# Informe de proceso de desarrollo de modelo de predicción de fraude

El desarrollo del modelo de detección de fraude inicia con la caracterización del dataset, donde se identifica la existencia de varias variables categóricas y se realiza un agrupamiento de las variables según la entidad que representan. Se determinan tres entidades que agrupan las variables:

- Cliente
- Comercio que solicita la transacción
- Transacción

Se inicia con un filtrado heurístico donde se busca eliminar variables redundantes o de poca relación con la determinación de fraude. Se identifican 5 variables en esta etapa:

- Nombre y apellido del cliente
- Dirección y código postal del cliente
- Id de la transacción

Para diseñar y evaluar modelos predictivos, independiente del tipo de algoritmo que se emplee, es deseable tener la menor cantidad posible de variables categóricas no numéricas. De modo que se inicia un acondicionamiento del dataset para convertir la mayor cantidad posible de variables en numéricos.

Se inicia con la conversión de string a número de las variables de **monto de transacción** y ubicación geográfica tanto del cliente como del comercio. Estas 5 variables se encontraban registradas como string y separaban los decimales con carácter coma ",". De modo que se realiza el acondicionamiento correspondiente para obtener los valores en formato de punto flotante.

Las variables de ciudad y estado se binarizan empleado listas de los códigos FIPS y un listado de ciudades a las que se les asigna un código no estandarizado internacionalmente, pero que permite traducir los nombres en variables categóricas. Los ficheros originales se encuentran disponibles públicamente en mi repositorio de GitHub[1] [2]

De manera similar, se realiza la binarización de la variable **category,** con la diferencia que el código numérico equivalente se obtiene de listar todos los valores registrados de la variables, dando un total de 14 categorías. La variable **Merchant** se descarta por dos razones:

- Cada Merchant tiene asociado una categoría
- El 97.97% de los **merchant** tienen asociado al menos un evento de fraude, de modo que no resulta en una variable diferenciadora

[1] https://github.com/Dialvec/USA_Locations_Codes/raw/main/US_States_FIPS_Code.csv
[2] https://github.com/Dialvec/USA_Locations_Codes/raw/main/USA_cities.csv

Diego A. Velandia C.
C.C. 1019054912

Para binarizar la variable **category** se asigna un valor entero en función del ordenamiento de estas categorías en función del conteo de fraudes registrados, como se muestra en la figura 1. A la variable de mayor conteo de fraudes se le asigna el valor 1, a la segunda categoría en orden se le asigna el valor 2, y así sucesivamente. Esta tabla puede actualizarse periódicamente para reorganizar la prioridad de cada categoría.

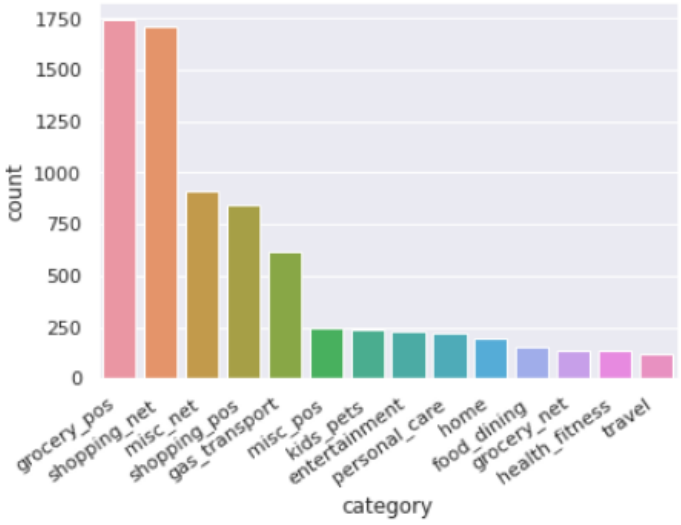


*Figura 1. Clasificación de variable category por conteo de fraudes*

Al estudiar la variable **job** del cliente, se encuentra que el 89.67% de los Jobs existentes en el dataset tienen asociado al menos un evento de fraude, de modo que se descarta la variable.

Finalmente, las variables de estampa de tiempo de la transacción y fecha de nacimiento del cliente se descomponen y se extraen los siguientes valores almacenados en columnas (variables individuales):

- Mes de la transacción
- Día de la transacción
- Hora de la transacción
- Año de nacimiento del cliente

Posterior a estos procesos se tienen todas las variables en formato numérico, de modo que se realiza una evaluación del nivel de importancia de las variables en la predicción de la variable de fraudes detectados. El análisis se realiza desde dos perspectivas:

- Clasificador con f-score para la suposición de que existe una relación lineal entre los *features* y la variable a predecir
- Clasificador SVM (Support Vector Machine) que no requiere relaciones lineales

Diego A. Velandia C.
C.C. 1019054912

Se encuentra que no existen relaciones lineales entre los features y la variable de fraude. Sin embargo, se identifica que la gran mayoría la mayor parte de la importancia de los features se concentra en pocas variables. La figura 2 y la tabla 1 muestran la clasificación de los 17 features para un modelo de SVM.

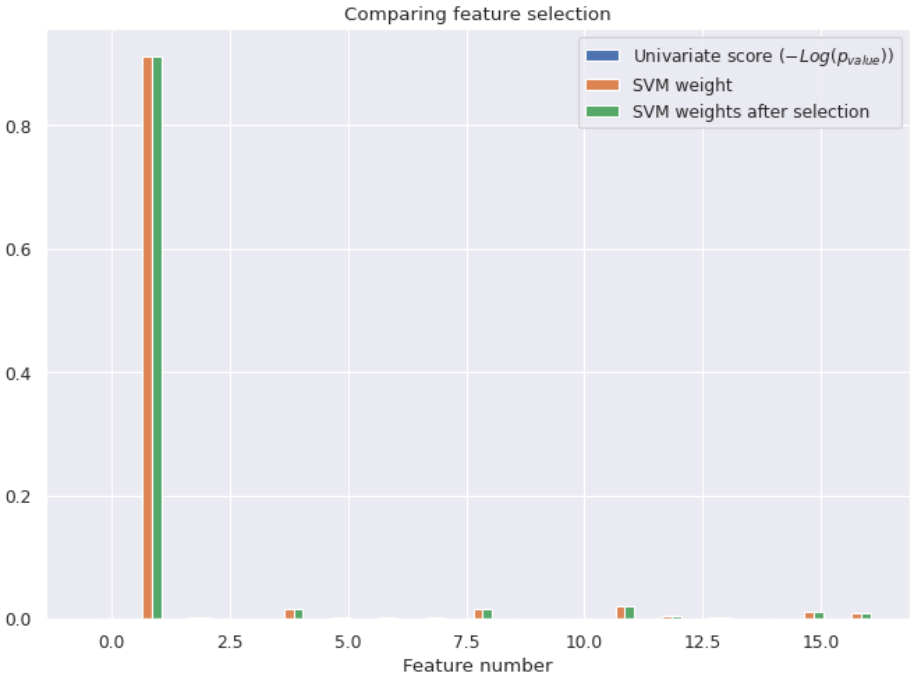


*Figura 2. Importancia de los features para un modelo de SVM*

| | f_name | f_index | univariate_score | svm_weight | svm_weight_selected |
|---|---|---|---|---|---|
| **1** | amt | 1 | NaN | 0.911100 | 0.911100 |
| **11** | cat_risk_score | 11 | NaN | 0.019234 | 0.019234 |
| **4** | long | 4 | 0.0 | 0.016651 | 0.016651 |
| **8** | merch_long | 8 | 0.0 | 0.015560 | 0.015560 |
| **15** | trans_hour | 15 | 0.0 | 0.011553 | 0.011553 |
| **16** | client_yob | 16 | 0.0 | 0.009800 | 0.009800 |
| **12** | trans_month | 12 | 0.0 | 0.004175 | 0.004175 |
| **7** | merch_lat | 7 | 0.0 | 0.001902 | 0.001902 |
| **13** | trans_day | 13 | 0.0 | 0.001726 | 0.001726 |
| **5** | city_pop | 5 | 0.0 | 0.001711 | 0.001711 |
| **2** | gender | 2 | 0.0 | 0.001701 | 0.001701 |
| **6** | unix_time | 6 | 0.0 | 0.001561 | 0.001561 |
| **9** | state_code | 9 | 0.0 | 0.000892 | 0.000892 |
| **0** | cc_num | 0 | 0.0 | 0.000891 | 0.000891 |
| **3** | lat | 3 | 0.0 | 0.000764 | 0.000764 |
| **14** | trans_weekday | 14 | 0.0 | 0.000736 | 0.000736 |
| **10** | city_code | 10 | 0.0 | 0.000043 | 0.000043 |

*Tabla 1. Clasificación de features para SVM*

Diego A. Velandia C.
C.C. 1019054912

Se encuentra que los 6 primeros features reúnen el 98.38% de la importancia del total de los features. De modo que se emplean dichas variables para el entrenamiento de modelos de aprendizaje supervisado.

## *Resultado de modelos de Machine Learning*

A partir de los features identificados, se extrae un dataset exclusivamente con dichas variables y se genera un dataset de entrenamiento balanceado, donde se toman el 70% de los registros donde se identificó fraude y se añade la misma cantidad de registros en donde se identificó que no hubo fraude.

Para calcular el desempeño de los modelos se consideran 4 tipos de observaciones, con base en el valor predicho por el modelo y el valor verdadero de la variable predicha, como se muestra en la tabla 3.

| Nombre | Valor predicho | Valor Verdadero |
|---|---|---|
| True Positive | 1 | 1 |
| False Negative | 1 | 0 |
| True Negative | 0 | 0 |
| False Positive | 0 | 1 |

*Tabla 2. Valores en matriz de confusión*

Estos valores se muestran en una gráfica llamada matriz de confusión, que agrupa el conteo de las predicciones de las variables, como se muestra en la figura 3.

| | |
|---|---|
| True Negative | False Positive |
| False Negative | True Positive |

*Figura 3. Ejemplo de matriz de confusión*

Se determinan tres parámetros para medir el desempeño de los modelos con base en la matriz de confusión. Los parámetros se calculan conforme a las siguientes fórmulas:

$$Precisión = 100 * \frac{(\sum True\ Positive) + (\sum True\ Negative)}{Total\ observaciones}$$

$$Sensibilidad = 100 * \frac{(\sum True\ Positive)}{(\sum True\ Positive) + (\sum False\ Negative)}$$

$$Especificidad = 100 * \frac{(\sum True\ Negative)}{(\sum True\ Negative) + (\sum False\ Positive)}$$

Diego A. Velandia C.
C.C. 1019054912

El dataset balanceado es empleado para entrenar diferentes modelos de Supervised Machine Learning y a partir de la matriz de confusión resultante se obtiene, para modelos sin optimización de hiperparámetros, precisiones sobre el 94%, como se muestra en la tabla 2.

| Modelo | Precisión [%] | Sensibilidad [%] | Especificidad [%] |
|---|---|---|---|
| Support Vector Machine | 94.83 | 75.84 | 95.05 |
| Extreme Gradient Boosting | 96.39 | 95.69 | 96.40 |
| AdaBoost | 95.22 | 92.45 | 95.25 |
| Decision Tree | 96.19 | 96.58 | 96.18 |
| Random Forest | 97.29 | 96.71 | 97.34 |
| Gaussian NB | 96.91 | 73.44 | 97.19 |

*Tabla 3. Desempeño de algunos modelos implementados de Machine Learning*

Se tiene que el modelo con la mayor sensibilidad y especificidad es el Random Forest, seguido de los modelos de Decision Tree y Extreme Gradient Boosting (XGBoost). El desempeño de los tres modelos puede mejorarse empleando optimización de hiperparámetros.

Diego A. Velandia C.
C.C. 1019054912