UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN

FACULTAD DE CIENCIAS FÍSICO MATEMÁTICAS

MAESTRÍA EN CIENCAS DE DATOS

APRENDIZAJE AUTOMÁTICO

MAESTRO: JOSÉ ALBERTO BENAVIDES VÁZQUEZ

TAREA #4 SELECCIÓN DE CARACTERÍSTICAS

ALUMNO: EDWIN MARTÍN ROMERO SILVA

MATRÍCULA: 1731276

Índice

Selección de Características

Introducción

Esta tarea está enfocada en aplicar técnicas que nos permitan eliminar características que sean estadísticamente redundantes o aporten poca información a los modelos.

Los beneficios de estas técnicas están enfocados en disminuir tiempos de entrenamiento de modelos con muchas características e incluso mejorar sus resultados.

Recordemos que el problema que estamos solucionando es predecir una variable binaria (0/1) con variables predictoras continuas y categóricas, así que probablemente unos métodos sean más favorables que otros.

Aplicaré 4 métodos de filtro y 1 método de envoltura, los 5 vistos en clase.

- ANOVA
- Valor R de correlación
- Umbral de la varianza
- Información mutua
- Selección exhaustiva de características

Adicional a esto, investigué cuales métodos son mas efectivos para mi problema y encontré el siguiente recuadro en un video de youtube
https://www.youtube.com/watch?v=l3OmtvcaTmM

		Predicción (y)	
		Continuo	**Categórico**
Características (x)	**Continuo**	Correlación de Pearson	Análisis Discriminante Lineal
	Categórico	ANOVA	Chi-cuadrado

Ya que la mayoría de mis variables son continuas y mi variable objetivo es binaria, me pareció interesante probar el método Análisis Discriminante Lineal.

Al final, concluiré sobre lo observado con todos los métodos aplicados.

Desarrollo

1)Métodos de Filtro

Estos métodos usan estadísticos para determinar umbrales sobre los que elegir características. Suelen ser más rápidos que otros métodos, mas no suelen incluir interacción entre variables. (Párrafo extraído de la clase).

Apliqué en Python los métodos de filtro utilizados en clase excluyendo las variables categóricas y enfocándonos en las variables continuas y discretas.

```
y = df[['Loan Status']]
x = df.select_dtypes(include=['float64', 'int32', 'int64'])
x = x.drop(columns = ['Loan Status'])
```

a) ANOVA de valor F

El estadístico F se utiliza para evaluar si hay diferencias significativas entre grupos. Un valor alto, indica alta relación lineal; valores menores, lo contrario.

- **Hipótesis Nula (H0):** No hay diferencias significativas entre los grupos.
- **Hipótesis Alternativa (H1):** Al menos un grupo es diferente de los demás.

Si p < α, generalmente se rechaza H0.

En otras palabras, este test sirve para comprobar si todas las medias poblacionales (en el caso de ANOVA) son iguales o que el modelo más simple es tan bueno como el modelo más complejo (en el caso de regresión).

```
from sklearn.feature_selection import f_regression

f_value = f_regression(x, y)
```

```
f_value = pd.DataFrame(f_value[0], f_value[1]).reset_index(drop = False)
f_value = pd.concat([pd.DataFrame(x.columns), f_value], axis = 1)
f_value.columns = ['VARIABLE', 'F_PVALOR' , 'F_ESTADISTICO']

def decreto(p):
    if p < 0.05:
        resp = 'Significativa'
    else:
        resp = 'No Significativa'
    return resp
f_value['F_DECRETO'] = f_value['F_PVALOR'].apply(decreto)
f_value
```

	VARIABLE	F_PVALOR	F_ESTADISTICO	F_DECRETO
0	Loan Amount	0.245327	1.349741	No Significativa
1	Funded Amount	0.723112	0.125531	No Significativa

F_DECRETO	F_DECRETO
No Significativa	21
Significativa	5

Utilizando este método con un Alpha de 0.05, 5 de las 26 variables resultan significativas.

b) Valor R de correlación

Este valor mide la fuerza de la relación lineal entre 2 variables.

- Un valor cercano a 1 indica una relación positiva fuerte
- Un valor cercano a -1 indica una relación negativa fuerte
- Un valor cercano a 0 indica una relación débil

```
from sklearn.feature_selection import r_regression

r_value = pd.DataFrame(r_regression(x,y))
r_value = pd.concat([pd.DataFrame(x.columns), r_value ], axis = 1)
r_value.columns = ['VARIABLE' , 'R_CORRELACION']
r_value
```

```
r_value['R_TIPO_RELACION'] = np.where((r_value['R_CORRELACION'] > 0 ), 'positiva', 'negativa')
r_value['R_CORRELACION_ABS'] = abs(r_value['R_CORRELACION'])
r_value
#plt.bar(r_value['VARIABLE'], r_value['R_CORRELACION_ABS'])
#plt.show()
```

	VARIABLE	R_CORRELACION	R_TIPO_RELACION	R_CORRELACION_ABS
0	Loan Amount	-0.004473	negativa	0.004473
1	Funded Amount	0.001364	positiva	0.001364

Con este método obtenemos que 15 de las variables tienen una relación positiva y 11 una relación negativa.

R_TIPO_RELACION	R_TIPO_RELACION
negativa	11
positiva	15

El valor absoluto del coeficiente de correlación es 0.01059, este es un valor muy bajo e indica que prácticamente ninguna de las variables tiene una relación lineal fuerte con la variable objetivo.

Utilizando este método, prácticamente todas las variables estarían eliminadas.

	R_CORRELACION	R_CORRELACION_ABS
count	26.000000	26.000000
mean	0.001500	0.003956
std	0.004987	0.003307
min	-0.007073	0.000091
25%	-0.002561	0.001175
50%	0.000845	0.003416
75%	0.004019	0.005706
max	0.010590	0.010590

c) Umbral de la Varianza

Consiste en descartar características con baja varianza, en el supuesto de que no aportan tanta información al modelo. Requiere que las características estén normalizadas.

Se suelen eliminar características con varianza menor a 0.2

```
# Normalización de variables
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
scaled = scaler.fit_transform(x)
x_scaled = pd.DataFrame(scaled,columns=x.columns)
x_scaled
```

```
from sklearn.feature_selection import VarianceThreshold

selector=VarianceThreshold()
selector.fit_transform(x_scaled)
selector.variances_
```

Con este método todas las variables estarían eliminadas ya que ninguna varianza es mayor a 0.2

	VARIANZA_UMBRAL
count	26.000000
mean	0.024866
std	0.019369
min	0.000438
25%	0.010246
50%	0.020888
75%	0.031695
max	0.072383

d) Información Mutua

Este método mide la dependencia entre variables, captura relaciones no lineales entre variables.

Funciona tanto para regresiones como para clasificaciones.

```
from sklearn.feature_selection import mutual_info_classif

mi = mutual_info_classif(x, y, random_state=0)
#mutual_info_regression

mi = pd.DataFrame(mi)
mi = pd.concat([pd.DataFrame(x.columns), mi], axis = 1)
mi.columns = ['VARIABLE', 'MI_CLASSIF_Y']
mi
```

	VARIABLE	MI_CLASSIF_Y
0	Loan Amount	0.002454
1	Funded Amount	0.000493
2	Funded Amount Investor	0.000370
3	Term	0.005418
4	Interest Rate	0.000155

No existe un umbral sugerido para eliminar/conservar variables con este método, pero se sabe que un valor de 0, indica que no existe una relación y que las variables son independientes.

Parece que individualmente, no existen relaciones 'no lineales' fuertes entre las variables predictoras y la variable objetivo Loan Status.

Si eliminaramos todas las variables con un valor de información mutua de 0, estaríamos eliminando 11 variables y conservando 15.

	MI_CLASSIF_Y
count	26.000000
mean	0.000704
std	0.001214
min	0.000000
25%	0.000000
50%	0.000352
75%	0.000671
max	0.005418

e) Linear Discriminant Analysis

El Análisis Discriminante Lineal (Linear Discriminant Analysis, LDA) es en una técnica de reducción de dimensionalidad.

Su objetivo principal es proyectar datos multidimensionales en un espacio de menor dimensión (generalmente un espacio unidimensional) mientras se intenta maximizar la separación entre las clases. LDA es ampliamente utilizado en problemas de clasificación y reconocimiento de patrones.

- Mejora la separabilidad de las clases: LDA selecciona proyecciones que maximizan la distancia entre las medias de las clases y minimizan la varianza intraclase.
- Reducción de dimensionalidad: Al proyectar datos en un espacio de menor dimensión, LDA puede ser visto como un método de reducción de dimensionalidad.

Esto significa que, en la práctica, las características que contribuyen más a la separación entre clases tienden a tener un peso más alto en la proyección resultante.

En este proceso, algunas de las características originales pueden no ser seleccionadas en la proyección resultante, lo que puede considerarse como una forma indirecta de selección de características.

LinearDiscriminantAnalysis

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.2,

# Inicializar y ajustar el modelo LDA
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)

# Realizar predicciones en el conjunto de prueba
y_pred = lda.predict(X_test)

# Calcular la precisión del modelo
accuracy = accuracy_score(y_test, y_pred)
print("Precisión del modelo LDA:", accuracy)
```

```
Precisión del modelo LDA: 0.9097309716149115
```

Feature	Coefficient
Collection 12 months Medical	2.923756e-02
Term	-5.026772e-03
Open Account	-4.149908e-03
Interest Rate	3.128776e-03
Collection Recovery Fee	-3.083685e-03
Revolving Utilities	5.769760e-04
Debit to Income	-2.762871e-04
Total Accounts	-2.160012e-04
Total Collection Amount	2.473351e-05
Recoveries	-1.095797e-05
Total Received Interest	3.585787e-06
Loan Amount	-1.767066e-06
Funded Amount Investor	4.714760e-07
Funded Amount	2.667103e-07
Revolving Balance	2.302312e-07

La función de LDA arroja un score de métrica **accuracy** de 0.9097. Podría verse como una buena puntuación ya que se acerca a 1, pero la realidad es que el 10% de los casos son de clase “1” y 90% de los casos son de clase “0”. Así que en realidad una persona alcanzaría la misma puntuación si arbitrariamente a todos los casos los señalara como de clase “0”.

2)Métodos de envoltura

Métodos que exploran subconjuntos de combinaciones de características que mejoren algún desempeño de modelos de AA.

- **Ventajas**: Se estudian las relaciones de las características en el modelo, a diferencia de los métodos de filtro, donde la relación de características dependía de estadísticos.
- **Desventajas**: A mayor complejidad del modelo y número de características, mayor consumo de recursos y tiempos de ejecución.

a) Selección exhaustiva de características

La Selección de características exhaustiva evalúa todas las combinaciones de características y devuelve los valores que optimizan el modelo.

```
from sklearn.linear_model import LogisticRegression
from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS
```

```
lr = LogisticRegression

efs = EFS(estimator=lr,
          min_features=1,
          max_features=4,
          scoring='roc_auc'
          cv=3,
          n_jobs=-1)

efs = efs.fit(x, y)
```

Escogí la regresión logística en lugar de la regresión lineal, ya que mi problema es de clasificación binaria, reduje el número máximo de variables para que no le costara tanto a mi computadora, y escogí el ROC como métrica de desempeño, obtuve lo siguiente:

```
metric_dict = efs.get_metric_dict()
df_efs = pd.DataFrame(metric_dict).T
df_efs.sort_values('avg_score', ascending=False,  inplace = True)
df_efs_best_10 = df_efs.iloc[:10]
df_efs_best_10
```

	feature_idx	cv_scores	avg_score	feature_names	ci_bound	std_dev	std_err
14	(14,)	[0.5254068917166721, 0.5208805940862223, 0.531...	0.526006	(Total Collection Amount,)	0.010013	0.00445	0.003146
119	(13, 14)	[0.5255931874378071, 0.5222018798752452, 0.528...	0.525443	(Collection 12 months Medical, Total Collectio...	0.005823	0.002588	0.00183
116	(11, 14)	[0.5237751315472061, 0.5175578232370586, 0.529...	0.523747	(Recoveries, Total Collection Amount)	0.011345	0.005042	0.003565

Tomé un curso en el trabajo llamado **Modelos de Riesgo de Crédito**, impartido por la ABM y el profesor nos compartió una tabla para interpretar el resultado del ROC.

Interpretation	Area under the ROC curve
Bad	50% a 60%
Poor	60% a 70%
Fair	70% a 80%
Good	80% a 90%
Excelent	90% a 100%

La mejor combinación de variables que encontró este modelo obtiene un área bajo la curva de 52%, la tabla nos dice que esta es una puntuación mala.

Eliminar todas las variables que no se encuentren en la mejor combinación, no serviría de nada, ya que incluso tomando solo las de la mejor combinación, no obtendré un buen modelo, según la curva ROC.

Conclusiones

Sobre el método ANOVA y Correlación:

El método de ANOVA eliminaría 21 variables utilizando un alfa de 0.05.

El método de correlación eliminaría todas las variables utilizando un criterio de >0.5

Estos métodos no son adecuados para el problema que quiero resolver, ambos métodos funcionarían al tratar de predecir variables continuas y este es un problema de clasificación.

		Predicción (y)	
		Continuo	Categórico
Características (x)	Continuo	Correlación de Pearson	Análisis Discriminante Lineal
	Categórico	ANOVA	Chi-cuadrado

Sobre el método de umbral de la varianza:

El método de umbral de la varianza utilizando límite inferior de 0.2, eliminaría todas las variables, por lo tanto, no puedo utilizarlo para seleccionar las mejores características.

Sobre el método de selección exhaustiva de características:

Este método nos permite incorporar el modelo que deseamos utilizar, escogí una regresión logística ya que me parece adecuado para resolver mi problema de clasificación binaria.

Sin embargo, este método nos dice que la mejor combinación de variables obtiene un área bajo la curva de 52%, lo cual es muy malo.

Conservar únicamente las variables que sugiere este método significaría quedarnos con variables que no podrán resolver este problema de clasificación, según la curva ROC, por lo que no me parece adecuado utilizar sus resultados.

Interpretation	Area under the ROC curve
Bad	50% a 60%
Poor	60% a 70%
Fair	70% a 80%
Good	80% a 90%
Excelent	90% a 100%

Sobre el método de información mutua y LDA:

El método de LDA funciona para seleccionar las características mas importantes en un problema de clasificación con variables predictoras continuas.

El método de información mutua es capaz de detectar relaciones no lineales y Python ofrece opciones tanto de regresión como de clasificación (utilicé la de clasificación) por lo que ambos métodos son adecuados para este problema.

No existe un umbral sugerido para eliminar variables, pero sabemos que un valor de Información mutua de 0 es el peor puntaje e indica que las variables son independientes. También sabemos que los coeficientes de LDA indican una mayor relación con la variable objetivo entre más grandes sean.

Una forma de utilizar ambos métodos en conjunto sería eliminar las variables que obtengan un valor de 0 en ambos métodos, es decir que ambos métodos indican que la variable aporta poca información para resolver el problema de clasificación.

Bajo este criterio estaríamos eliminando 6 variables y conservando 20.

		De LDA:	De Información Mutua:	
#	Variable	Abs(Coefficient)	MI_CLASSIF_Y	Dictámen
1	Public Record	0.0637	-	Pasa al menos 1 método
2	Inquires - six months	0.0421	-	Pasa al menos 1 método
3	Delinquency - two years	0.0372	-	Pasa al menos 1 método
4	Collection 12 months Medical	0.0261	0.0007	Pasa al menos 1 método
5	Total Received Late Fee	0.0068	-	Pasa al menos 1 método
6	Term	0.0048	0.0054	Pasa al menos 1 método
7	Open Account	0.0040	0.0004	Pasa al menos 1 método
8	Collection Recovery Fee	0.0032	0.0026	Pasa al menos 1 método
9	Interest Rate	0.0031	0.0002	Pasa al menos 1 método
10	Revolving Utilities	0.0006	0.0016	Pasa al menos 1 método
11	Last week Pay	0.0005	-	Pasa al menos 1 método
12	Total Accounts	0.0003	0.0004	Pasa al menos 1 método
13	Debit to Income	0.0002	0.0005	Pasa al menos 1 método
14	Total Collection Amount	0.0000	0.0016	Pasa al menos 1 método
15	Recoveries	0.0000	0.0004	Pasa al menos 1 método
16	Total Received Interest	-	0.0007	Pasa al menos 1 método
17	Loan Amount	-	0.0025	Pasa al menos 1 método
18	Revolving Balance	-	0.0003	Pasa al menos 1 método
19	Funded Amount	-	0.0005	Pasa al menos 1 método
20	Funded Amount Investor	-	0.0004	Pasa al menos 1 método
21	Dif_monto_sol_monto_aut	-	-	Eliminada
22	Dif_monto_sol_monto_fin	-	-	Eliminada
23	Home Ownership	-	-	Eliminada
24	Total Current Balance	-	-	Eliminada
25	Total Revolving Credit Limit	-	-	Eliminada
26	Dif_monto_fin_monto_aut	-	-	Eliminada

Las 20 variables que conservo son las más importantes ya que según los métodos adecuados para resolver un problema de clasificación, aportarían algo de información a un modelo de AA.