CREDIT CARD FRAUD DETECTION

Shruti Agarwal, Zihao Zhou, Zora Mardjoko

Motivation

$246,000,000

was lost due to credit card fraud in 2023.

Objective and Value Proposition

- Developing a predictive model to determine whether a transaction is fraudulent or not.
  - Flag fraudulent transactions and block them!
- Understanding which features play a significant role in predicting fraudulent transactions.

Exploratory Data Analysis

Unnamed: 0	trans_date_trans_time	cc_num	merchant	category	amt	first	last	gender	street	merch_lat	merch_long	is_fraud
0	21/06/2020 12:14	2.291160e+15	fraud_Kirlin and Sons	personal_care	2.86	Jeff	Elliott	M	351 Darlene Green	33.986391	-81.200714	0
1	21/06/2020 12:14	3.573030e+15	fraud_Sporer-Keebler	personal_care	29.84	Joanne	Williams	F	3638 Marsh Union	39.450498	-109.960431	0
2	21/06/2020 12:14	3.598220e+15	fraud_Swaniawski, Nitzsche and Welch	health_fitness	41.28	Ashley	Lopez	F	9333 Valentine Point	40.495810	-74.196111	0
3	21/06/2020 12:15	3.591920e+15	fraud_Haley Group	misc_pos	60.05	Brian	Williams	M	32941 Krystal Mill Apt. 552	28.812398	-80.883061	0

Dataset

#	Column	Non-Null Count	Dtype
0	Unnamed: 0	555719 non-null	int64
1	trans_date_trans_time	555719 non-null	object
2	cc_num	555719 non-null	float64
3	merchant	555719 non-null	object
4	category	555719 non-null	object
5	amt	555719 non-null	float64
6	first	555719 non-null	object
7	last	555719 non-null	object
8	gender	555719 non-null	object
9	street	555719 non-null	object
10	city	555719 non-null	object
11	state	555719 non-null	object
12	zip	555719 non-null	int64
13	lat	555719 non-null	float64
14	long	555719 non-null	float64
15	city_pop	555719 non-null	int64
16	job	555719 non-null	object
17	dob	555719 non-null	object
18	trans_num	555719 non-null	object
19	unix_time	555719 non-null	int64
20	merch_lat	555719 non-null	float64
21	merch_long	555719 non-null	float64
22	is_fraud	555719 non-null	int64

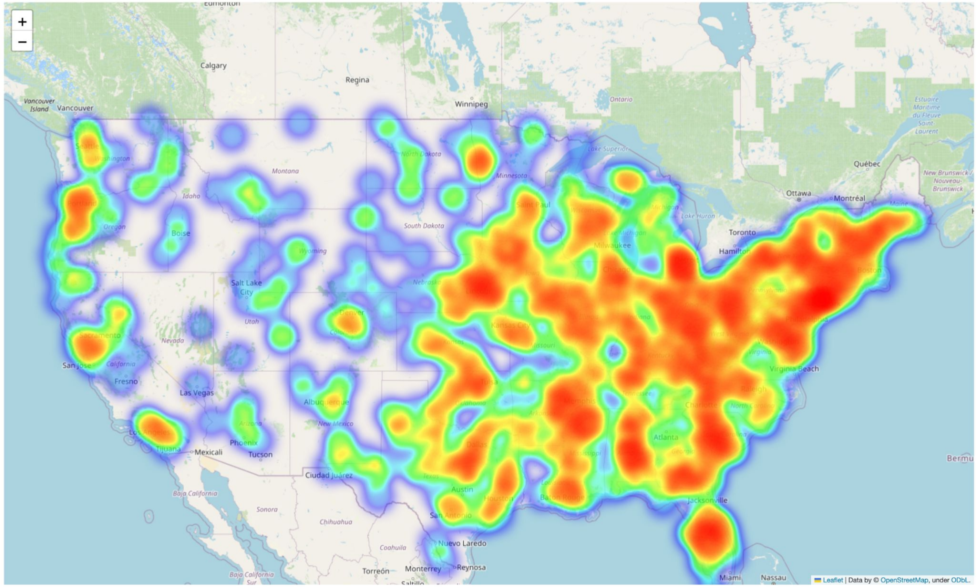
columns (total 14 columns):

	Column	Non-Null Count	Dtype
	merchant	555719 non-null	object
1	category	555719 non-null	object
2	amt	555719 non-null	float64
3	gender	555719 non-null	int64
4	city_pop	555719 non-null	int64
5	is_fraud	555719 non-null	int64
6	full_name	555719 non-null	object
7	job_category	555719 non-null	object
8	month	555719 non-null	int32
9	day_of_week	555719 non-null	int32
10	time_of_day	555719 non-null	int64
11	distance_between	555719 non-null	float64
12	age	555719 non-null	int64
13	division	555719 non-null	object

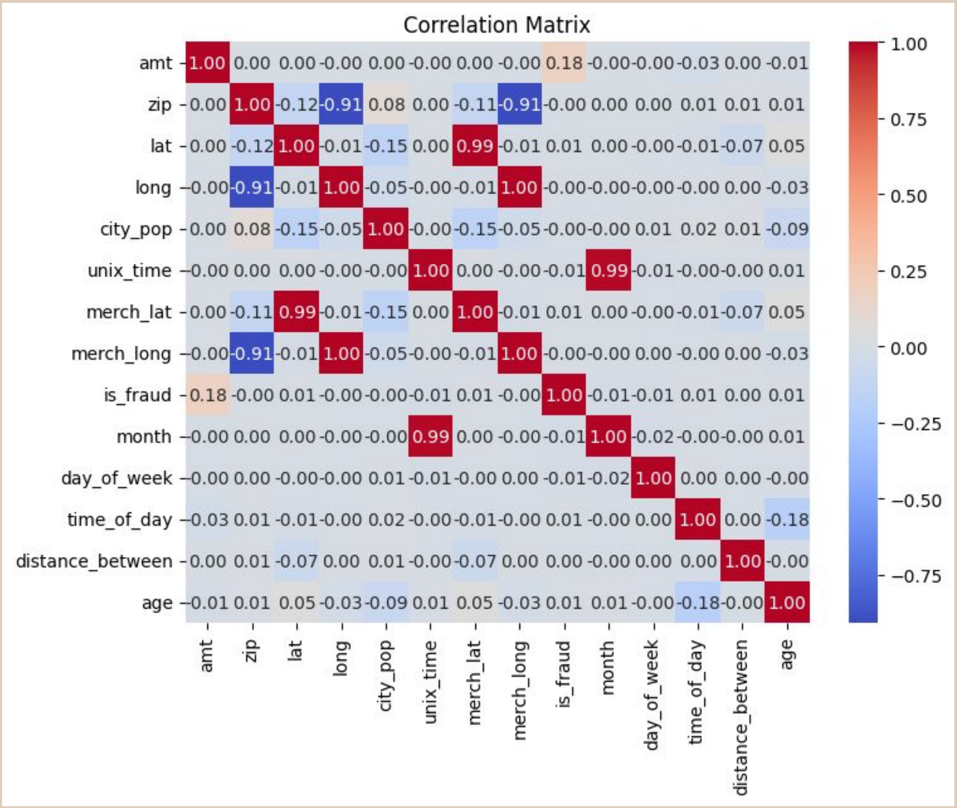
Feature Engineering

1. **unix_time** → time_of_day, month, day_of_week
2. **merch_lat, merch_long, lat, long** → distance_between
3. **dob** → age
4. **state** → division

EDA: Graphed



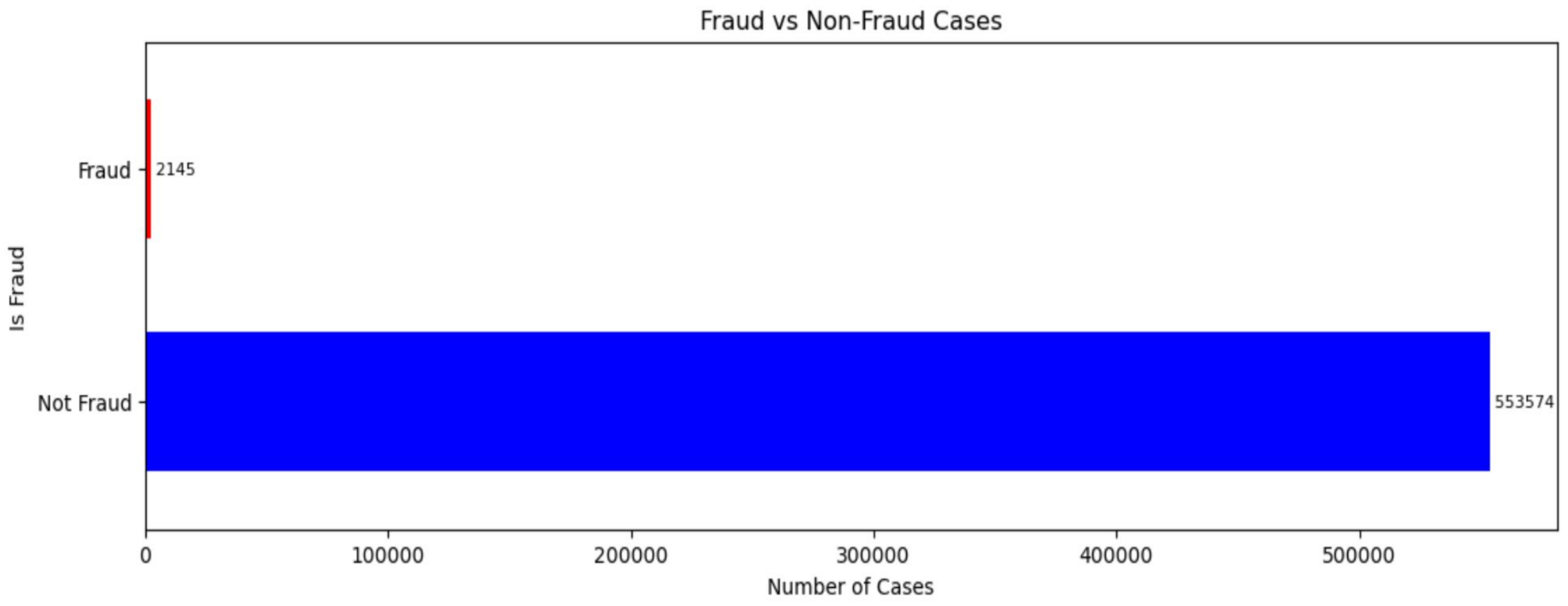
- Transactions were dispersed throughout
- Divided location (long/lat) into geographic “zones”
- Calculated distance between for relative measure

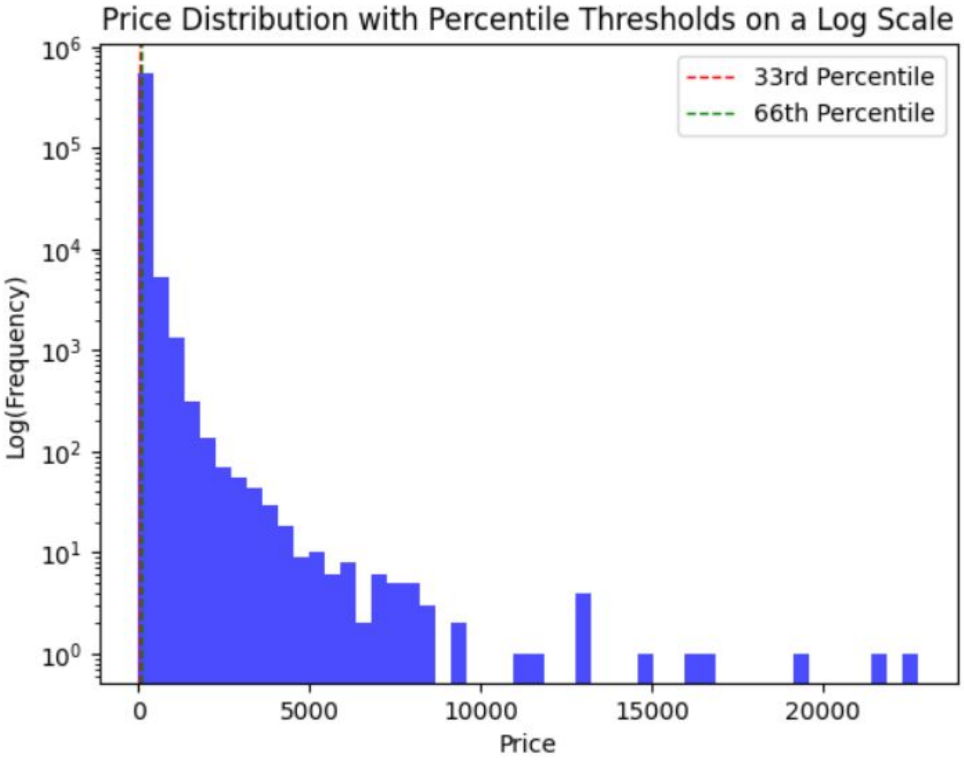


What We Learned from EDA: Correlation Mx

- Correlated features: zip, lat, long, merch_lat, merch_long
- Computed distance_between and dropped correlated features

What We Learned from EDA: Categories





What We Learned from EDA: Price

- Skewed price distribution

Modeling

Method 1: Vanilla Logistic Regression

Metrics:

```
Logistic Regression Metrics
Accuracy: 0.9953303759047397
Precision: 0.07142857142857142
Recall: 0.0078125
F1 Score: 0.014084507042253521
Train precision: 0.16666666666666666
Train recall: 0.01556420233463035
```

Takeaways:

High accuracy, low precision/recall
→ **class imbalance**

High training/testing error
→ **underfitting**

Logistic Regression Tuning

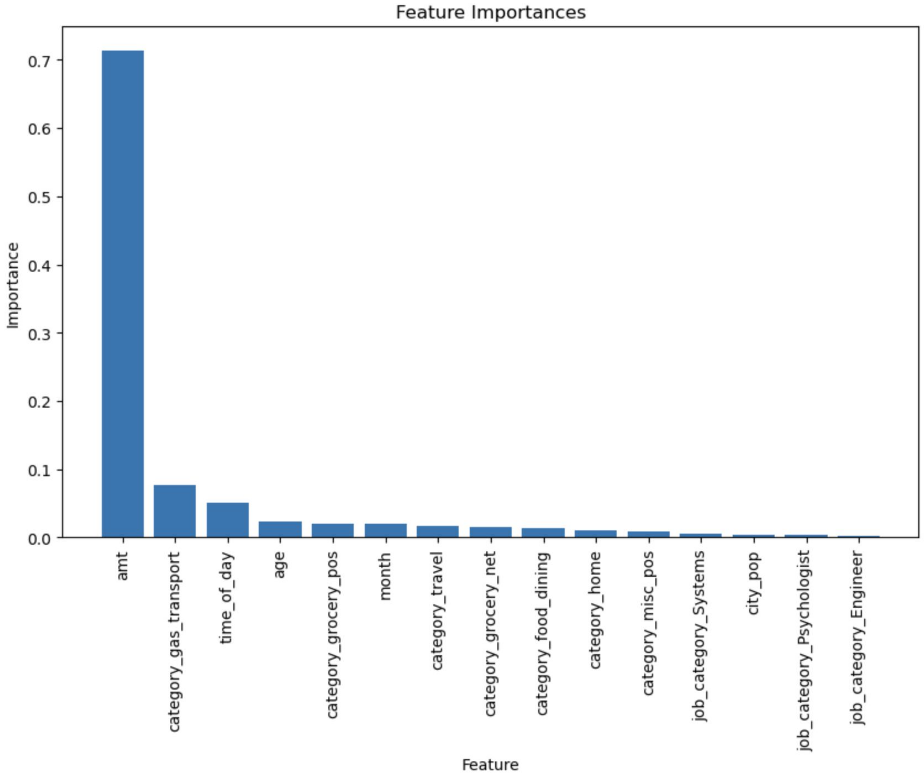
SMOTE-d on Normalized

Address class imbalance!

Recall → out of all true fraudulent, % correct prediction

Precision → out of fraudulent predictions, % correct prediction

```
Logistic Regression Metrics
Accuracy: 0.896367699543044
Precision: 0.031456432840515886
Recall: 0.78125
F1 Score: 0.06047777441790142
```



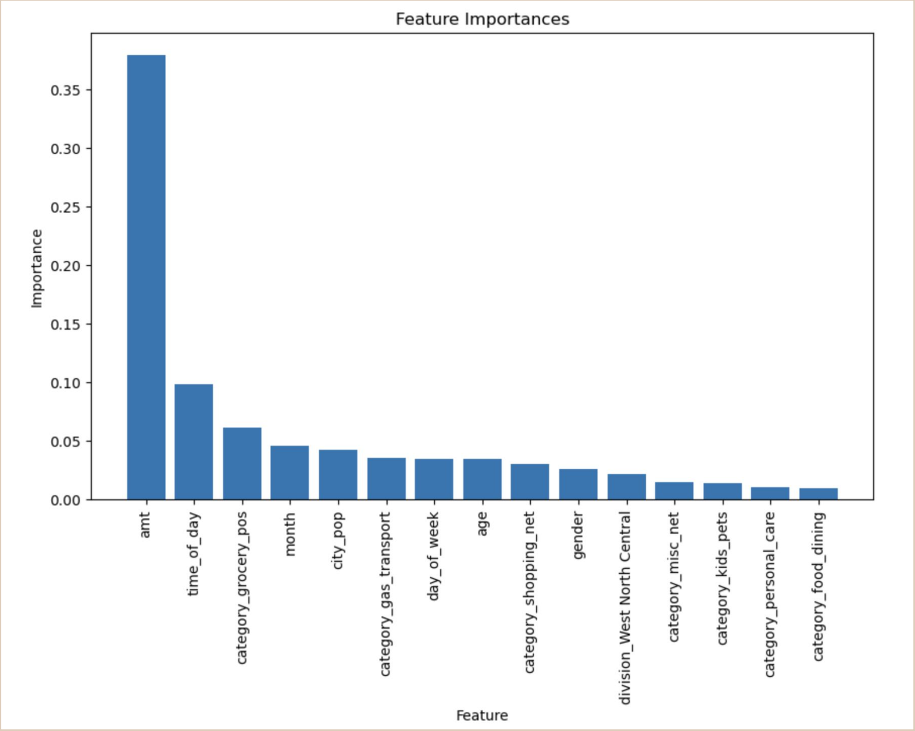
Method 2: Decision Tree Classification

```
Precision: 0.1601796407185629
Recall: 0.8359375
Accuracy: 0.9805877055468464
F1-score: 0.2688442211055276
```

Performed GridSearch + 3-Fold Cross Validation.

Optimal parameters:
max_depth = 10

Method 3: Random Forest Classification




```
Precision: 0.9375
Recall: 0.703125
Accuracy: 0.9985324038557754
F1-score: 0.8035714285714286
```

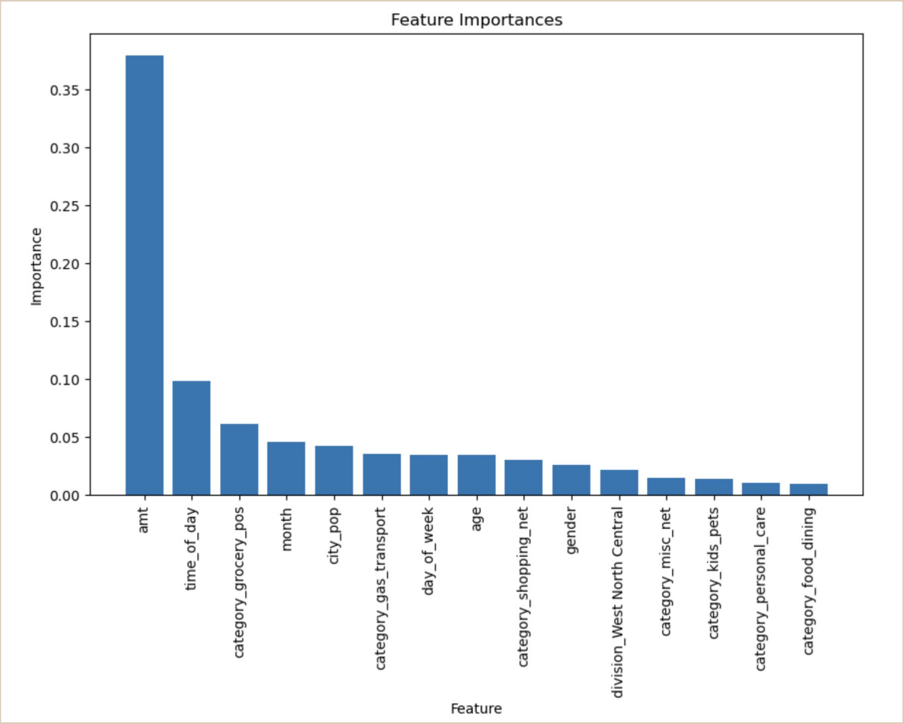
Performed GridSearch + 2-Fold Cross Validation.

Optimal parameters:
max_depth = 30
n_estimators = 35

Conclusion

Best Overall Model:

Random Forest Classifier



Random Forest Classification

```
Precision: 0.9375
Recall: 0.703125
Accuracy: 0.9985324038557754
F1-score: 0.8035714285714286
```

- **Highest precision** (0.94 > 0.17)
- **Improved recall score** (low number of false negatives)
- **Most even distribution of weights on features**

Limitations and Areas to Explore

- Limitations
  - Model results display tradeoff between high precision and high recall
  - Quantity of one hot-encoded data (at what point does it become “too much” data?)

- In the future, it would be interesting to consider implementing the following modifications:
  - Complex feature engineering (i.e. repeated transactions)
  - Job categories into sectors to reduce blow up