Data Mining Project Phase 2

Faridreza Momtaz Zandi 9812762601 Alireza Noorbakhsh 9812762496

In this phase, our main goal is to work on our data and have a clear mindset of how is our dataset's quality. We'll achieve this goal by evaluating five inherent quality attributes: Accuracy, Completeness, Consistency, Credibility, and Currentness. Only after using all these models, we can have a full understanding on how's the quality of our data but we'll see later on that most of our data isn't suitable for our data sets.

Before we start on our datasets let's see what are these five quality model attributes. Accuracy: The degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use. Completeness: The degree to which subject data associated with an entity has values for all expected attributes and related entity instances in a specific context of use. Consistency: The degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use. It can be either or both among data regarding one entity and across similar data for comparable entities. Credibility: The degree to which data has attributes that are regarded as true and believable by users in a specific context of use. Credibility includes the concept of authenticity (the truthfulness of origins, attributions, and commitments). Currentness: The degree to which data has attributes that are of the right age in a specific context of use.

INOUT Dataset

As we said before in phase one there are no specific and clear attributes to find the quality of it and even if we find the null count or other quality model attributes we won't be able to use this data in any useful way. So it's better to skip this dataset and get to one with more sense.

INOUTLINE Dataset

Again like in the last Phase, now in this dataset, we can work on some attributes which we have found before. these attributes are ACCUMULATEDEPRECIATION, BOOKVALUE, PRIMALVALUE and DEPRECATION_PERIOD. first, we'll find the total count of their record and null counts and then we'll talk about our quality model attributes.

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
```

```
In [2]: df = pd.read_csv('E:\ce\data mining\DataSets\INOUTLINE.csv' , low_memory = False)
        df = df[['ACCUMULATEDEPRECIATION' , 'BOOKVALUE' , 'PRIMALVALUE' ,'DEPRECATION_PERIO
        print('\033[94m Lets see these columns info first, this gives us total record and n
        print('\033[90m')
        print(df.info())
        print('\033[95m So our null count is:')
        print('\033[90m')
        print(df.isnull().sum())
```

```
 Lets see these columns info first, this gives us total record and non-null count:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103410 entries, 0 to 103409
Data columns (total 4 columns):
 #   Column                  Non-Null Count   Dtype  
---  ------                  --------------   -----  
 0   ACCUMULATEDEPRECIATION  7680 non-null    float64
 1   BOOKVALUE               66503 non-null   float64
 2   PRIMALVALUE             103342 non-null  float64
 3   DEPRECATION_PERIOD      4 non-null       float64
dtypes: float64(4)
memory usage: 3.2 MB
None
 So our null count is:

ACCUMULATEDEPRECIATION     95730
BOOKVALUE                  36907
PRIMALVALUE                   68
DEPRECATION_PERIOD        103406
dtype: int64
```

Accuracy: for example when we look at ACCUMULATEDEPRECIATION attributes there is no data out of acceptable range so we can imagine accuracy is considered and there is no other way to make sure cause we do not know the exact or wished range! this is also true for other chosen attributes.

Completeness: for each attribute, we just need to find the non-null to all ratio so we have: ACCUMULATEDEPRECIATION: 7.42%, BOOKVALUE: 64.31%, PRIMALVALUE: 99.94% and DEPRECATION_PERIOD: 0.004%.

Consistency: after taking a look at our attributes we see DEPRECATION_PERIOD, BOOKVALUE and PRIMALVALUE are value types, and there is no specific rule that we can have for them but on the other hand DEPRECATION_PERIOD which is Depreciation period might be in contact with other attributes such as ENDOFUSEFULLIFE.

Credibility: for this quality model attribute we take the same approach we did with accuracy and while having not enough knowledge about our dataset there is no way to have a valid evaluation.

Currentness: We can only accept what our data provider promised and there isn't any other way to check if our data is up to date or even updated enough to not put us in any trouble.

PRODUCTINSTANCE Dataset

We proceed like the previous dataset and first take a look at our chosen attributes from the last phase which are: PRICE9, SALVAGEVALUE, PI_VALUEAFTERCOEFFICIENTINC, COEFFICIENTVALUE, ANTIQUITYCOEFFICIENT, PRESENTVALUE, BOOKVALUE and AREA_TOTAL.

In [3]:
```
df = pd.read_csv('E:\ce\data mining\DataSets\PRODUCTINSTANCE.csv' , encoding='cp125
df = df[['PRICE9' , 'SALVAGEVALUE' , 'PI_VALUEAFTERCOEFFICIENTINC' , 'COEFFICIENTVA
       , 'PRESENTVALUE' , 'BOOKVALUE' , 'AREA_TOTAL']]
print('\033[94m Lets see these columns info first, this gives us total record and n
print('\033[90m')
print(df.info())
print('\033[95m So our null count is:')
print('\033[90m')
print(df.isnull().sum())
```

```
 Lets see these columns info first, this gives us total record and non-null count:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 706204 entries, 0 to 706203
Data columns (total 8 columns):
 #   Column                       Non-Null Count   Dtype  
---  ------                       --------------   -----  
 0   PRICE9                       660 non-null     float64
 1   SALVAGEVALUE                 6795 non-null    float64
 2   PI_VALUEAFTERCOEFFICIENTINC  3332 non-null    float64
 3   COEFFICIENTVALUE             3105 non-null    float64
 4   ANTIQUITYCOEFFICIENT         3105 non-null    float64
 5   PRESENTVALUE                 834 non-null     float64
 6   BOOKVALUE                    130898 non-null  float64
 7   AREA_TOTAL                   388455 non-null  float64
dtypes: float64(8)
memory usage: 43.1 MB
None
 So our null count is:

PRICE9                         705544
SALVAGEVALUE                   699409
PI_VALUEAFTERCOEFFICIENTINC    702872
COEFFICIENTVALUE               703099
ANTIQUITYCOEFFICIENT           703099
PRESENTVALUE                   705370
BOOKVALUE                      575306
AREA_TOTAL                     317749
dtype: int64
```

This time we're checking eight attributes but almost all quality model attributes work the same as our last dataset for example there isn't any specific data about some attributes such as PRICE9, SALVAGEVALUE, and PI_VALUEAFTERCOEFFICIENTINC and with limited knowledge like this, there is no way to have a Safe choice for accuracy and credibility.

Completeness: unlike accuracy and credibility we can have an almost accurate rate for completeness which is: PRICE9: 0.01%, SALVAGEVALUE: 0.96%, PI_VALUEAFTERCOEFFICIENTINC: 0.47%, COEFFICIENTVALUE: 0.43%, ANTIQUITYCOEFFICIENT: 0.43%, PRESENTVALUE: 0.12%, BOOKVALUE: 18.54% and AREA_TOTAL : 55.01%.

Again there isn't enough description in our dictionary about these attributes so we can't make sure our data is consistent. And once again we need to believe our data provider whatever he says about the currentness of our data!

TRANSFER_ITEM and TRANSFER_ITEM_D Datasets

As we talked about before there isn't any useful insight we can from these datasets even if we check it's record count or the null count.

For the next part of this phase, we wish to find some examples of Data Quality Problems which we categorize into two main groups: Single-Source Problems(schema/instant) and Multi-Source Problems(Shema/instant).

Single-Source Problems

The data quality of a source largely depends on the degree to which it is governed by schema and integrity constraints controlling permissible data values. Schema-related data quality problems thus occur because of the lack of appropriate model-specific or application-specific integrity constraints, e.g., due to data model limitations or poor schema design, or because only a few integrity constraints were defined to limit the overhead for integrity control. Instance-specific problems relate to errors and inconsistencies that cannot be prevented at the schema level (e.g., misspellings). Now let's try to find some single-source problems in our datasets.

One of the most obvious single source problems are missing values in attribute scope and is an instance level problem which we saw in the last part how much of an attribute is null.

If we look closely at INOUT dataset and especially at the DESCRIPTION attribute we see some records are integer but are shown in a way that can't be accessed and somehow has some question marks before and after the number we need! this is also an instance level problem.

We can find an embedded value problem in the PRODUCTINSTANCE dataset and PARENT_ID attribute which we have both string and integer attributes and this is a problem.

Multi-Source problems

The problems present in single sources are aggravated when multiple sources need to be integrated. Each source may contain dirty data and the data in the sources may be represented differently, overlap, or contradict. This is because the sources are typically developed, deployed, and maintained independently to serve specific needs.

One of the things we need to check for multi-source problems is the probability of causing a problem in case of merging datasets or extracting a single product data from different datasets, for example, if we merge INOUTLINE dataset with TRANSFER_ITEM, both have C_YEAR_ID attribute which is Fiscal year and we may get the same record but with a different value for C_YEAR_ID. We can use this situation to show one more probable multi-source problem which is showing the date in different ways such as using the year in a full format like 2022 or only the second two digits like 22.

If we look closely at INOUTLINE data set and DISCRIPTIONKALA attribute we can guess having two languages for records might have happened because of a multi-source problem.

When we examine the product dataset we'll find it very messy and it can be an example of multi-source problems. Almost any problem at any level can be found! other than lots of null records and missing data it suffers from being inconsistent.

suggestions for data quality improvement

In my opinion, the biggest flaw of our datasets is the missing values, as we saw in this phase we're suffering from a huge rate of missing data even in our critical records, so my first suggestion is to handle these missing data.

The second problem with these datasets is how much messy it is! especially with datasets like PRODUCTS. If data is too dirty we can't have a clear solution to how to use this data so we have to clean it. This cleaning can be filling out null records or removing noises and even Correcting inconsistent data.

The last thing that comes to mind when working on our datasets is that it was really to our advantage if we had a better dictionary or at least more information about our attributes. It's frustrating figuring out a dataset with nest to minimum knowledge about it.