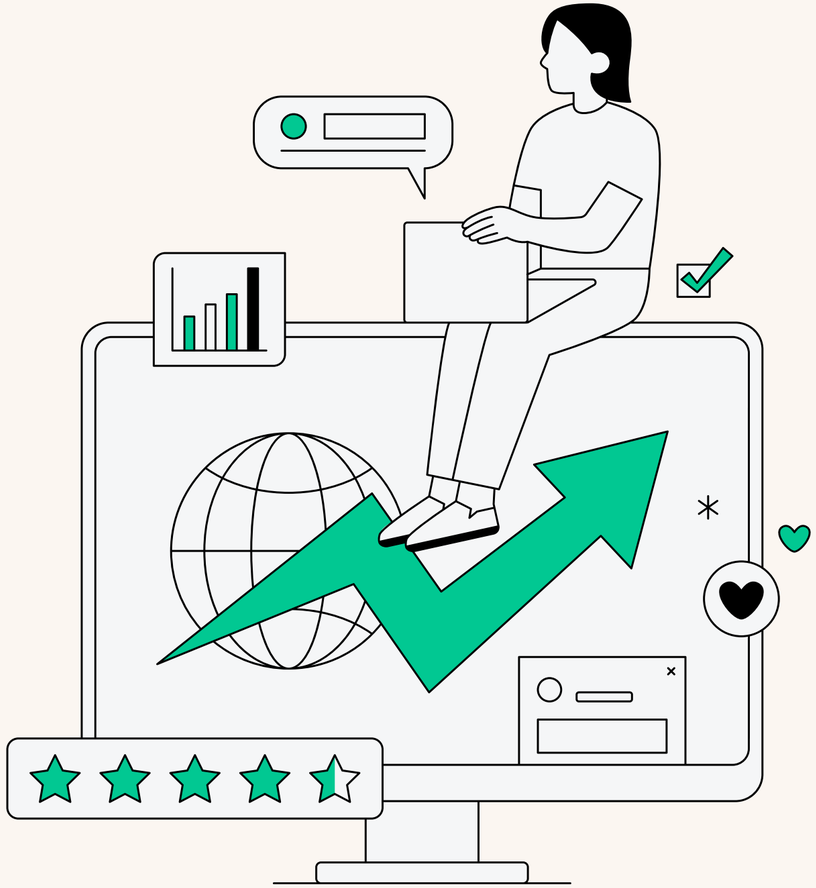
Presented by Ahmad Alqaisi

Claims Data Exploration

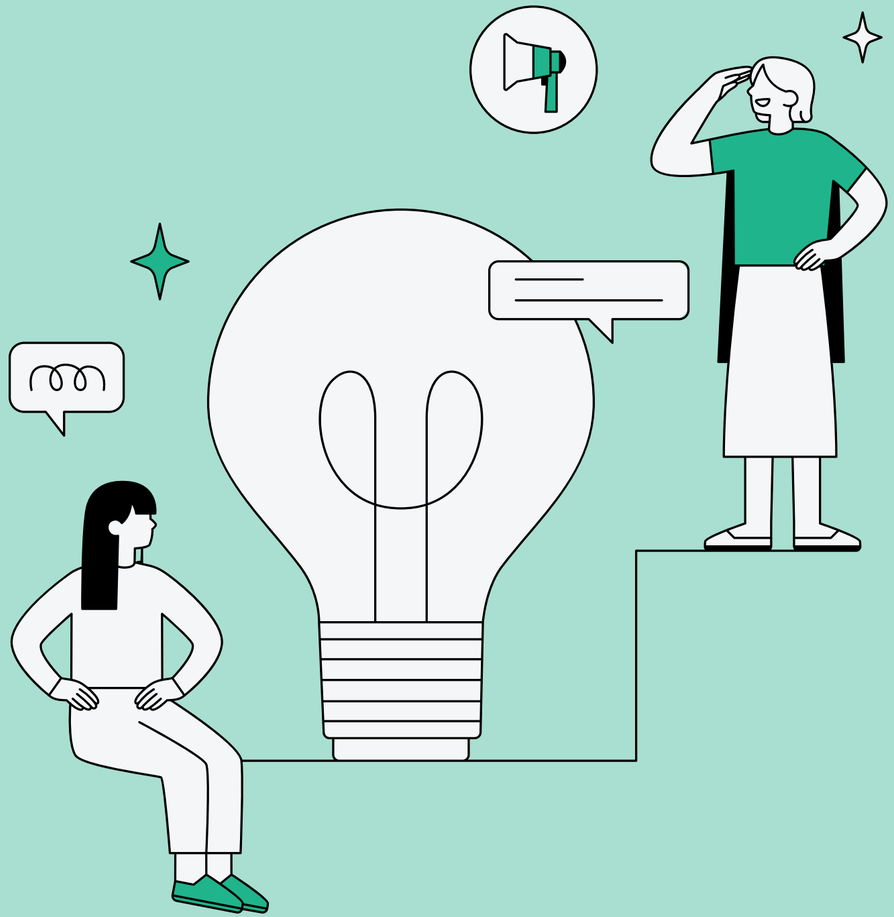
Identifying Key Insights and Trends

Monday, July 8, 2024



Outlines

- **Getting Familiar with Data**
  - Data Structure
  - Data splitting
  - Data Pre-processing Steps
  - Overview of the Dataset
- **Collectors Performance Analysis**
- **Collection Amount Analysis**
- **Takeaways**



Data Structure

The table below shows the structure of the dataset. It has a *shape* of (33219, 8) and there are some missing values.

Two of them are identity variables **(DebitID and CollectorID)** and two of them are continuous numerical variables **(Amount** and **CollectionAmount)** representing the original value of the debit and the collected debit value.

One of which is discrete variable **(Number of Contact)** representing how many times the collector reach the debtor.

One variable is categorical one **(DebitType).**

Lastly, there are two datetime variables **(EnterDate and DateTrans)** representing the time the claim was issued and when the debtor pays the claim or an installment thereof.

#	Column	Non-Null Count	Dtype
0	EnterDate	33219 non-null	object
1	DebitID	33219 non-null	int64
2	Number of Contact	33219 non-null	int64
3	DebitType	32298 non-null	object
4	Amount	33219 non-null	float64
5	CollectorID	33218 non-null	float64
6	CollectionAmount	33202 non-null	float64
7	DateTrans	33219 non-null	object

Data Splitting

Next, I have split the dataset into **training** and **testing** dataset at this early time to avoid what is so called *data snooping bias*,.

The model learns and identify any trends and patterns from the training set, while it is tested for performance on an unseen testing dataset.

```
X_train, X_test = train_test_split(X)
X_train.shape
```

```
(24914, 8)
```

About 75% of the dataset are training dataset and the remaining are for testing.
The resulted train set has 24914 rows.

Data Pre-processing Steps

Data Cleansing

Involving checking for any missing values, duplicated rows, outliers or any something else that makes the data ready to analysis:

```
X_train.isna().sum().sum(), X_train.duplicated().sum()
```

```
(715, 9)
```

There are 715 missing values and 9 duplicated rows.

In addition, all collected amounts should be less than or equal to the actual claim value, so let's discard any record that dose not mach this condition:

```
print(f"{(X_train.CollectionAmount > X_train.Amount + 1).sum()}"
      " records dose not mach the above condition.")
```

```
528 records dose not mach the above condition.
```

Data Pre-processing Steps

Data Transformation

For consistency, I have cast the date variables into pandas datetime type and CollectorID to int type instead of floating point decimal.

The resulted dataset structure is:

```
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   EnterDate          23670 non-null  datetime64[ns]
 1   DebitID            23670 non-null  category
 2   Number of Contact  23670 non-null  int64
 3   DebitType          23670 non-null  object
 4   Amount             23670 non-null  float64
 5   CollectorID        23670 non-null  category
 6   CollectionAmount   23670 non-null  float64
 7   DateTrans          23669 non-null  datetime64[ns]
dtypes: category(2), datetime64[ns](2), float64(2), int64(1), object(1)
```

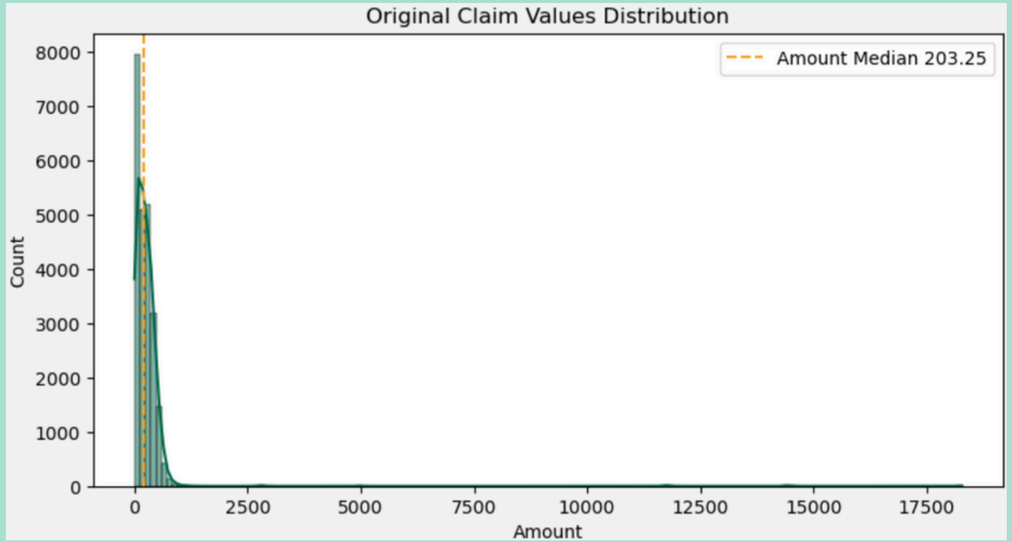
As shown, the data is now cleaned and ready for analysis.

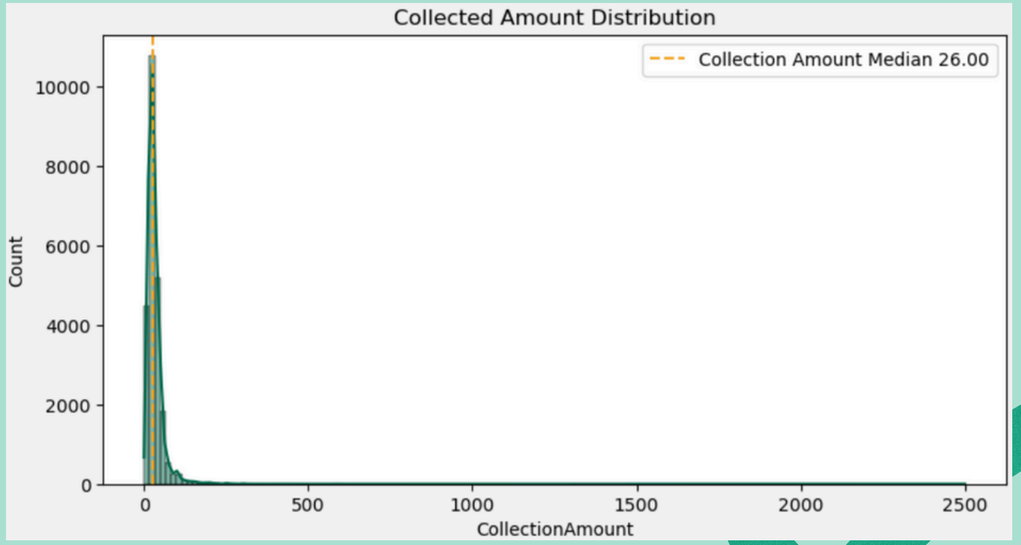
Overview of the Dataset

Consider the following statistical summary table of the numerical attributes:

	count	mean	std	min	25%	50%	75%	max
Number of Contact	23686.0	6.101790	6.518927	0.0000	2.000000	4.0000	8.000000	79.00
Amount	23686.0	269.375977	710.018365	0.1169	83.506925	201.3871	346.532125	18283.21
CollectionAmount	23686.0	33.142817	45.465084	0.0010	18.550000	26.1000	39.440000	4500.00

The results of standard deviations shows the data has a very high variability!
and the quartiles shows an extreme right skewness.





Overview of the Dataset

The median number of contacts was 4, and some of debits reach a 79 number of contacts. Claim values are ranging up to 18283 JOD with a median of 194 JOD, indicating a diverse portfolio of debit amounts.

Finally, the collected debit values are ranging up to 4500 JOD with a median of 26.5 JOD. This again points to a wide distribution, where most collections are relatively small, but there are a few cases of much larger collected amounts.

Overview of the Dataset

In addition, there are 2512 unique debit types and the most frequent debit is of type *100GB Evo 4G Extra Free*:

	count	unique	top	freq
DebitType	23686	2482	100GB Evo 4G Extra Free	2209
CollectorID	23686.0	15.0	19.0	5838.0
DebitID	23686	20196	445	25

On the right, the world cloud shows the frequency distributions of Debit types .

Overview of the Dataset

Finally, no claims were made for some collections, while a portion of the claim value has been collected. This may pose a risk in the future and also affect the reliability of the data.

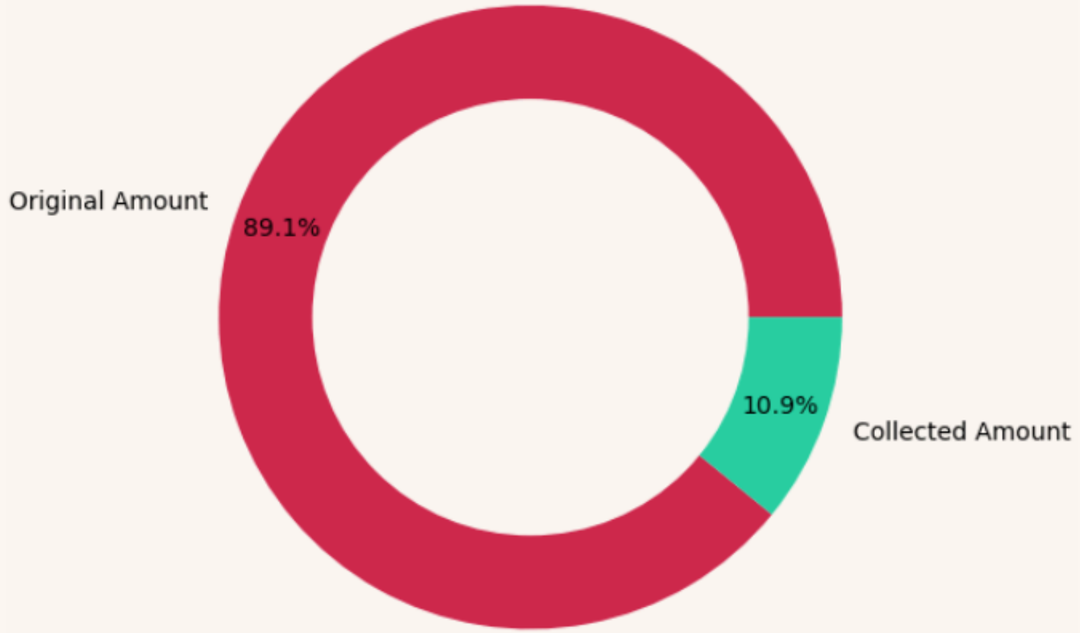
	CollectionAmount
count	168.0000
sum	5229.8151
min	0.0100
max	231.7730

There are 168 claim are collected with 0 number of contacts.
The total amount of them were 5230 JOD ranging from 0 to 231 JOD..

10.9%

From the original claim values were collected!

Achievement: Collected Amount to Original Amount



Collectors Performance

Consider the following table about how each collector performs with respect to claims that responsible to do:

	CollectorID	CollectionAmount sum	Amount sum	median	Ratio
0	7.0	3251.2690	1.097505e+05	18283.21000	2.962418
1	9.0	99.5000	9.900000e+02	990.00000	10.050505
2	10.0	3334.3450	3.348248e+04	135.88000	9.958475
3	14.0	103835.8038	8.810901e+05	251.39985	11.784924
4	15.0	5081.9106	5.264894e+04	376.37240	9.652446
5	16.0	144477.1740	9.718127e+05	198.50250	14.866772
6	17.0	21248.2480	8.865366e+05	2800.00000	2.396771
7	18.0	4109.7160	3.953348e+04	110.20230	10.395534
8	19.0	200310.2764	1.359646e+06	193.58260	14.732534
9	20.0	31603.5845	1.784826e+05	155.42800	17.706815
10	21.0	118251.2973	9.052831e+05	228.76970	13.062355
11	27.0	52.9150	5.291480e+01	52.91480	100.000378
12	28.0	15618.8843	1.459096e+05	232.63230	10.704493
13	30.0	113627.1053	7.887466e+05	187.18175	14.406034
14	45.0	16980.8179	1.187226e+05	146.76700	14.302940

This table shows a noticeable decline in the level of performance!

The median is 11.78, which means that claims collector attempts collect about 11.78% of the total amount of claims they contact.

One of them has a ratio of 100% with **CollectorID** 27 indicating that he collects all claims he is responsible for, which is just one claim of 52 JOD.

The **Ratio** is diverse based on the nature of claims amount they responsible to collect.

The collector of ID 19 has the largest collected amount of 200310 JOD of 1359646 JOD.

Collectors Performance

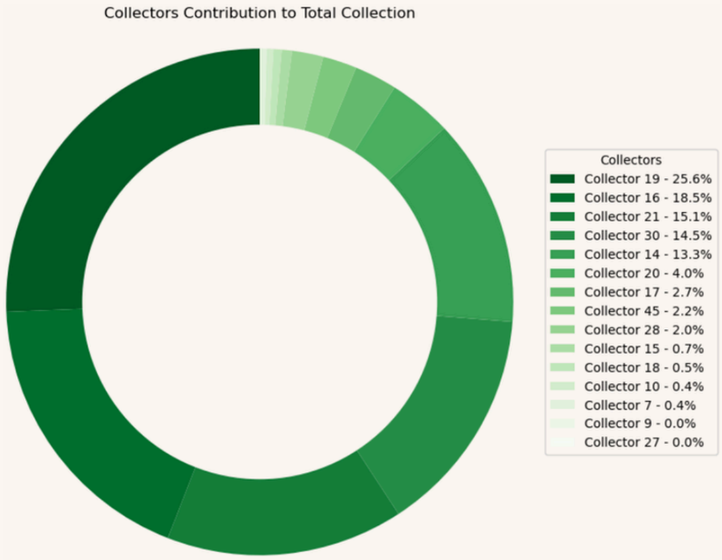
Overall performance is poor in terms of original claim values.

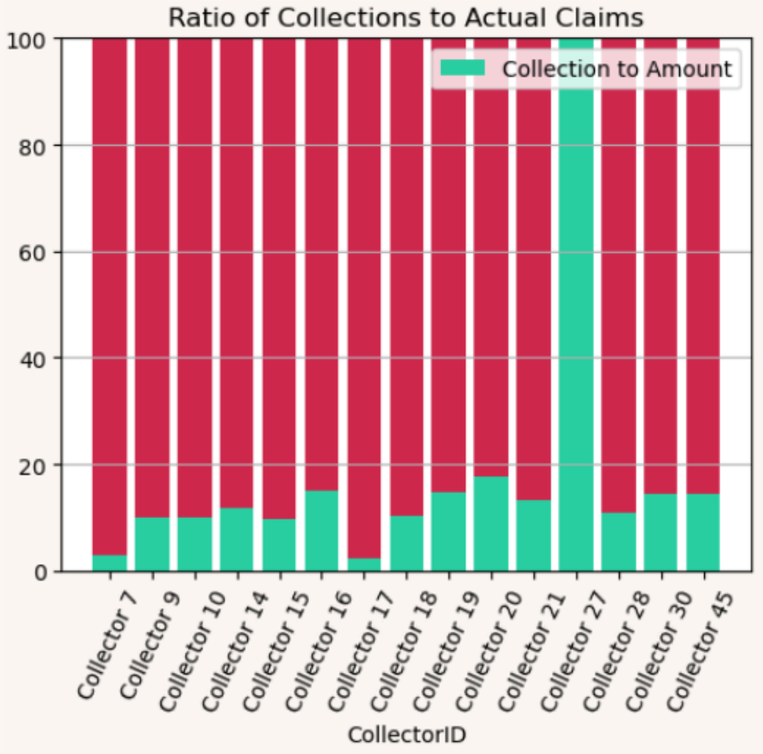
The collector numbered 19 contributes 26% of the total amount collected, but at the same time it collects about 14.7% of the total amount it is responsible for.

On the other hand, the collector numbered 20, which complete about 19% of the total amount he is responsible for, contributes about 4% from total amount collected.

This variation in performance is due to the distribution of claim values performed by each collector.

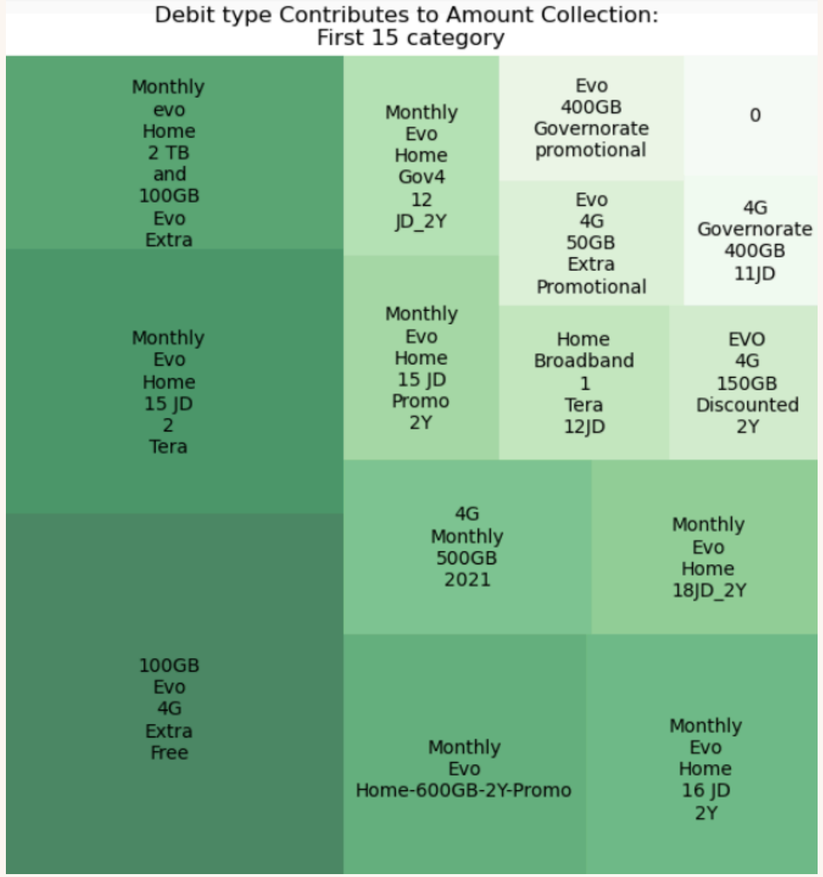
However, more monitoring and robust collecting strategies should be implemented.





Collectors Performance

Next, I showing the most debt types contributes to the overall collected amount:



They are summarized in the table below:

DebitType	CollectionAmount
100GB Evo 4G Extra Free	74541.3997
Monthly Evo Home 15 JD 2 Tera	53382.3199
Monthly evo Home 2 TB and 100GB Evo Extra	39166.3106
Monthly Evo Home-600GB-2Y-Promo	34979.1331
Monthly Evo Home 16 JD 2Y	34711.5481
4G Monthly 500GB 2021	25015.9932
Monthly Evo Home 18JD_2Y	23610.8051
Monthly Evo Home 15 JD Promo 2Y	18528.8186
Monthly Evo Home Gov4 12 JD_2Y	18090.9204
Home Broadband 1 Tera 12JD	15205.2027
EVO 4G 150GB Discounted 2Y	14062.8383
Evo 4G 50GB Extra Promotional	13454.7100
Evo 400GB Governorate promotional	13300.3369
4G Governorate 400GB 11JD	10853.2125
0	9845.9450

The most debit types with largest collected amounts was *100GB Evo 4G Extra Free* with 74541 JOD.

Collection Amount Analysis

Well, the data was start collected since of 2020-08-18 to 2024-06-02.

```
X_train.EnterDate.min(), X_train.EnterDate.max()
```

```
(Timestamp('2020-08-18 13:04:02.693000'), Timestamp('2024-06-02 00:00:00'))
```

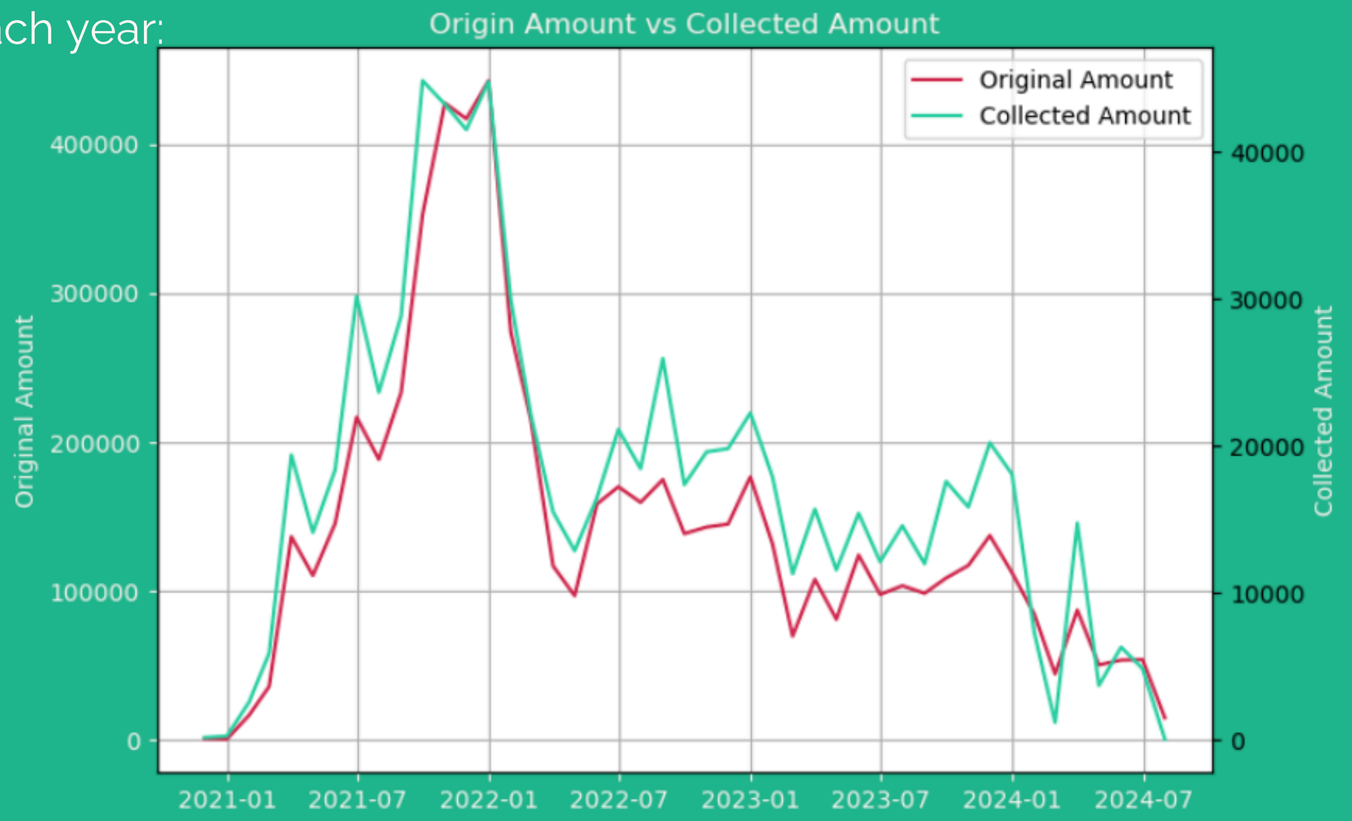
However, normally, the entry date should always be greater than the transaction date, so let's check that this condition is true and discard any observation that does not meet it:

```
print(f"{(X_train.DateTrans < X_train.EnterDate).sum()} un-matched records.")
X_train.drop(X_train[X_train.DateTrans < X_train.EnterDate].index, inplace=True)
```

```
78 un-matched records.
```

Collection Amount Analysis

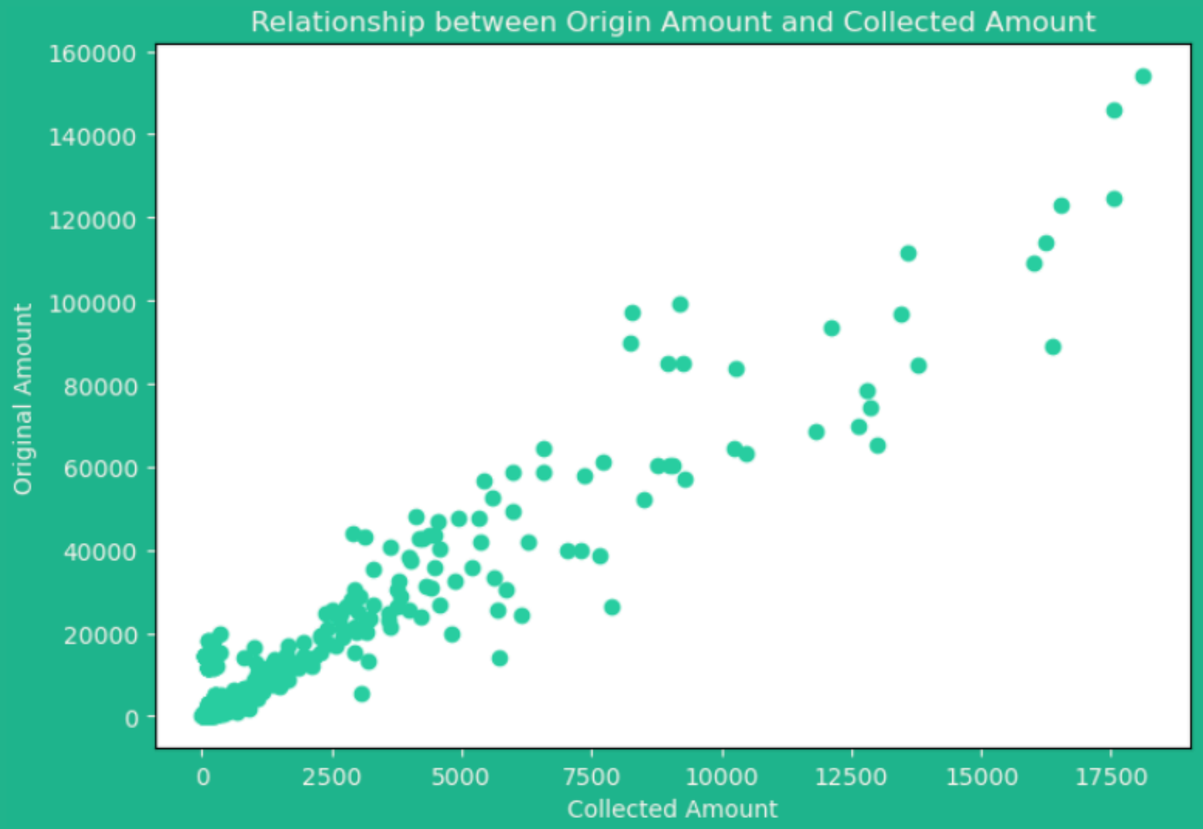
Now, let's examine the performance at each year:



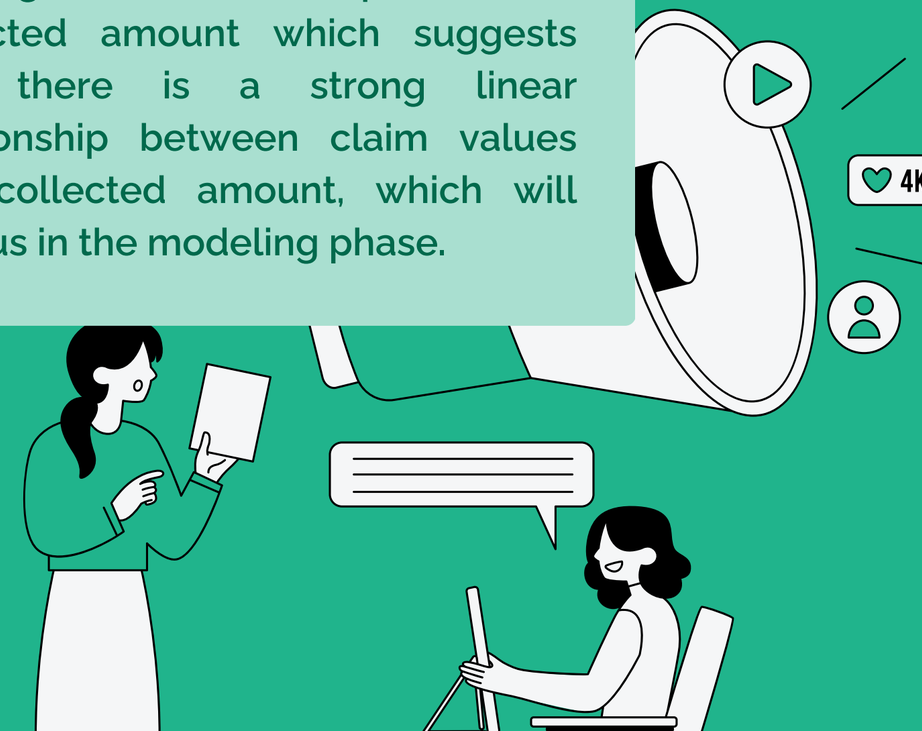
This says a lot!

The collection amount lines changes as the original amount values in the first line chart, which indicates there is some sort of relationship between the collected amount and the value of the original amount.

Collection Amount Analysis

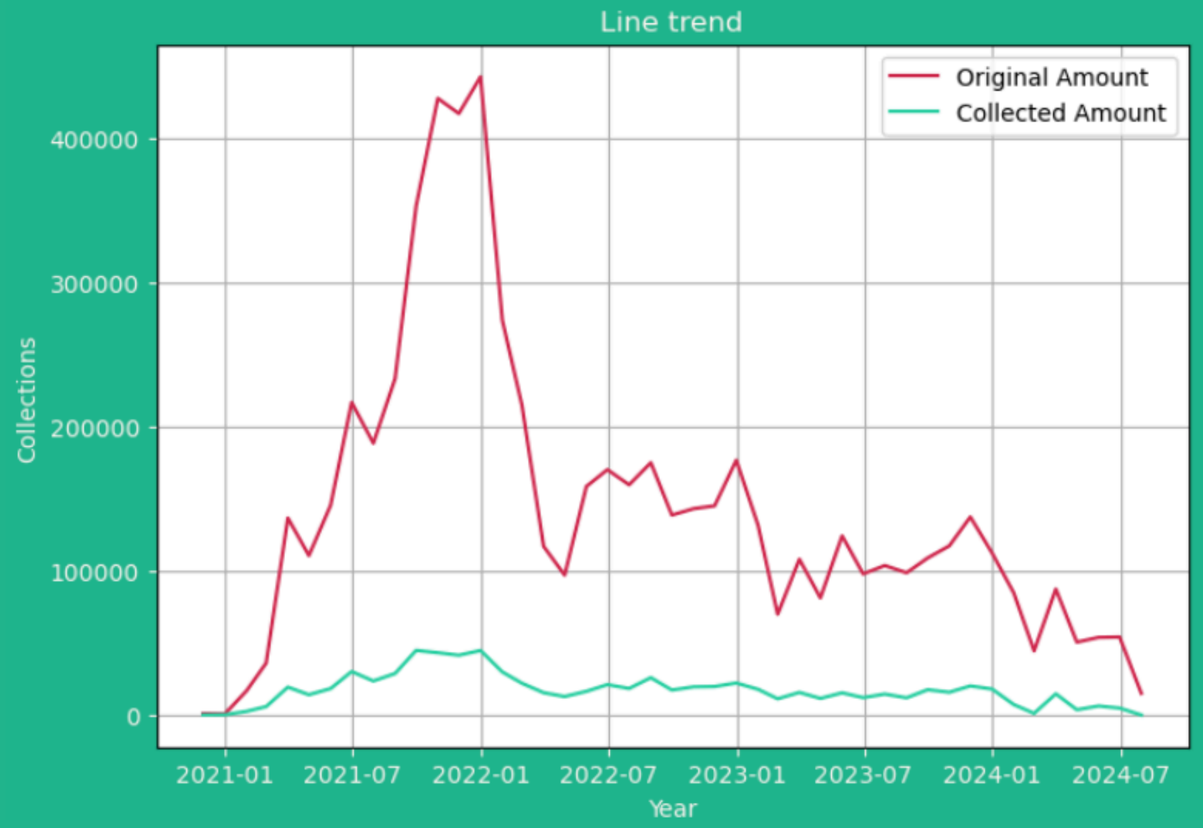


The scatter plot of the collected amounts and original claims values on the left make it is easier to to understand that how the values of the original claim can impact on the collected amount which suggests that there is a strong linear relationship between claim values and collected amount, which will help us in the modeling phase.



Collection Amount Analysis

Well, the first line graph seems a bit misleading if you don't interpret the scales of unit carefully, the following line graph illustrates this and make it more clearer:



Collection Amount Analysis

This finding opens the doors and make it possible for thinking in a linear models solutions at the modeling phase.

We can calculate the person coefficient between them:

```
grouped_by_ymd_collections.corr()
```

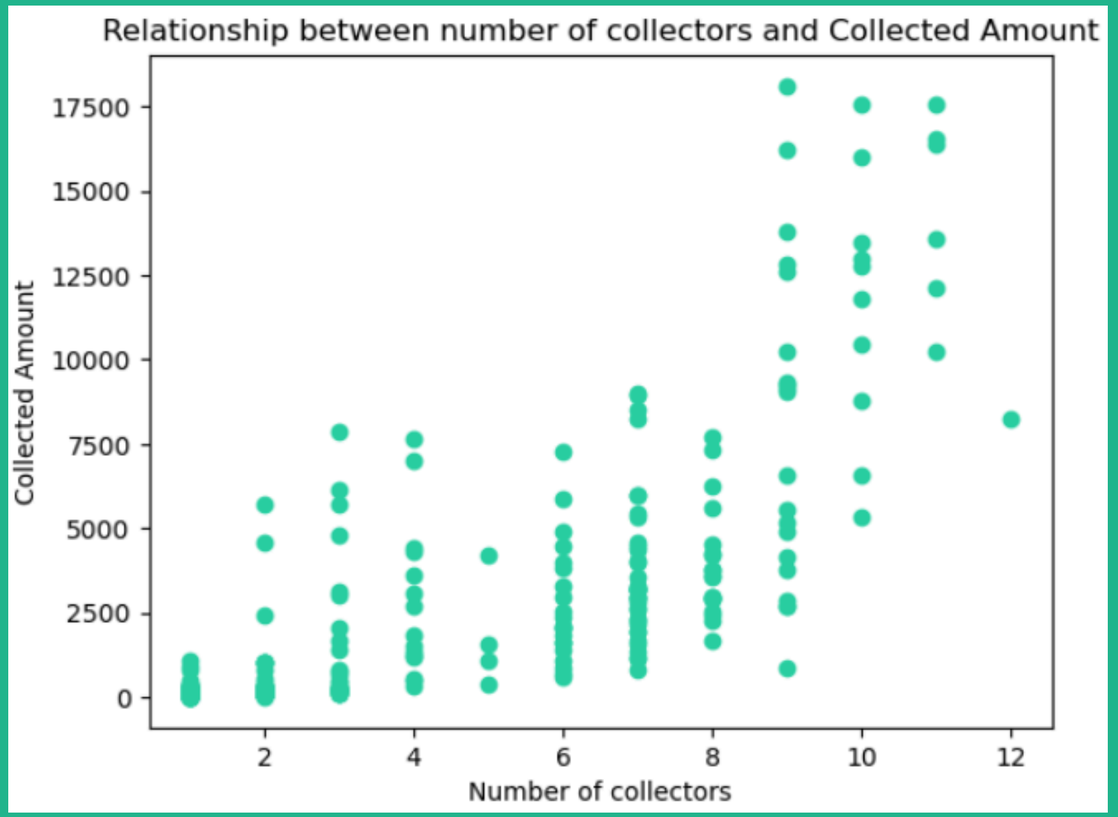
	CollectionAmount	Amount
CollectionAmount	1.000000	0.952352
Amount	0.952352	1.000000

A value of 0.95 indicates that there is a *strong positive linear* relationship between the total daily original claim values and the daily collected amounts from that claim values, such that they move in the same direction (as the amount increases, the collection value also increases).

Collection Amount Analysis

Another factor the amount of collection depends on is the number of collector.

The scatter plot on the right shows that there is a quadratic relationship between the number of collectors and the collection amount.



Takeaways

- Conduct a Data Analysis project
- Try to get more claims with larger amounts
- Build a Dashboard to monitor collection process
- Increase the number of collector to increase the amount of collections

Presented by Ahmad Alqaisi

Thank you very much!

Tuesday, July 9, 2024