Healthcare Partners Inc

Check Amount Predictive Model

The Problem Statement

How can the forecasting team at Healthcare Partners predict Administrative Fees forecast (Check Amount) for the next Fiscal Year, with 80% accuracy for its market types: Food, Services, Surgical and Pharmaceuticals, so that the senior leadership can prepare the company budget to present in 3 months.

The raw data file - CSV

- A data dump from multiple systems.

Cont_Title	Allocation
Market_Type	Accrual_Amount
Contract_Freq	Allocation_Dt_Start
Fiscal_Earned_Year	Allocation_Dt_End
Due_Month	Submission_Id
Due_Year	Submission_Date_Received
Fiscal_Year_Due	Submission_Reported_Adminfees
Check_Amount	Submission_Reported_Salesvolume
Check_Admin_Amount	Category_Desc
Paid_Datekey	Admin_Percent
Fiscal_Year_Received	Cont_Num
Chk_Dt_Deposit	Chk_Owner
Estimate	Previous_Contract_Date

Initial Data exploration

```
]:    df.shape
ut[3]: (720487, 26)
```

Data Wrangling

- Dealing with Missing Values, Redundant Market Types

	count	%
submission_id	710039	98.549870
submission_date_received	710039	98.549870
submission_reported_AdminFees	710039	98.549870
submission_reported_SalesVolume	710039	98.549870
PREVIOUS_CONTRACT_DATE	317865	44.118076
allocation_dt_start	239285	33.211564
allocation_dt_end	239285	33.211564
Check_Amount	233076	32.349786
paid_datekey	233076	32.349786
Fiscal_Year_Received	233076	32.349786
CHK_DT_DEPOSIT	233076	32.349786
CHK_OWNER	233076	32.349786
Check_Admin_Amount	233076	32.349786
ADMIN_PERCENT	1873	0.259963
Contract_freq	1315	0.182515
CATEGORY_DESC	0	0.000000
CONT_NUM	0	0.000000
Cont_title	0	0.000000
Accrual_Amount	0	0.000000
Market_type	0	0.000000
ESTIMATE	0	0.000000
Fiscal_Year_Due	0	0.000000
due_year	0	0.000000
due_month	0	0.000000
Fiscal_earned_year	0	0.000000
ALLOCATION	0	0.000000

```
#Verify the mapping
df[['Market_type','Market_category']].value_counts()
```

```
Market_type                  Market_category
Nursing                      Services                   154994
Rx - Pharmaceuticals         Pharmaceuticals             92786
Surgical PPI                 Services                    78934
Facilities                   Facilities and Material     66286
FS - Food                    Food                        63450
Surgical                     Services                    57398
Imaging                      Services                    46407
Purchased Services           Services                    41182
Laboratory                   Facilities and Material     38321
CV PPI                       Services                    27394
IT/ Telecom                  Facilities and Material     18806
FS - Non-Foods               Food                         9149
Rx - Wholesaler              Pharmaceuticals              9075
Distribution                 Services                     7350
FS - Nutritionals            Food                         6378
PI - PIMS                    Services                      928
MM - Materials Management    Services                      876
PS - Alternate Site          Facilities and Material       311
FS - Chemicals               Food                          293
ARC - Admin Opportunities    Services                      169
dtype: int64
```

Data Wrangling …

- The Tidy data set

```
tidyset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 469981 entries, 0 to 469980
Data columns (total 10 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   Market_type         469981 non-null  object
 1   Contract_freq       469952 non-null  object
 2   Fiscal_earned_year  469981 non-null  int64
 3   due_month           469981 non-null  int64
 4   due_year            469981 non-null  int64
 5   Check_Amount        469981 non-null  float64
 6   CATEGORY_DESC       469981 non-null  object
 7   Market_category     469981 non-null  object
 8   percent_vals        469937 non-null  object
 9   average             469937 non-null  float64
dtypes: float64(2), int64(3), object(5)
memory usage: 35.9+ MB
```

```
tidyset.shape
```

```
(469981, 10)
```

```
tidyset['Market_category'].value_counts()
```

```
Services                   275683
Facilities and Material     75345
Pharmaceuticals             60834
Food                        58119
Name: Market_category, dtype: int64
```

Exploratory Data Analysis - EDA

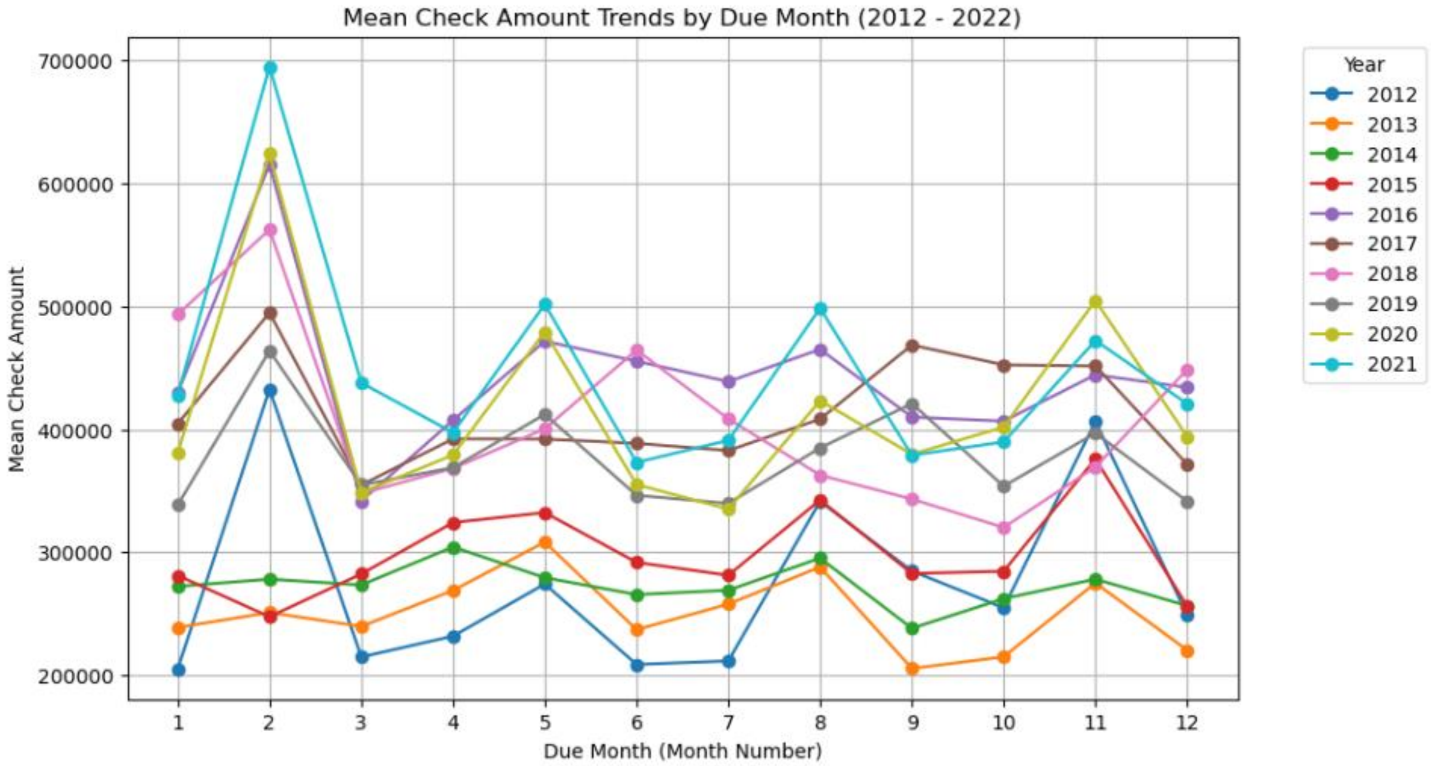
The process to identify distribution of data, interesting patterns and relationships

```
df.describe()
```

	Fiscal_earned_year	due_month	due_year	Check_Amount	average
count	469981.000000	469981.000000	469981.000000	4.699810e+05	469937.000000
mean	2017.432907	6.527130	2017.080503	3.756390e+05	2.669564
std	3.348826	3.385413	3.339391	8.396662e+05	0.829403
min	2003.000000	1.000000	2003.000000	1.000000e-02	0.000000
25%	2015.000000	4.000000	2014.000000	4.333560e+03	2.166667
50%	2018.000000	7.000000	2017.000000	4.283750e+04	3.000000
75%	2020.000000	10.000000	2020.000000	2.854871e+05	3.000000
max	2023.000000	12.000000	2023.000000	1.040222e+07	50.125000

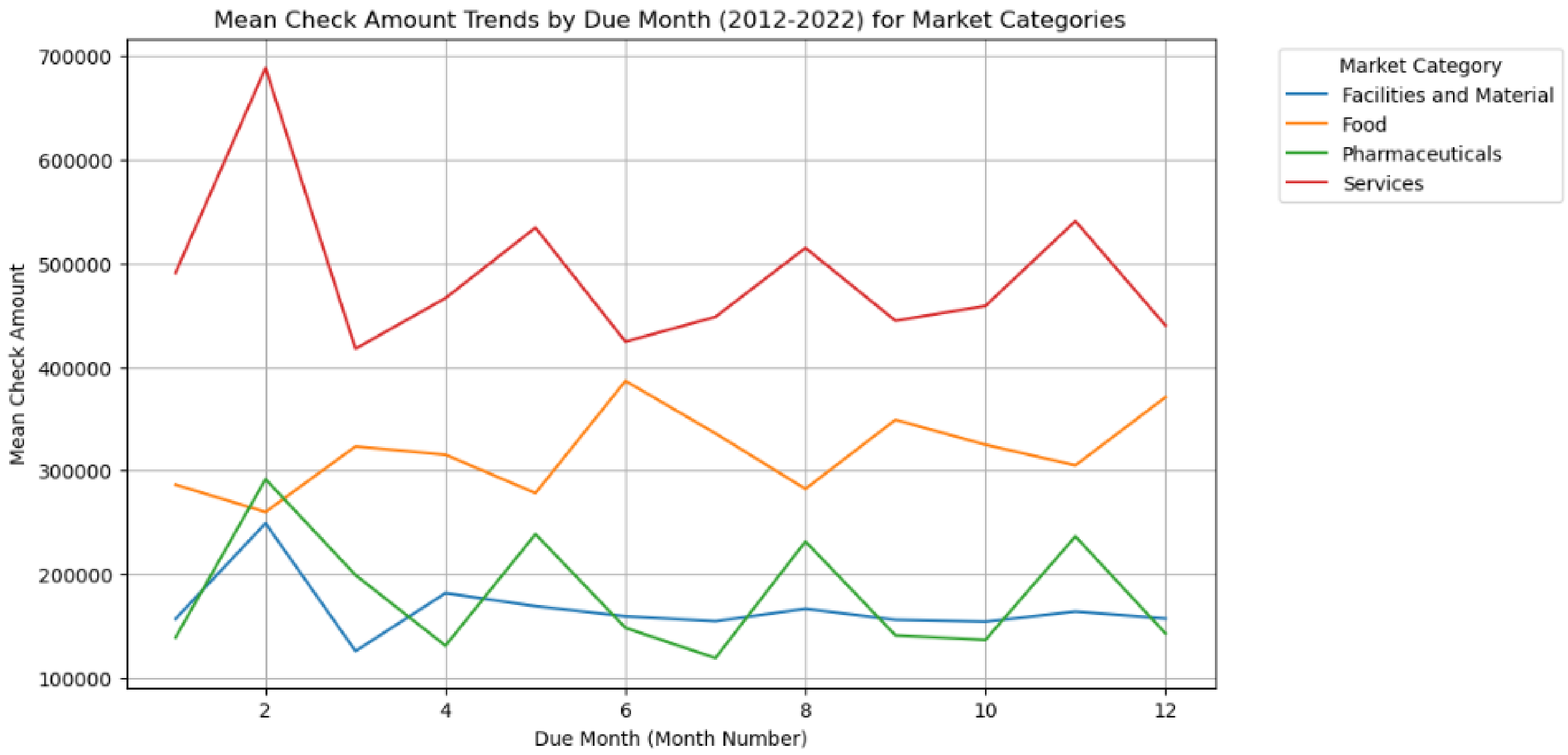
EDA – Explore check amount over the months

due_month	count	mean	std	min	25%	50%	75%	max
1	40770.0	368087.547236	8.475536e+05	0.01	4440.180	38393.150	269093.400	10402223.49
2	20262.0	538780.313036	1.001674e+06	0.01	10132.600	100251.765	558136.060	10402223.49
3	50697.0	339123.331443	7.912393e+05	0.01	3411.230	32170.510	221934.790	10402223.49
4	44889.0	354586.877413	7.793226e+05	0.01	4167.720	37241.010	252359.750	10402223.49
5	43601.0	405626.971865	8.401811e+05	0.01	4526.440	48736.460	330080.840	10402223.49
6	31995.0	351205.058508	8.367194e+05	0.01	4204.030	42492.100	266368.220	10402223.49
7	40279.0	339548.794049	7.798081e+05	0.01	3860.475	34755.750	249318.175	10402223.49
8	44939.0	387962.722329	8.061968e+05	0.01	4628.280	49827.650	320025.380	10402223.49
9	33560.0	359304.664603	8.701706e+05	0.01	3958.965	40570.850	260535.150	10402223.49
10	40886.0	355829.950017	8.274714e+05	0.01	3861.620	36150.110	259233.310	10402223.49
11	44890.0	420196.619363	9.157700e+05	0.01	4795.200	48971.200	320025.380	10402223.49
12	33213.0	361504.696024	8.490653e+05	0.01	4522.910	44503.200	261524.390	10402223.49

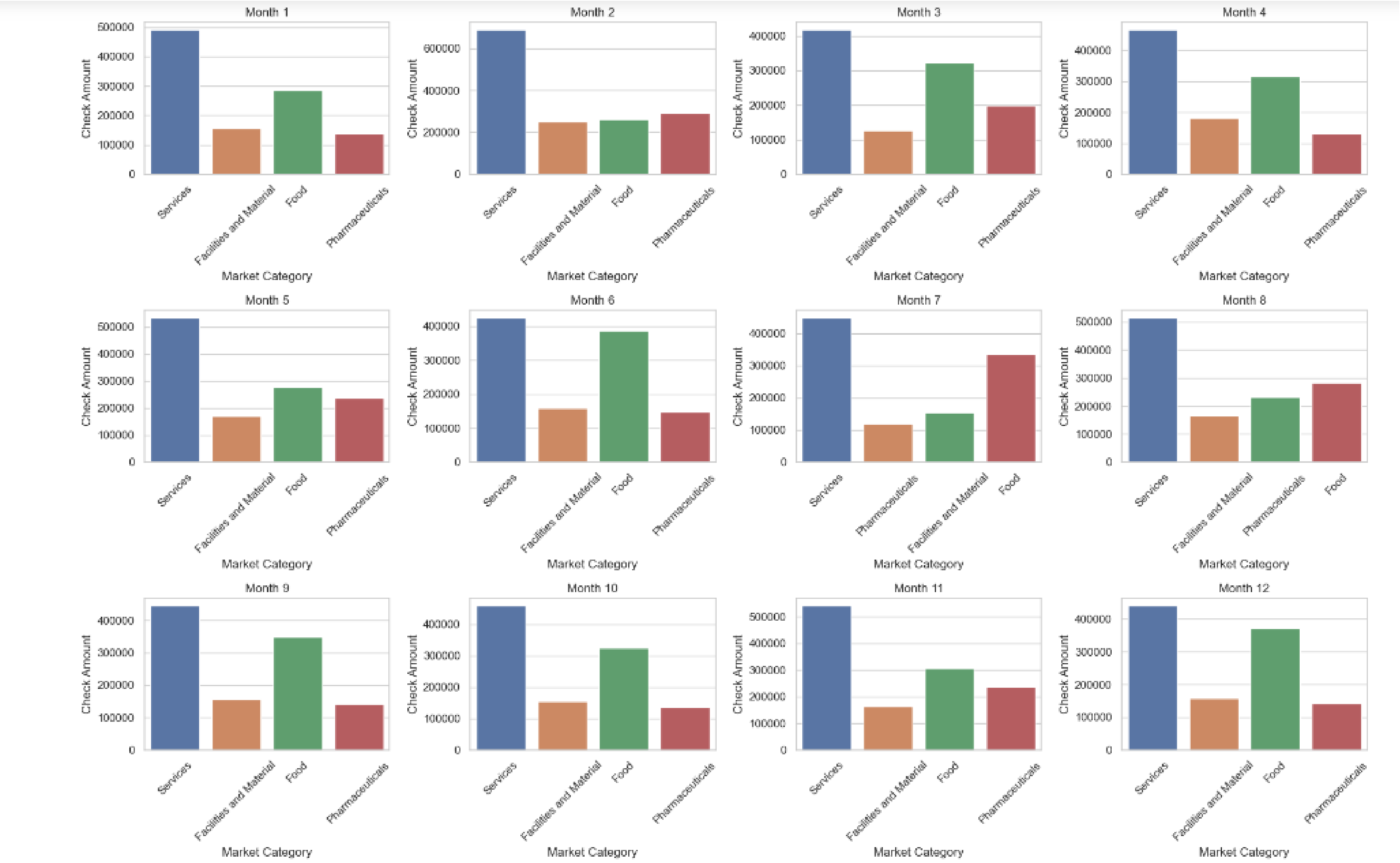


EDA – Explore check amount over different Market Categories

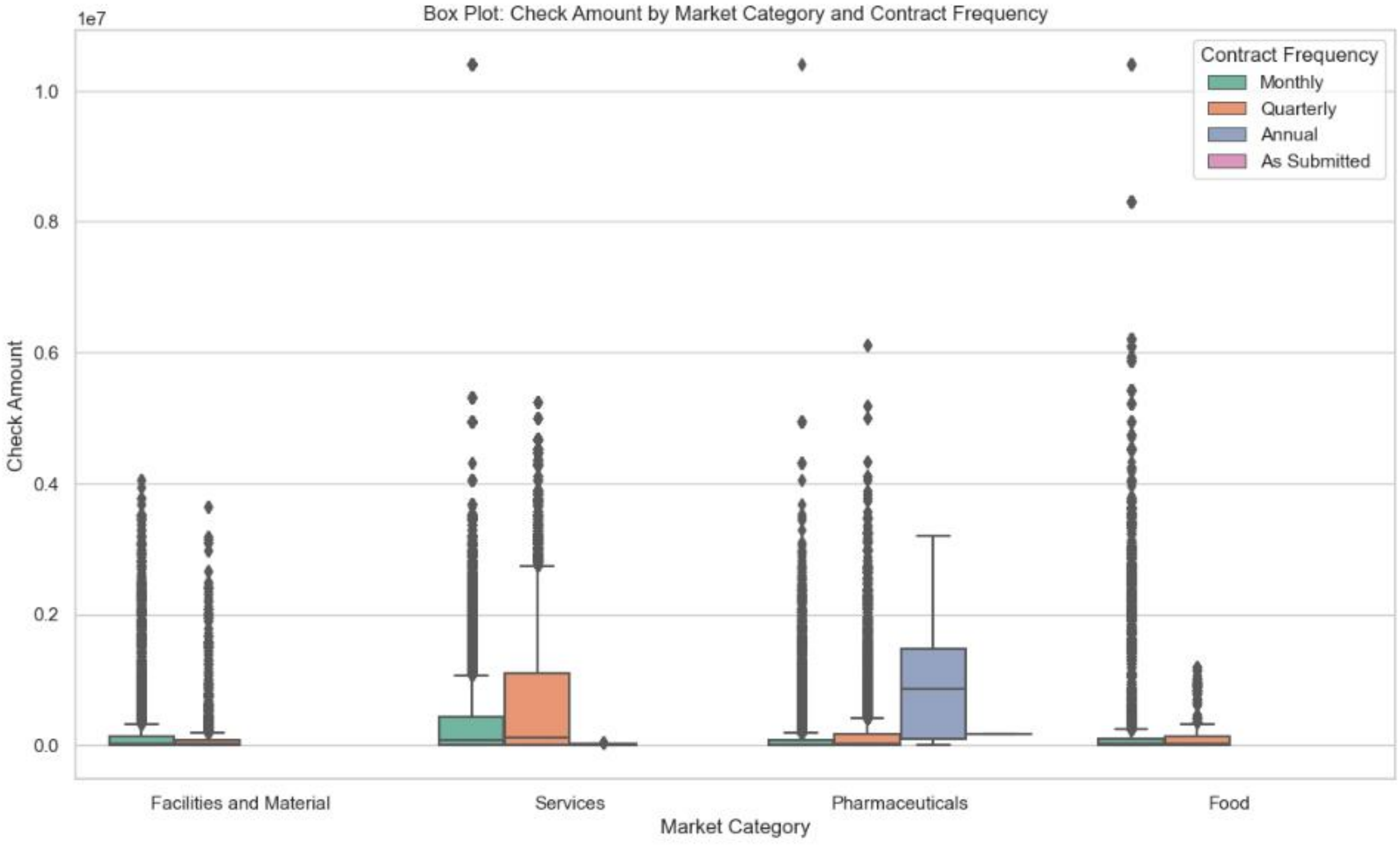
Market_category	count	mean	std	min	25%	50%	75%	max
Facilities and Material	75345.0	161928.971556	383905.206032	0.01	2744.10	20734.46	120187.64	4055237.17
Food	58119.0	334425.961958	928337.729229	0.01	3691.25	25266.35	101703.17	10402223.49
Pharmaceuticals	60834.0	188127.402373	452287.878056	0.01	1974.46	14226.44	113995.83	10402223.49
Services	275683.0	484112.543506	949987.127179	0.01	7031.06	82961.98	478868.47	10402223.49



EDA – Explore Due Month and Market Categories



EDA – Data Distribution



Observations / Findings

- The most influential feature (predictor) is the Market Category. However, based on the feature important analysis, year and contract frequency which is how often the check payments are received seem highly influential.
- Of the various market categories, Services is the biggest contributor.
- The contracts that are set up to pay annually has the highest contribution, when compared to contracts that are setup to pay monthly and quarterly.

Modeling

- Applied 3 different models
  - Linear Regression
  - Random Forest
  - Gradient Boost Regressor

- Comparisons were made after Hyperparameter Tuning

Model Comparison

Standard Models comparison

Linear Regression

Mean Squared Error (MSE):	3.23E+16
Root Mean Squared Error (RMSE):	1.80E+08
Mean Absolute Error (MAE):	1.15E+08
R-squared:	0.45

Random Forest

Mean Squared Error (MSE):	4.36E+15
Root Mean Squared Error (RMSE):	6.60E+07
Mean Absolute Error (MAE):	2.86E+07
R-squared:	0.93

Gradient Boosting Regressor

Mean Squared Error (MSE):	5.54E+15
Root Mean Squared Error (RMSE):	7.44E+07
Mean Absolute Error (MAE):	4.10E+07
R-squared:	0.90

Hyperparameter - Models comparison

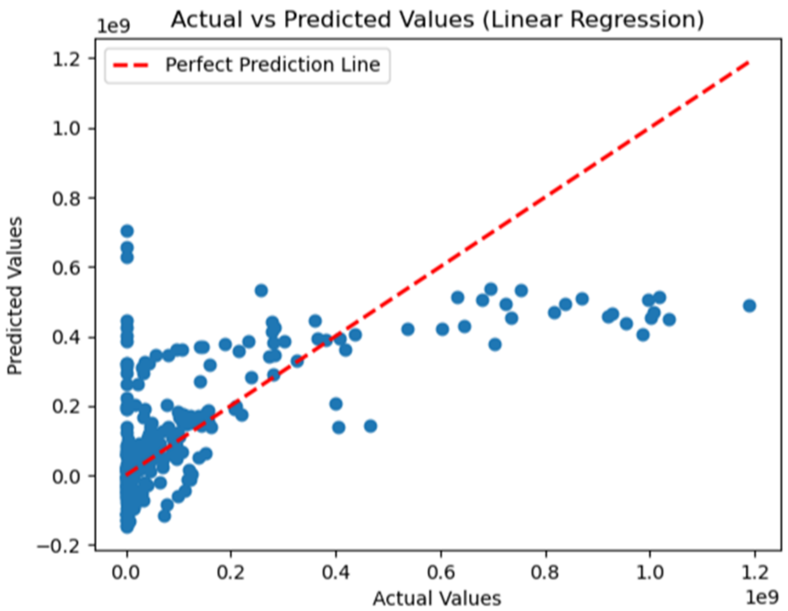
Random Forest

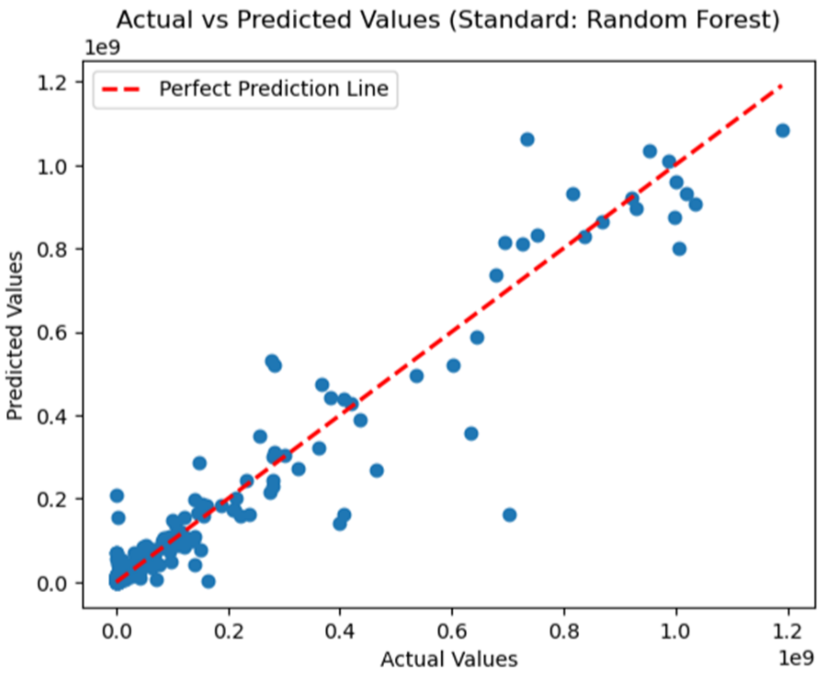
Mean Squared Error (MSE):	5.31E+15
Root Mean Squared Error (RMSE):	7.29E+07
Mean Absolute Error (MAE):	3.17E+07
R-squared:	0.91

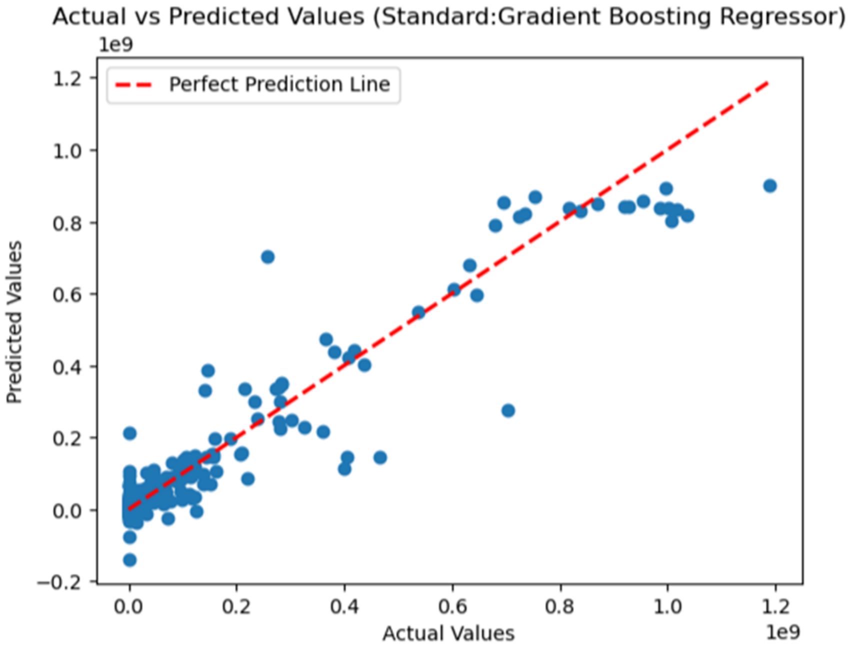
Gradient Boost

Mean Squared Error (MSE):	4.12E+15
Root Mean Squared Error (RMSE):	6.42E+07
Mean Absolute Error (MAE):	2.64E+07
R-squared:	0.93

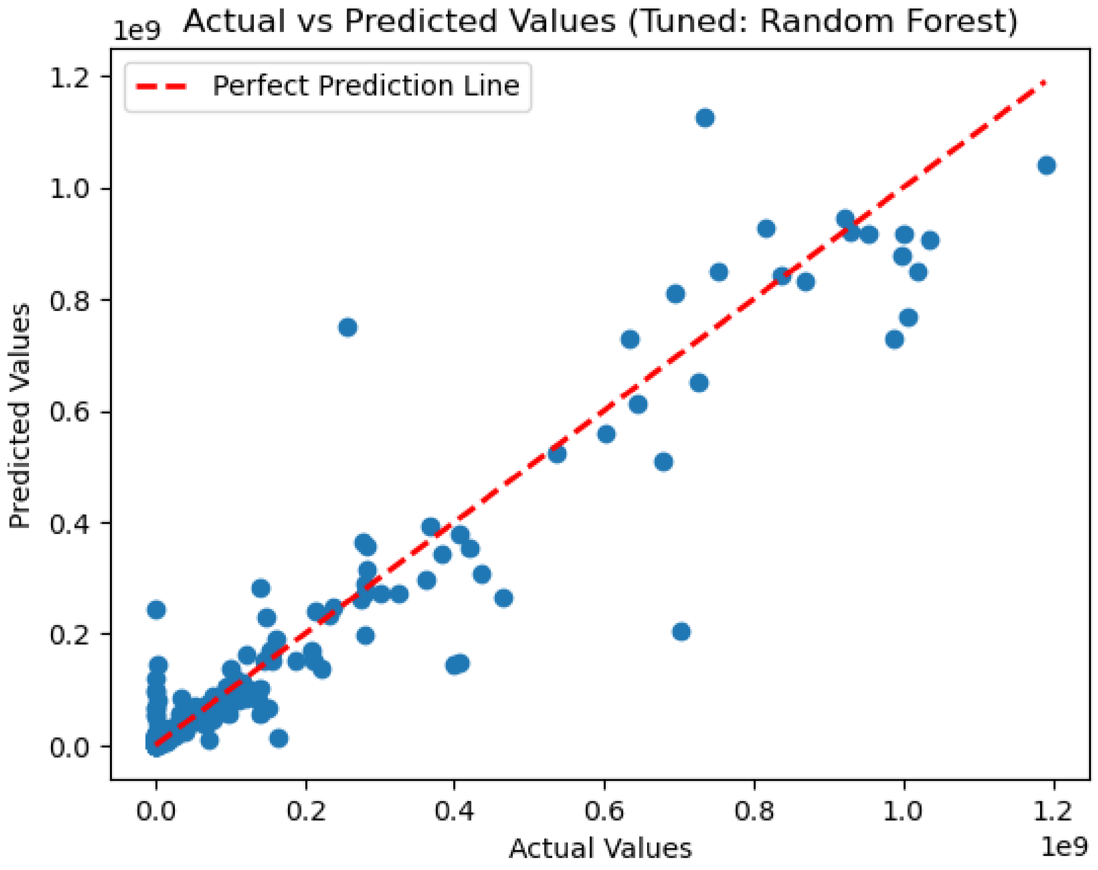
Model Performance – before Hyperparameter Tuning

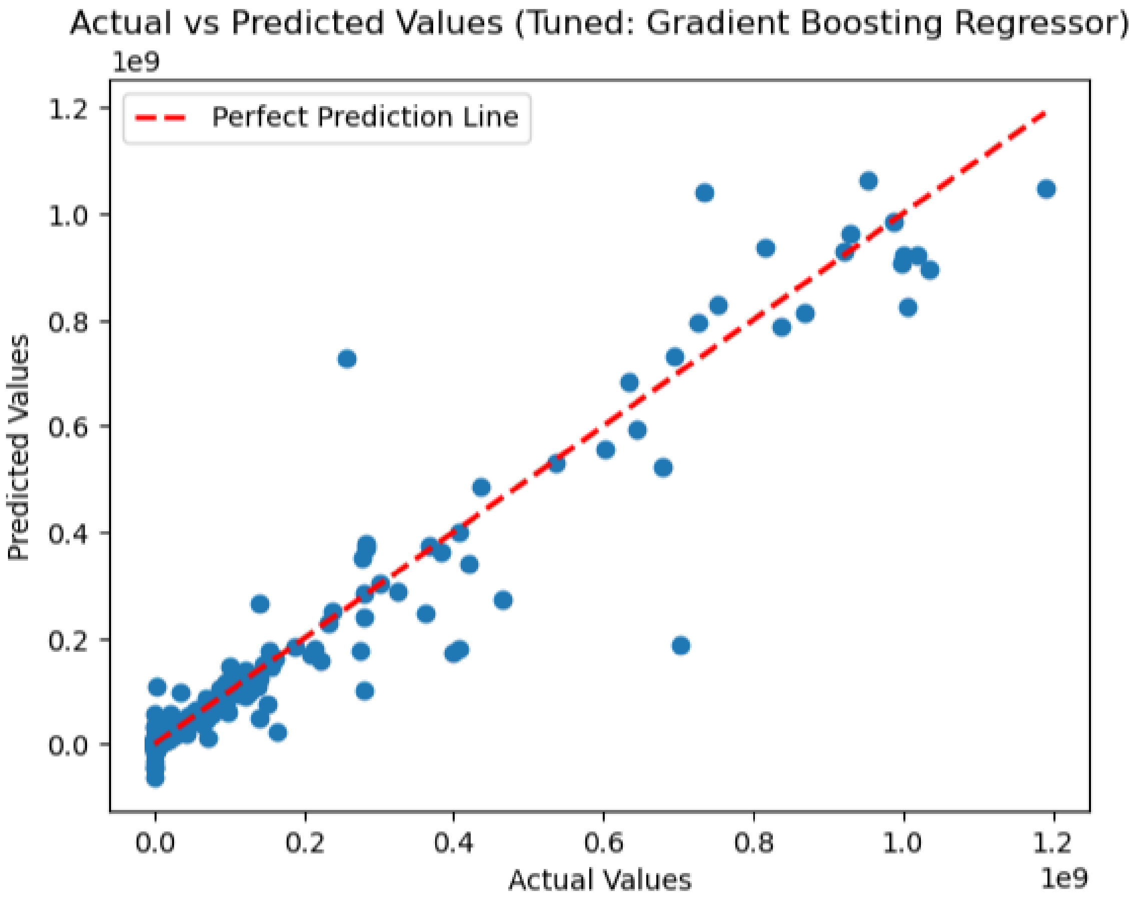






Model Performance – After Hyperparameter Tuning





Future Work

- The model performance in my opinion is moderate. I believe the performance can be improved and is proposed as future work.
- For the lack of experience and understanding I realized feature engineering was not successfully implemented when training the models.
- As future work I suggest feature engineering specifically scaling and imputation to better handle the huge variances that are shown as outliers.
- Additionally apply binning on the check amount to see Better and more feature engineering techniques – imputation, scaling, binning and bucketing