quantiles	1	Min
	2	25%
	3	50%
	4	75%
	4	Max

StandardHours

Numérica - Discreta

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	80	
Std. Deviation	0	
Quantiles	80	Min
	80	25%
	80	50%
	80	75%
	80	Max

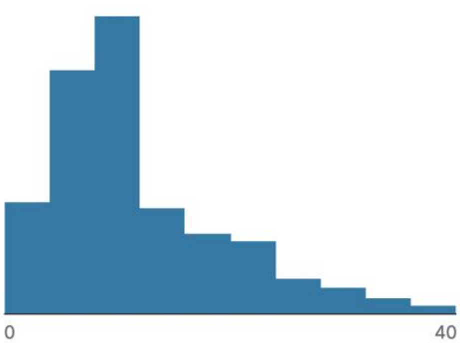
StockOptionLevel

Categórica

Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	0.79	
Std. Deviation	0.85	
Quantiles	0	Min
	0	25%
	1	50%
	1	75%
	3	Max

TotalWorkingYears

Numérica - Discreta



Valid	1470	100%
Mismatched	0	0%
Missing	0	0%
Mean	11.3	
Std. Deviation	7.78	
Quantiles	0	Min
	6	25%
	10	50%
	15	75%
	40	Max

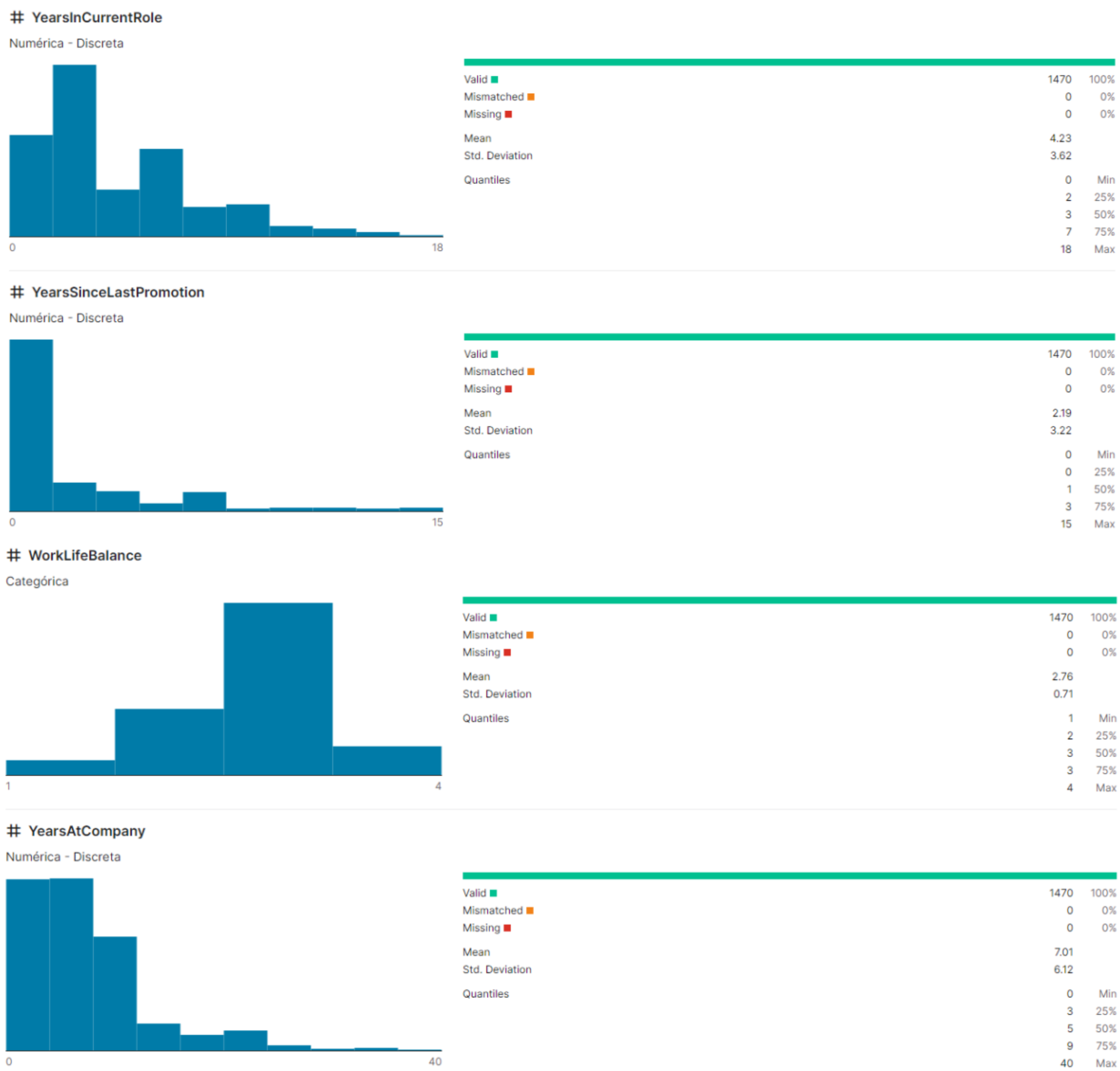
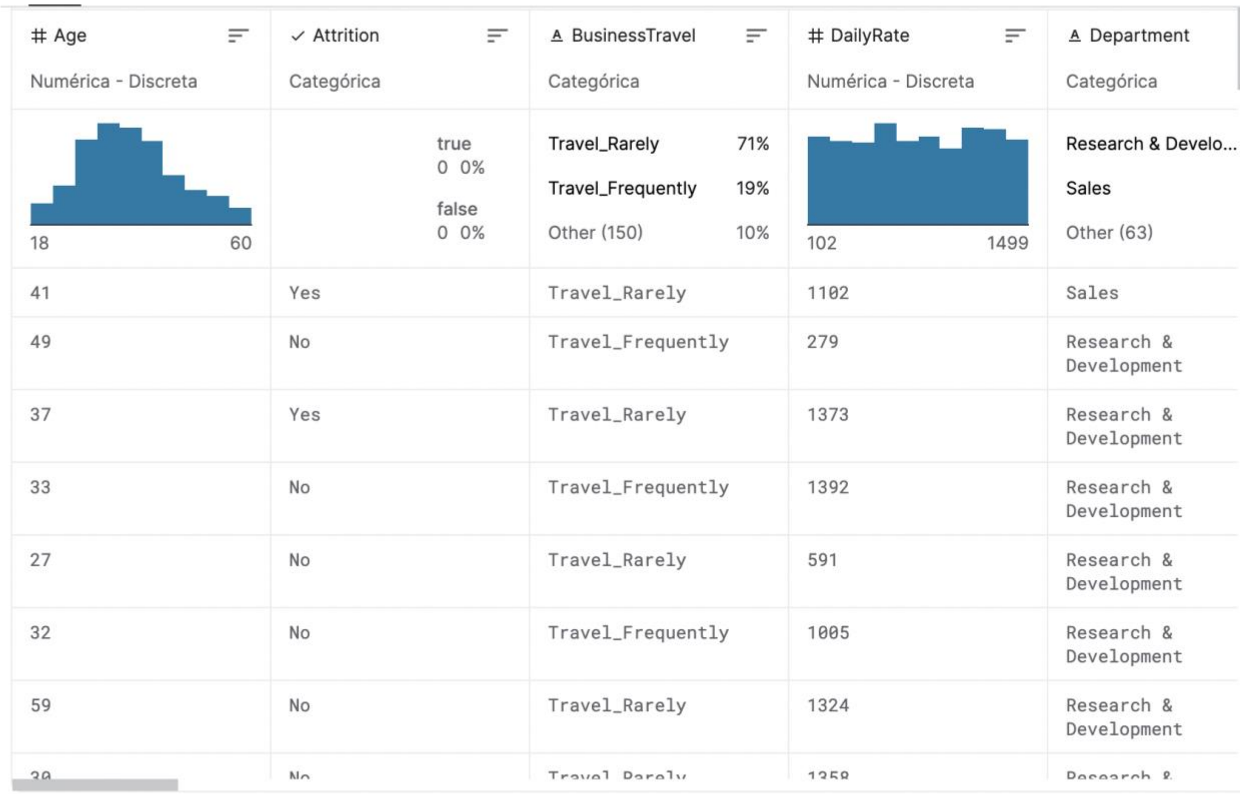


Figure 1.4: Statistical information of the dataset (*Source: screenshot from https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset*)

Details:

The sample of the data available in .CSV file of the dataset is shown below



# Age	✓ Attrition	BusinessTravel	# DailyRate	Department
Numérica - Discreta	Categórica	Categórica	Numérica - Discreta	Categórica
18 – 60	true 0 0% false 0 0%	Travel_Rarely 71% Travel_Frequently 19% Other (150) 10%	102 – 1499	Research & Develo... Sales Other (63)
41	Yes	Travel_Rarely	1102	Sales
49	No	Travel_Frequently	279	Research & Development
37	Yes	Travel_Rarely	1373	Research & Development
33	No	Travel_Frequently	1392	Research & Development
27	No	Travel_Rarely	591	Research & Development
32	No	Travel_Frequently	1005	Research & Development
59	No	Travel_Rarely	1324	Research & Development

Figure 1.5: Summary of the dataset (*Source: screenshot from https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset*)

Activity Overview:

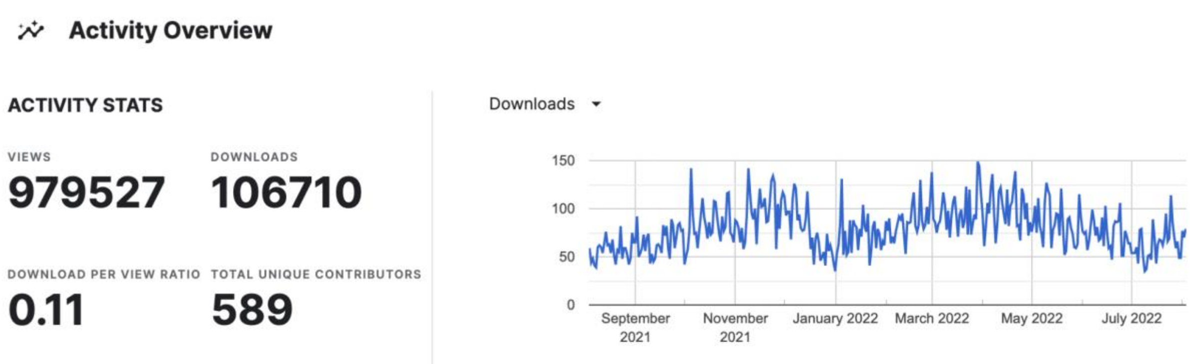


Figure 1.6: Activity overview of the dataset (*Source: screenshot from https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset*)

Data loading and splitting in timeframes

Preparing the programming scripts or analytical tools for collecting the base measures and calculating the derived measures.

Firstly, implemented a python script for data reading using libraries.

```
[ ] from pandas.io.parsers.readers import read_csv
    dir_path = "/content/drive/My Drive/SOEN6611_ProjectDataset/IBM.csv"
    dir_path_1 = "/content/drive/My Drive/SOEN6611_ProjectDataset/BigBasket.csv"
    df = read_csv(dir_path)

    df.head()
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	RelationshipSatisfac
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	

Divided dataset into 3 timeframes T1, T2, and T3.

```
[ ] #Getting rows and columns
    df.shape

    #Dividing dataset into three dataframes t1, t2 and t3 for quality analysis.
    df_T1 = df.iloc[:400,:]
    df_T2 = df.iloc[401:900,:]
    df_T3 = df.iloc[901:,:]

    print("The Size of all three timeframes are :  {}, {}, and {}".format(df_T1.shape, df_T2.shape, df_T3.shape))

    The Size of all three timeframes are :  (400, 35), (499, 35), and (569, 35)
```

We have used these data frames for the calculations of base and derived measures for different timeframes in upcoming steps.

2. Base measures data collection procedure

2.1 Collection Procedure to measure the length of big data (LBD)

We first divided our dataset into three different timeframes T1, T2, and T3. Then we calculated the length of big data (LBD) in each timeframe using pandas and google colab. Our team worked for half an hour to collect this measure. This base measure will be used to calculate the **veracity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT04** of step 4. The python script for Lbd is shown below.

```
#Getting the length of big data.
lbd_T1 = df_T1.shape[0]
lbd_T2 = df_T2.shape[0] + lbd_T1
lbd_T3 = df_T3.shape[0] + lbd_T2

print("The length of big data at each time frame would be: {}, {}, and {}".format(lbd_T1, lbd_T2, lbd_T3))
```

```
The length of big data at each time frame would be: 400, 899, and 1468
```

2.2 Collection Procedure to measure the number of datasets (Nds) of big data

After calculating LBD, we proceed to calculate the number of datasets in big data. Here, we have used only one dataset as we are provided with one dataset only. We have divided this dataset into three different timeframes T1, T2, and T3. So the number of datasets at each timeframe will be exactly one. This base measure will be used to calculate the **validity** and **Vincularity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT02** of step 4.

We have calculated the value of Nds manually.

```
nds_T1 = 1
nds_T2 = 1
nds_T3 = 1
```

2.3 Collection Procedure to measure the number of distinct data elements (Ndde) in big data

After calculating Nds, our team proceed to calculate the number of distinct data elements in big data. We calculated Ndde for each of the different timeframes T1, T2, and T3 using pandas and google colab. Our team worked for half an hour to collect this base measure. This base measure will be used to calculate the **veracity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT03** of step 4. The python script is shown below.

```
numberOfDistinctDataElements_T1 = 0
numberOfDistinctDataElements_T2 = 0
numberOfDistinctDataElements_T3 = 0

for col in df_T1.columns:
  numberOfDistinctDataElements_T1 += len(df_T1[col].unique())
print("Ndde at Time frame T1: " , numberOfDistinctDataElements_T1)

for col in df_T2.columns:
  numberOfDistinctDataElements_T2 += len(pd.concat([df_T1, df_T2])[col].unique())
print("Ndde at Time frame T2: " , numberOfDistinctDataElements_T2)

for col in df_T3.columns:
  numberOfDistinctDataElements_T3 += len(pd.concat([df_T1, df_T2, df_T3])[col].unique())
print("Ndde at Time frame T3: " , numberOfDistinctDataElements_T3)
```

```
Ndde at Time frame T1:  1896
Ndde at Time frame T2:  3655
Ndde at Time frame T3:  5500
```

2.4 Collection Procedure to measure the total number of records with no null values (Rec_no_null) in big data

After calculating Ndde, we proceed to calculate the total number of records with no null values, meaning, we calculated the total number of complete records in big data at different timeframes T1, T2, and T3. To calculate this measure, we traversed through each element of each data frame and checked whether there exists a null value for any element of any record. If we find such an element for any record, then we increase the counter storing the number of records with a null value. We subtracted this counter from the total number of records (Lbd) to find the number of complete records for each timeframe. This base measure will be used to calculate the **veracity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT07** of step 4. The python script is shown below.

```python
def get_rec_no_null(df, offset):
  count_null = 0
  count_no_null = 0

  # print(df.isnull())

  for i, j in df.iterrows():
    for col in range(len(j)):
      if(df.iat[i-offset,col] == 'NaN'):
        count_null = count_null + 1
        break;

  #print("Records with null values: ", count_null)
  count_no_null = df.shape[0] - count_null
  #print("Records with no null values: ", count_no_null)
  return count_no_null

print("Rec_no_null at T1: ", get_rec_no_null(df_T1, 0))
print("Rec_no_null at T2: ", get_rec_no_null(pd.concat([df_T1, df_T2]), 401))
print("Rec_no_null at T3: ", get_rec_no_null(pd.concat([df_T1, df_T2, df_T3]), 901))
```

```
Rec_no_null at T1:  400
Rec_no_null at T2:  899
Rec_no_null at T3:  1468
```

2.5 Collection Procedure to measure the total number of records within an acceptable age range (Rec_acc_age) in big data

After calculating Rec_no_null, we proceed to calculate the total number of records within an acceptable age range at different timeframes T1, T2, and T3. To calculate this measure, we considered the **YearAtCompany** column to filter out the records. For each timeframe, we draw the box plot from the values available in the **YearAtCompany** column. From the box plot, we found this column's acceptable range of values. Then we traversed through the entire data frame to find out the number of records falling into an acceptable range. This base measure will be used to calculate the **veracity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT07** of step 4. The python script is shown below.

```python
def calculate_acc_records(df):
  df = df.sort_values('YearsAtCompany')
  # print(df_T1)
  lq = df.iloc[round(df.shape[0] * 1/4), df.columns.get_loc('YearsAtCompany')]
  uq = df.iloc[round(df.shape[0] * 3/4), df.columns.get_loc('YearsAtCompany')]
  # print(lq)
  # print(uq)
  # Acceptable range is [lq, uq]]
  # So the values outside of this range are considered as outliers and we will consider them as outdated records

  count_acc_records = 0 # Count of acceptable records
  for column in df['YearsAtCompany']:
    if column >= lq and column <= uq:
      count_acc_records = count_acc_records + 1
  return count_acc_records

print("Rec_acc_age at T1: ", calculate_acc_records(df_T1))
print("Rec_acc_age at T2: ", calculate_acc_records(df_T2))
print("Rec_acc_age at T3: ", calculate_acc_records(df_T3))
```

```
Rec_acc_age at T1:  236
Rec_acc_age at T2:  302
Rec_acc_age at T3:  355
```

2.6 Collection Procedure to measure the total number of successful requests (N_succ_req) in big data

After calculating Rec_acc_age base measure, we proceed to calculate the total number of successful requests in big data at different timeframes T1, T2, and T3. We assume that the total number of successful requests to our dataset is 75% of the size of the dataset at each timeframe. We assumed this value as there is no way to calculate this value from the dataset. This base measure will be used to calculate the **veracity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT07** of step 4.

We have calculated the value of N_succ_req manually.

2.7 Collection Procedure to measure the total number of requests (N_req) in big data

After calculating N_succ_req measure, we proceed to calculate the total number of data requests in big data at different timeframes T1, T2, and T3. We assume that the total number of data requests to our dataset is 85% of the size of the dataset at each timeframe. We assumed this value as there is no way to calculate this value from the dataset. This base measure will be used to calculate the **veracity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT07** of step 4.

We have calculated the value of N_req manually.

2.8 Collection Procedure to measure the total number of compliant records (Nrec_comp) in big data

After calculating the N_req measure, our team proceed to calculate the total number of compliant records in big data at different timeframes T1, T2, and T3. To calculate this measure, we checked whether the value of each data element is compliant with the corresponding column, meaning, the type of each data element must be matching with that of the corresponding column. For each timeframe, we traversed through each element of the data frame and checked whether the type of the data element is matching with the type of the column. If all the data elements of an entire record are compliant with their columns, then we increased the counter storing the number of compliant records. This base measure will be used to calculate the **validity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT08** of step 4. The python script is shown below.

```
def get_complaince_record(df, offset):
  count_compliant = 0
  compliant = True
  for i, j in df.iterrows():
    for col in range(len(j)):
        # print(type(df_T1.iat[i,col]))
        if(type(df.iat[i-offset,col]) != df[df.columns[col]].dtype):
          compliant = False
          break
    if compliant == False:
      count_compliant = count_compliant + 1
      compliant = True
  return count_compliant

print("comp_record at T1: ", get_complaince_record(df_T1, 0))
print("comp_record at T2: ", get_complaince_record(pd.concat([df_T1, df_T2]), 401))
print("comp_record at T3: ", get_complaince_record(pd.concat([df_T1, df_T2, df_T3]), 901))
```

```
comp_record at T1:  400
comp_record at T2:  899
comp_record at T3:  1468
```

2.9 Collection Procedure to measure the total number of credible datasets (Nds_cr) in big data

After calculating the Nrec_comp measure, our task was to calculate the total number of credible datasets in our big data at different timeframes T1, T2, and T3. We have given only one dataset in our big data. Also, the dataset is downloaded from a reliable and verified source. So, we can consider our dataset credible. As mentioned earlier, we have divided our dataset into three different timeframes. So total number of credible datasets in each timeframe will be one. This base measure will be used to calculate the **validity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT08** of step 4.

We have calculated the value of Nds_cr manually.

2.10 Collection Procedure to measure the total number of traceable records (Rec_trac) in big data

After calculating the Nds_cr measure, our task is to calculate the total number of traceable records at different timeframes T1, T2, and T3. To calculate the total number of traceable records, we

```
def get_rec_trace(df):
  rec_no_trac = 0
  for column in df['Attrition']:
    if (column != 'Yes' and column != 1) and (column != 'No' and column != 0):
      rec_no_trac = rec_no_trac + 1

  rec_trace = df.shape[0] - rec_no_trac
  return rec_trace

print("rec_trace at T1: ", get_rec_trace(df_T1))
print("rec_trace at T2: ", get_rec_trace(pd.concat([df_T1, df_T2])))
print("rec_trace at T3: ", get_rec_trace(pd.concat([df_T1, df_T2, df_T3])))
```

```
rec_trace at T1:  400
rec_trace at T2:  899
rec_trace at T3:  1468
```

2.11 Collection Procedure to measure the length of the dataset (Ldst) in big data

After calculating the Rec_trac measure, we proceed to calculate the length of each dataset at different timeframes T1, T2, and T3. Here, in our case, we have used only one dataset and we have divided it into three different timeframes. We will find the total number of records available in our data frame at each timeframe to find Ldst. This base measure will be used to calculate the **vincularity** of big data in different timeframes. This base measure collection procedure is traceable with the measurement task **MT09** of step 4. The python code is shown below.

```
print("ldst at T1: ", df_T1.shape[0])
print("ldst at T2: ", pd.concat([df_T1, df_T2]).shape[0])
print("ldst at T3: ", pd.concat([df_T1, df_T2, df_T3]).shape[0])
```

```
ldst at T1:  400
ldst at T2:  899
ldst at T3:  1468
```

3. Attach a detailed view of the collected data values

The values of collected base measures at different timeframes are shown below in the table.

Data Collected	T1	T2	T3
Length of Big data (LBD)	400	899	1468
Number of Datasets (Nds)	1	1	1
Number of distinct data elements (Ndde)	1896	3655	5500
Records with no null values (Rec_no_null)	400	899	1468
Records with acceptable age range (Rec_acc_age)	236	302	355
Number of successful requests (N_succ_req)	300	675	1101
Number of requests (N_req)	340	765	1248
Number of compliant records (Nrec_comp)	400	899	1468
Number credible datasets (Nds_cr)	1	1	1
Number of traceable records (Nrec_trac)	400	899	1468
Length of dataset (Ldst)	400	899	1468

4. For each of the V's (Validity, Vincularity, and Veracity) indicators:

4.1 Attach the values of the corresponding derived measures(s)

For calculating the values of the derived measures, we used the values of the base measures collected in section 2.

Derived Measures	Formula	Base Measures Used
Accuracy	$H_{acc}(MDS) = \log_2(Lbd) - (1 / Lbd) \times \sum_{j=\{1..k\}} p_j \log_2(p_j)$ $H_{max}(MDS) = \log_2(Lbd)$ $Accuracy\ (MDS) = \frac{H_{acc}}{H\_max}$	**LBD** from section **2.1**
Completeness	$Com_m\ (MDS) = \frac{[rec\_no\_null\ (MDS)]}{Lbd(MDS)}$	**rec_no_null** from section **2.4** **LBD** from section **2.1**
Currentness	$Currentness\ (MDS) = \frac{[rec\_acc\_age\ (MDS)]}{Lbd(MDS)}$	**rec_acc_age** from section 2.5 **LBD** from section **2.1**
Availability	$Availability\ (MDS) = \frac{[n\_succ\_req\ (MDS)]}{n\_req(MDS)}$	**n_succ_req** from section **2.6** **n_req** from section **2.7**
Mver	$Mver\ (MDS) = Accuracy\ (MDS) * W_{Acc} + Completness(MDS) * W_{Comp} + Currentness\ (MDS) * W_{Curr} + Availability * W_{Avail}$	
Compliance	$Compliance\ (MDS) = \frac{\sum_{\forall\ DS \in MDS} Nrec_{comp}(DS)}{Nds(MDS)}$	**Nrec_comp** from section **2.8** **Nds** From section **2.2**
Credeability	$Credability\ (MDS) = \frac{Nds_{cr}(MDS)}{Nds\ (MDS)}$	**Nds_cr** from section **2.9** **Nds** From section **2.2**
Mval	$Mval\ (MDS) = Credability\ (MDS) * W_{Cred} + Compliance(MDS) * W_{Compli}$	
Traceability	$Traceability\ (DS) = \frac{Rec_{Trace}(DS)}{Ldst\ (DS)}$	**Rec_trace** from section **2.10** **Ldst** from section **2.11**
Mvin	$Mvin\ (MDS) = \frac{\sum_{\forall\ DS \in MDS} Traceability\ (DS)}{Nds\ (MDS)}$	

Step 1:

The value of derived measure(s) calculated at each team frame in step 1 is shown in the below table.

	T1	T2	T3
Accuracy	0.9999999982382612	0.9999999997238851	0.9999999999088572
Completeness	1.0	1.0	1.0
Currentness	0.59	0.6052104208416834	0.6239015817223199
Availability	0.75	0. 8522727272727273	0.8586956521739131
Compliance	1.0	1.0	1.0
Credibility	1.0	1.0	1.0
Traceability	1.0	1.0	1.0

Step 2:

The values of derived measures calculated at each process of the big data pipeline namely Data Extraction, Data Pre-processing / Data Cleaning, and Data Processing in three different time frames namely T1, T2, and T3 are shown in the below table.

Big Data V's / Time frames	T1			T2			T3		
Big Data Pipeline Process	Data Extraction	Data Cleaning	Data Processing	Data Extraction	Data Cleaning	Data Processing	Data Extraction	Data Cleaning	Data Processing
Accuracy	0.9999999982	0.9999999982	0.9999999982	0.9999999997	0.9999999997	0.9999999997	0.9999999999	0.9999999999	0.9999999999
Completeness	1	1	1	1	1	1	1	1	1
Currentness	0.59	0.59	0.59	0.511679644	0.511679644	0.511679644	0.5190735695	0.5190735695	0.5190735695
Availability	0.75	0.75	0.75	0.8522727273	0.8522727273	0.8522727273	0.8586956522	0.8586956522	0.8586956522
Compliance	1	1	1	1	1	1	1	1	1
Credibility	1	1	1	1	1	1	1	1	1
Traceability	1	1	1	1	1	1	1	1	1

4.2 Values of the V's (Veracity, Validity, and Vincularity) at each time frame / data pipeline phase

Step 1:

The value of 3 v's calculated at each team frame in step 1 (Data Extraction Process) is shown in below table.

Big Data V's / Time frames	T1	T2	T3

Big Data Veracity	0.835	0.840988093	0.844442305
Big Data Validity	1.00	1.00	1.00
Big Data Vincularity	1.00	1.00	1.00

Step 2:

The value of 3 v's calculated at each process of the big data pipeline namely Data Extraction, Data Pre-processing / Data Cleaning, and Data Processing in three different time frames namely T1, T2, and T3 are shown in the below table.

Big Data V's / Time frames	T1			T2			T3		
Big Data Pipeline Process	Data Extraction	Data Cleaning	Data Processing	Data Extraction	Data Cleaning	Data Processing	Data Extraction	Data Cleaning	Data Processing
Big Data Veracity	0.835	0.835	0.835	0.84098809	0.840988	0.840988093	0.84444231	0.844442	0.844442305
Big Data Validity	1	1	1	1	1	1	1	1	1
Big Data Vincularity	1	1	1	1	1	1	1	1	1

4.3 Average Value of each of the V's (Veracity, Validity, and Vincularity) at the end of the process

Step 1:

The average value of each of the V's at the end of each time frame of step 1 (Data Extraction Process) is as follows:

Big Data V's / Time frames	T1	T2	T3
Big Data Veracity	0.835	0.840988093	0.844442305
Big Data Validity	1.00	1.00	1.00
Big Data Vincularity	1.00	1.00	1.00

Step 2:

The average value of each of the V's at the end of each time frame namely T1, T2 and T3 is shown below in the table.

Big Data V's / Time frames	T1	T2	T3
Big Data Veracity	0.835	0.840988	0.844442
Big Data Validity	1	1	1
Big Data Vincularity	1	1	1

4.4 Final Value of each of the V's (Veracity, Validity, and Vincularity) at the end of the process

Step 1:

The final value of each of the V's at the end of each time frame of step 1 (Data Extraction Process) is as follows:

Big Data V's / Time frames	T1	T2	T3
Big Data Veracity	0.835	0.840988093	0.844442305
Big Data Validity	1.00	1.00	1.00
Big Data Vincularity	1.00	1.00	1.00

Step 2:

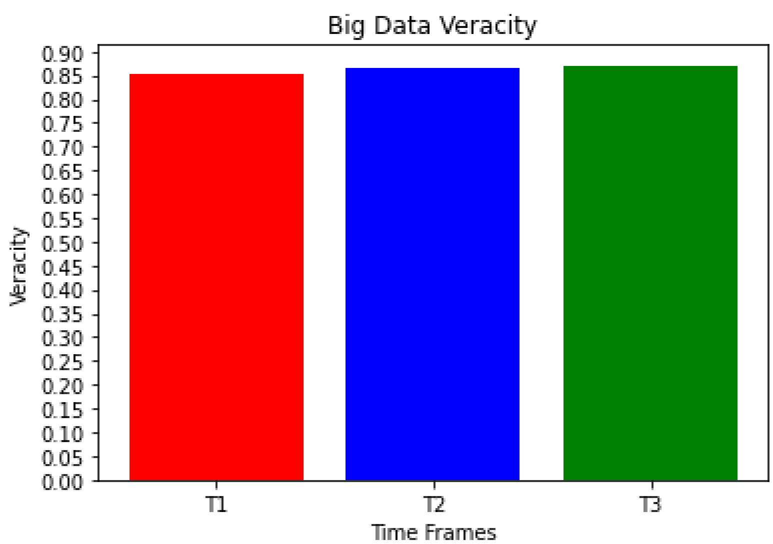
The final value of each of the V's at the end of each big data pipeline at time frames namely T1, T2, and T3 is shown below in the table.

Big Data V's / Time frames	T1	T2	T3
Big Data Veracity	0.835	0.840988	0.844442
Big Data Validity	1	1	1
Big Data Vincularity	1	1	1

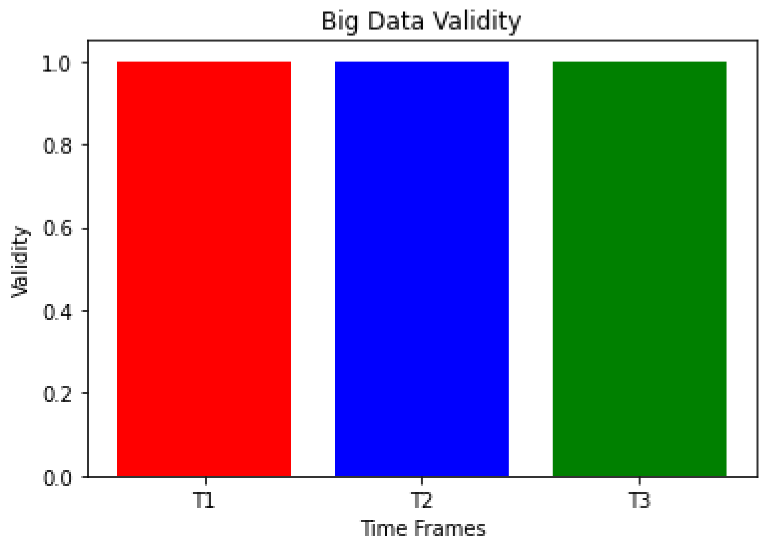
4.5 Draw the graphs of the indicators Mver, Mval, and Mvin generated from the values of the derived measures.

Step 1:

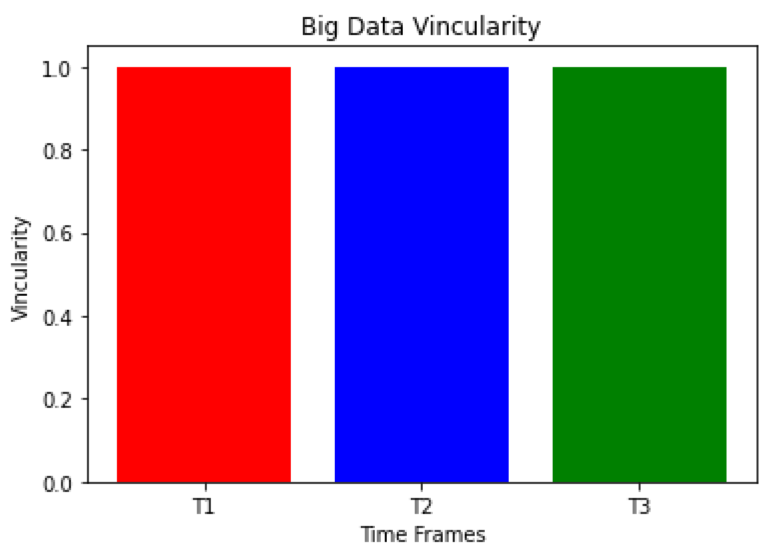
The graph of the Big data Veracity's indicator Mver at each time frame T1, T2, and t3 is shown below.



The graph of the Big data Validity indicator Mval at each time frame T1, T2, and t3 is shown below.

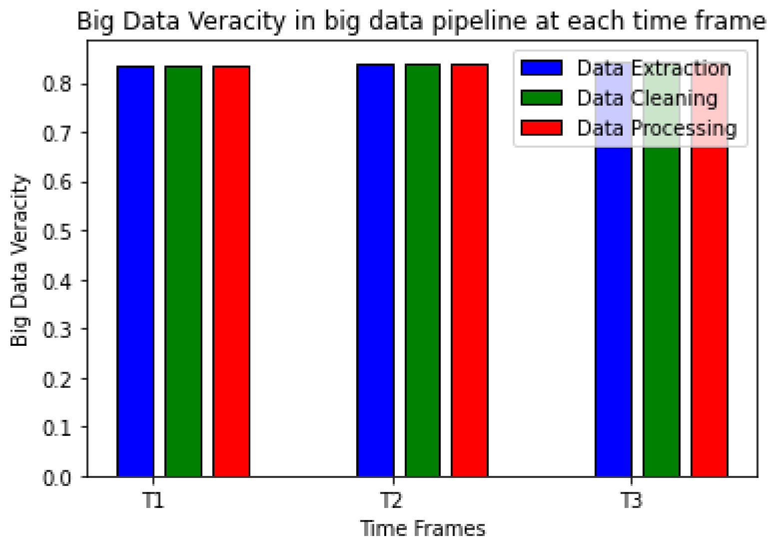


The graph of the Big data Vincularity indicator Mvin at each time frame T1, T2, and t3 is shown below.

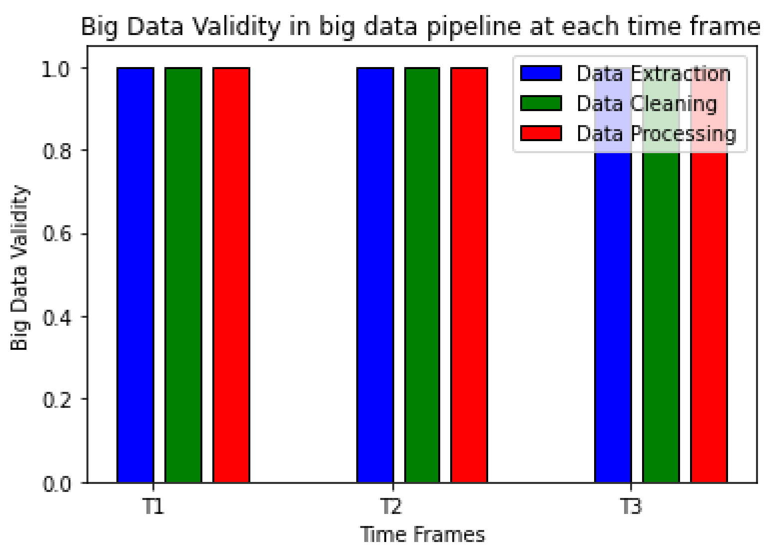


Step 2:

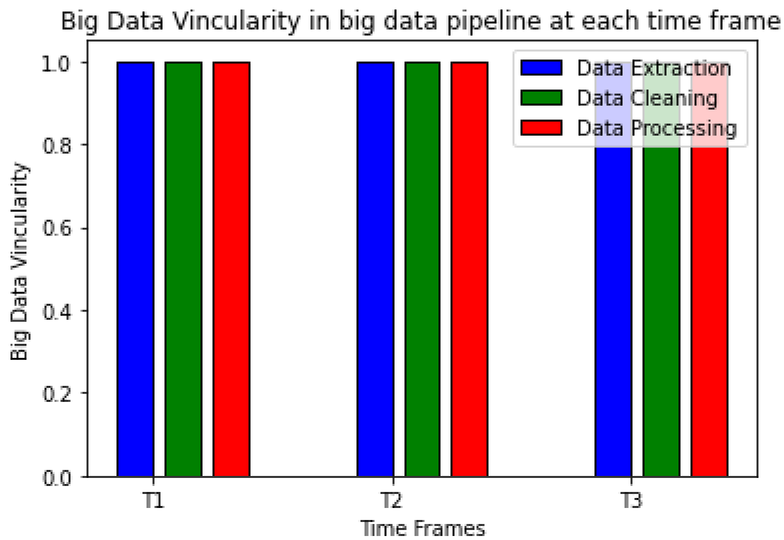
The graph of the Big data Veracity indicator Mver at the end of each big data pipeline namely Data Extraction, Data Cleaning / Data Pre-processing, and Data Processing in all time frames T1, T2, and T3 are shown below.



The graph of the Big data Validity indicator Mval at the end of each big data pipeline namely Data Extraction, Data Cleaning / Data Pre-processing, and Data Processing in all time frames T1, T2, and T3 are shown below.



The graph of the Big data Vincularity indicator Mvin at the end of each big data pipeline namely Data Extraction, Data Cleaning / Data Pre-processing, and Data Processing in all time frames T1, T2, and T3 are shown below.


Big Data Vincularity in big data pipeline at each time frame
Data Extraction
Data Cleaning
Data Processing
Big Data Vincularity
1.0
0.8
0.6
0.4
0.2
0.0
T1
T2
T3
Time Frames

5. Conclusion

The dataset IBM HR analytics Employee Attrition & Performance can be used for the machine learning algorithms. We split the dataset into three different time frames namely T1, T2, and T3, and then pass these data frames into the big data pipeline in each of these time frames. Thus, calculating the values of Big Data Veracity, Validity, and Vincularity in the big data pipeline at each time interval provides some vital information about data cleaning that the data frame in each time frame has no null values, no duplicate records, and no mismatched data elements in any column. Thus we can observe the values of these big data indicators do not change after each big data pipeline process in each consecutive time frame.

Comparing the first and second analyses, we cannot observe any significant change in the values of big data indicators. As mentioned above the data in the dataset was already cleaned and need not need any further cleaning, still, we performed some operations of data cleaning, and we did not find any changes. So, there was no difference in the quality of the big data when compared between Step 1 and Step 2. Although we can observe that the values of big data indicators do increase with each passing time frame in both step 1 and step 2.

The measures like N_succ_req which is the total number of successful requests made by some authorized entity like server, API calls, and N_req which is the total number of requests made. We do not have any exact figures for these base measures which forces us to assume the values of the measures. With a specific value, the availability of the dataset can be increased which then increase the veracity of the big data. Thus, having these attribute values would have increased the quality of big data.

6. Project Code Link

https://colab.research.google.com/drive/1tAuzSf3D1fQHTizFuHXnXYoWMToKMnQd?usp=sharing