# Fraudulent Credit Card charges

## 1. Introduction

### a) Problem statement

Credit card fraud is a significant and growing problem in the financial industry, causing substantial financial losses and undermining consumer trust. Fraudulent activities include unauthorized transactions, identity theft, and the misuse of stolen credit card information. As technology advances and online transactions become more prevalent, the methods and sophistication of fraudsters continue to evolve, making it increasingly challenging to detect and prevent fraudulent activities.


Figure 1: The rise of card fraud worldwide throughout the years and in the future years

### b) Goal

By using a test dataset of credit card transactions that was determined as fraudulent or not fraudulent, I created a model that best help predicts the attributes that best determines whether a credit card transaction is fraudulent

## 2. Dataset

### a) Fraudulent credit card dataset

**(https://www.kaggle.com/datasets/kartik2112/fraud-detection)**

The fraudulent detection dataset was retrieved from Kaggle website. The credit card transaction history comes in a csv format and is read as through fraudTest.csv. It is important to know that

There are 23 attributes which include but are not limited to:

- Merchant longitude and latitude
- Transaction category type
- Customer's long and latitude
- Gender
- Time of transaction
- CC number
- Date
- Address

## 3. Data Cleaning and Data Wrangling

The raw dataset contained 23 different columns with 1000 transactions from a pool of 800 merchants ranging from 1/1/2019 – 12/31/2020. With the 23 parameters to consider, I narrowed the attributes to help determine which parameters would best help us determine credit card fraudulent transactions. While the dataset presented the latitude and longitude of the merchant and the customer's address, I performed some data wrangling by finding the distance between the customer's address and the merchant's address. This would help determine if a transaction is fraudulent because the transactions may exist in a very distant place from the customer's primary residence.

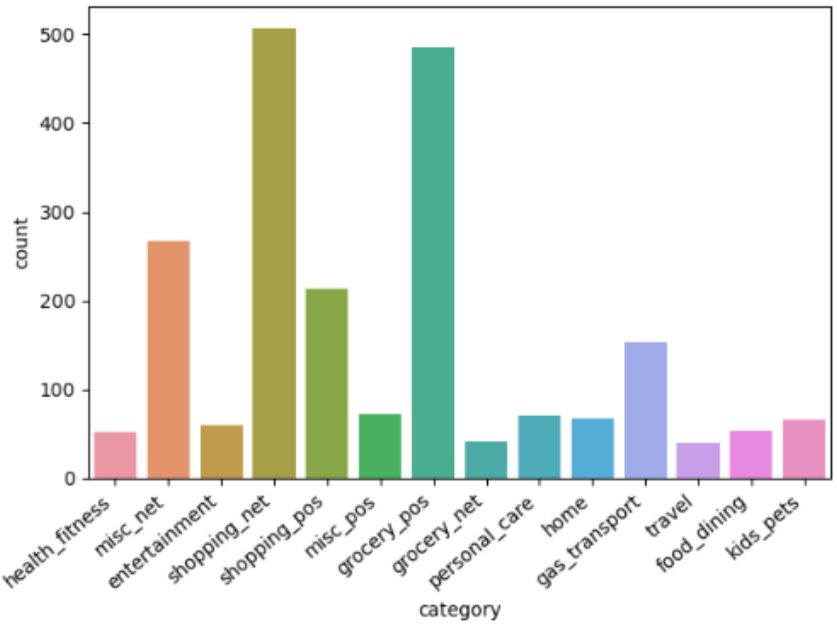
Based on the analysis and professional knowledge in the financial industry, I determined that the best parameters that helps determine fraudulent transactions are the following:

- Distance
- Age
- Transaction Category Type

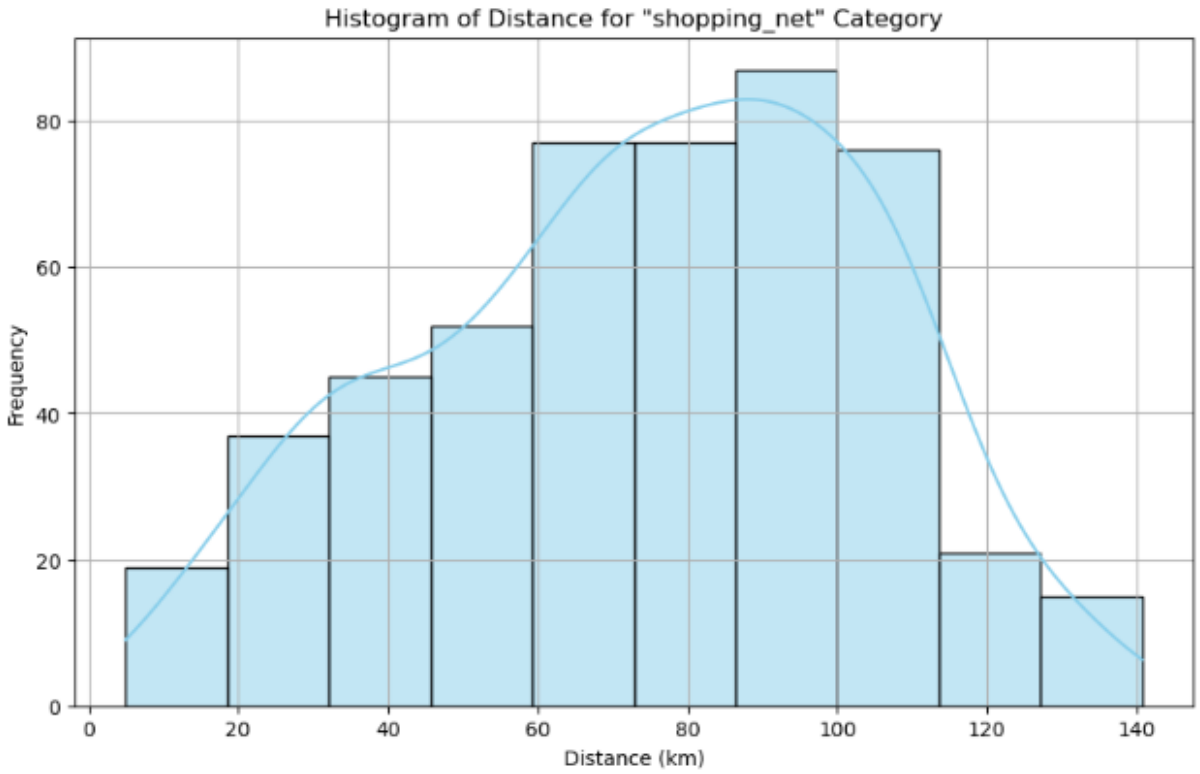
## 4. Exploratory Data Analysis and Initial Findings

In the initial exploratory data analysis and initial findings, I was able to identify, without filters, that categories are typically evenly distributed except for travel, misc_net, and grocery_net. When filtering for credit card transactions that have been identified truly for fraud, I was able to identify that misc_net, shopping_net, shopping_pos, grocery_pos, and gas_transport were typically the highest categories where transactions were identified as fraud.

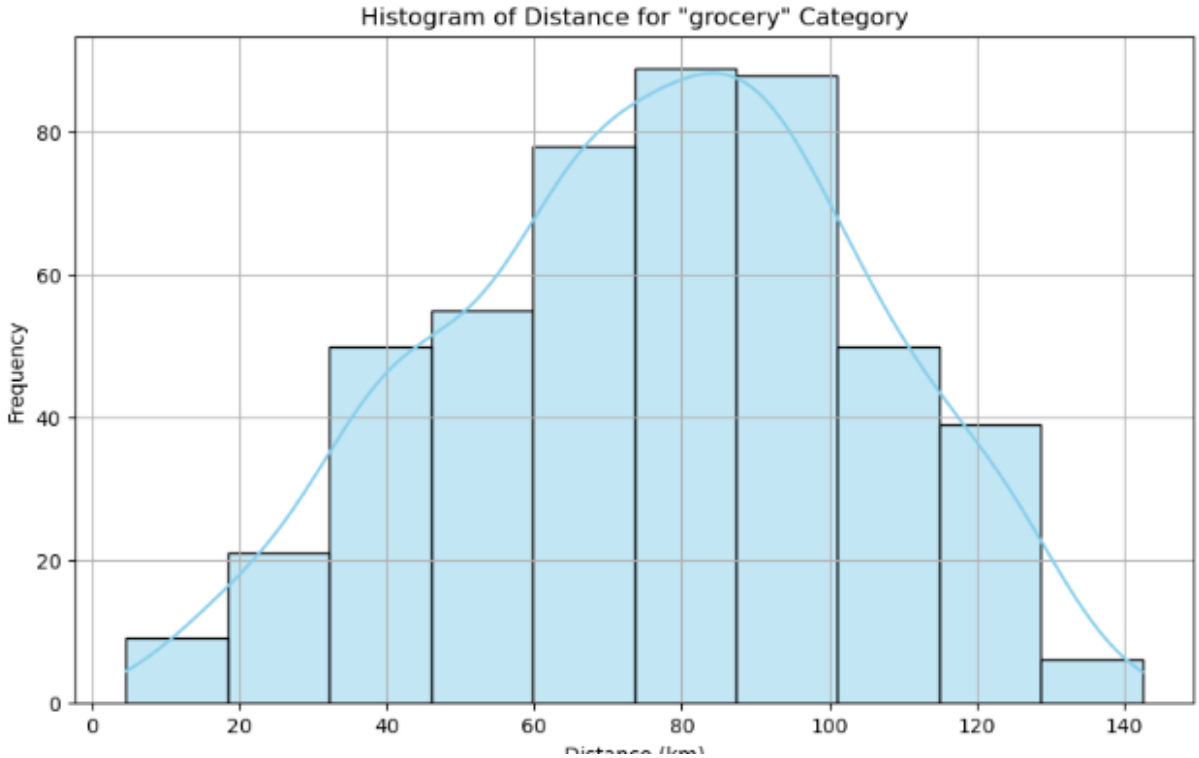
In [24]:
```
# % of fraud
fraud_df = df[df['is_fraud']==1]
fraud_df.dtypes
fraud_df['category']
ax1 = sns.countplot(data = fraud_df, x= 'category')
ax1.set_xticklabels(ax1.get_xticklabels(), rotation=40, ha="right")
plt.tight_layout()
plt.show()
```




In creating a histogram for misc_net, shopping_net, and grocer_pos, I was able to identify that the majority of transactions typically happened at least 40KM for the grocery, miscellaneous, and the online shopping categories.




Histogram of distances for identified fraud transactions for “shopping_net” category




Histogram of distances for identified fraud transactions for “grocery” category

## 5. Model Selection

For each of the categories, I used the following different learning classification models:

- Feature Scaling
- Logistic regression
- K Nearest Neighbor
- Support Vector Machine
- Random Forest
- Gradient Boosting
- Naïve Bayes

When it came to the models, the Machine Learning that helps best determine credit card fraud were the following:

- Random Forest
- Gradient Boost
- SVM (Support Vector Machine)

In [66]:
```
myLabels = [ 'Logistic Regression','KNN','SVM','Random Forest','Gradient Boost', 'Naive Bayes']
score_test= [  cv_scores_lr_test,cv_scores_knn_test,cv_scores_svm_test,cv_scores_rf_test,cv_scores_gbc_
score_train= [  cv_scores_lr_train,cv_scores_knn_train,cv_scores_svm_train,cv_scores_rf_train,cv_scores
Accuracy_score = [Accuracy_lr,Accuracy_knn,Accuracy_svm,Accuracy_rf,Accuracy_gbc,Accuracy_nb]

score_tab_acc = pd.DataFrame(list(zip(myLabels, Accuracy_score)),
               columns =['Algorithm', 'Model accuracy score'])

score_tab = pd.DataFrame(list(zip(myLabels, score_train, score_test)),
               columns =['Algorithm', 'ROC-AUC train score', 'ROC-AUC test score' ])
print(score_tab_acc)

score_tab
```

```
              Algorithm  Model accuracy score
0  Logistic Regression              0.995849
1                   KNN              0.996155
2                   SVM              0.996161
3         Random Forest              0.998093
4        Gradient Boost              0.997211
5           Naive Bayes              0.992268
```

Out[66]:

| | Algorithm | ROC-AUC train score | ROC-AUC test score |
|---|---|---|---|
| 0 | Logistic Regression | 0.827547 | 0.813130 |
| 1 | KNN | 0.682084 | 0.634340 |
| 2 | SVM | 0.639644 | 0.692787 |
| 3 | Random Forest | 0.956986 | 0.951107 |
| 4 | Gradient Boost | 0.968626 | 0.952159 |
| 5 | Naive Bayes | 0.842050 | 0.842580 |

Based on determining the best model to identify credit card fraud, it is important to know that all models are good at identifying fraud. However, when using the ROC-AUC score, it is important to know that the ROC-AUC best determined that the random forest and the gradient boost model algorithm best identifies credit card fraudulent activities.

In addition to determining the best model that can best identify transactions as fraudulent, I performed a feature analysis that best determines the attributes to help identify credit card fraud. Based on the analysis performed, it was determined that age and distance are well suited for identifying fraud.

```
In [72]: #Feature importances
         importances = list(rf.feature_importances_)
         imp=np.sort(importances)
         tab=pd.DataFrame(list(zip(X,imp)),columns =['Features', 'Importance scores'])
         print(tab)
```

```
          Features  Importance scores
0           cc_num           0.046497
1              amt           0.050527
2              lat           0.054020
3             long           0.054505
4  category_label           0.071652
5              age           0.162832
6         distance           0.559967
```

## 6. Future Work

This project allowed me to develop a ML model that helps best identify for customer transactions that are determined as fraudulent. While my models are valuable for this project, it is difficult to determine across the financial industry because there are other factors and variables that financial institutions may use. Because the data was limited to various categories and details, having more details such as FICO scores, Underwriting strategy, historical trends on the customer, # of fraudulent transactions historically from the start of credit transaction origination process.