

APRENDIZAJE AUTOMÁTICO AVANZADO
INFORME TÉCNICO UNIDAD I – INGENIERÍA DE CARACTERÍSTICAS

PRESENTADO POR:

Edgar Leandro Jiménez Jaimes

Santiago Echeverri Calderón

DOCENTE:

José Lisandro Aguilar Castro

UNIVERSIDAD EAFIT

MEDELLÍN

MAESTRÍA EN CIENCIAS DE LOS DATOS Y ANALÍTICA

FEBRERO DE 2020

1. Objetivo de la iteración

Investigar, analizar y aplicar la ingeniería de características a un dataset de estudio para la posterior construcción de modelos de clasificación binaria y estudiar los desempeños de los modelos.

2. Contextualización del problema

Las técnicas de Ingeniería de Características se implementarán en un modelo de clasificación de pacientes con riesgo de enfermedad cardíaca coronaria. La Organización Mundial de la Salud ha estimado que 12 millones de muertes ocurren en todo el mundo cada año, debido a enfermedades del corazón¹. El pronóstico temprano de las enfermedades cardiovasculares puede ayudar a tomar decisiones sobre los cambios en el estilo de vida en pacientes de alto riesgo y, a su vez, reducir las complicaciones.

El conjunto de datos a utilizar proviene de un estudio cardiovascular en curso en los residentes de la ciudad de Framingham, Massachusetts, el cual se encuentra disponible en:

<https://www.kaggle.com/dileep070/heart-disease-prediction-using-logistic-regression>

El objetivo de la clasificación es predecir si un paciente tiene riesgo de enfermedad coronaria en los próximos 10 años. El conjunto de datos proporciona la información de más de 4,000 pacientes y cuenta con 15 atributos y 1 etiqueta binaria de salida. Cada atributo es un factor de riesgo potencial. Existen factores de riesgo demográficos, conductuales y médicos.

3. Metodología

La metodología que se utilizó en el libro de jupyter consta de 4 etapas:

I.Exploración y preprocesamiento de los datos.

II.Implementación de técnicas de Ingeniería de Características utilizando Feature Tools.

¹ World Health Organization 2020, *Cardiovascular diseases (CVDs)*, viewed 16 Feb 2020, <[https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))>.

III.Implementación de técnicas de Ingeniería de Características utilizando enfoque mixto entre Backward Elimination y Feature Tools.


IV.Entrenamiento y comparación del desempeño de un clasificador con el conjunto de datos original y con el conjunto de datos después de las técnicas de Ingeniería de Atributos. Se utilizaron dos modelos de clasificación binaria para esta etapa: Random Forest y Regresión Logística.

3.1. Exploración y preprocesamiento de los datos.

Se realizó una estadística descriptiva básica de los datos y se encontró que el conjunto de datos ya había sido procesado, las variables categóricas habían sido convertidas a numéricas y no se presentaban datos atípicos. Sin embargo, se encontró que algunas características contenían valores nulos:

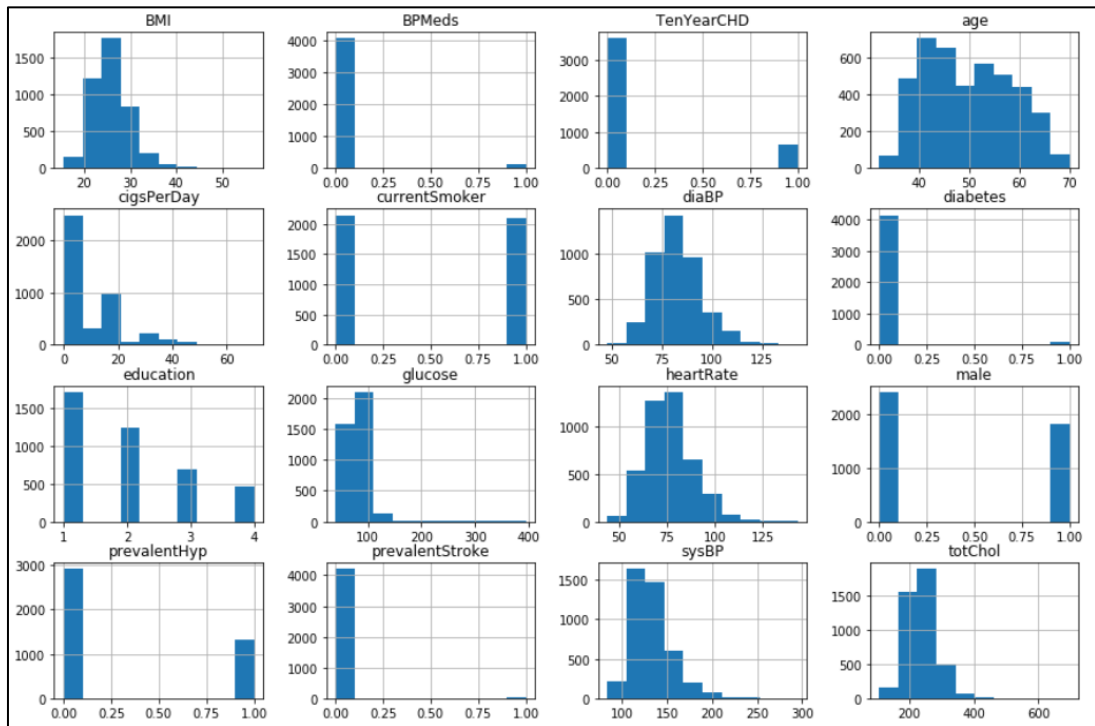
male	4238	non-null	int64
age	4238	non-null	int64
education	4133	non-null	float64
currentSmoker	4238	non-null	int64
cigsPerDay	4209	non-null	float64
BPMeds	4185	non-null	float64
prevalentStroke	4238	non-null	int64
prevalentHyp	4238	non-null	int64
diabetes	4238	non-null	int64
totChol	4188	non-null	float64
sysBP	4238	non-null	float64
diaBP	4238	non-null	float64
BMI	4219	non-null	float64
heartRate	4237	non-null	float64
glucose	3850	non-null	float64
TenYearCHD	4238	non-null	int64

Tipos de datos y cantidad de registros de las características

 male	0
age	0
education	105
currentSmoker	0
cigsPerDay	29
BPMeds	53
prevalentStroke	0
prevalentHyp	0
diabetes	0
totChol	50
sysBP	0
diaBP	0
BMI	19
heartRate	1
glucose	388
TenYearCHD	0

Cantidad de valores nulos para cada característica

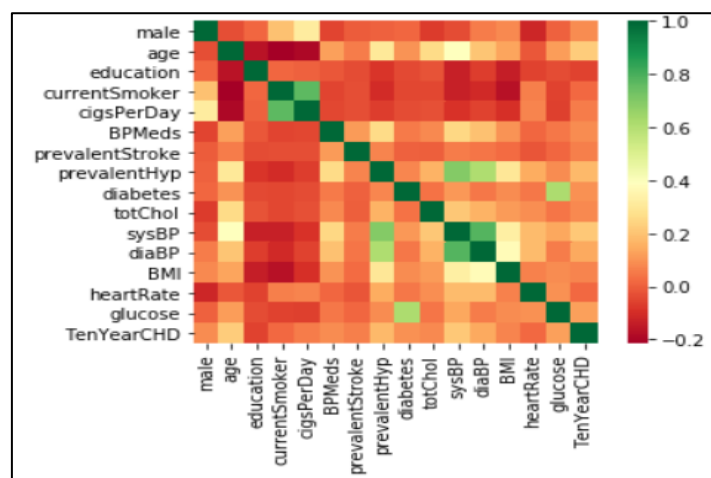
Como el propósito del presente ejercicio es la ingeniería de atributos y no el tratamiento de valores nulos se tomó la decisión de eliminar los registros que carecían de alguno de los atributos. Después de realizar el borrado el dataset se redujo a 3,656 observaciones.



Histogramas de cada variable del conjunto de datos

Revisamos también los histogramas de frecuencia de los datos, en el podemos ver a priori como se comportan las distribuciones de los datos, también podemos observar como se marcan claramente las variables que son categóricas, por ejemplo: BPMeds, currentSmoker, Diabetes, entre otras, y las variables que son continuas, por ejemplo: heartRate, sysBP, entre otras.

En esta etapa también se revisó la matriz de correlaciones y se evidenció que el conjunto contiene variables altamente correlacionadas como el atributo currentSmoker y cigsPerDay, lo cual tiene mucho sentido.



3.2. Implementación de técnicas de Ingeniería de Características utilizando Feature Tools.

Se plantearon 3 métodos de ingeniería de atributos utilizando *Feature Tools* con el fin de mejorar la calidad y reducir el tamaño del conjunto de datos:

Método	Propósito	Referencia
a) Nuevos atributos mediante transformaciones y agregados	Creación de nuevos atributos	Khalid, S., Khalil, T., & Nasreen, S. (2014, August). A survey of feature selection and feature extraction techniques in machine learning. In 2014 Science and Information Conference (pp. 372-378). IEEE.
b) Eliminación de variables altamente correlacionadas	Selección de atributos	
c) Selección basada en los pesos de un clasificador SVM lineal con regularización L1	Selección de atributos	Brank, Janez & Grobelnik, Marko & Milic-Frayling, Natasa & Mladenić, Dunja. (2002). Feature Selection Using Linear Support Vector Machines. Technical report, Microsoft Research.

a) Creación de nuevos atributos mediante transformaciones y agregados

El primer método consistió en crear nuevos atributos a partir de los datos que ya se tenían en el conjunto original. Para esto se usó el concepto de primitivas mediante agregaciones y transformaciones, el cual consiste en calcular funciones de agregado sobre los datos continuos en las diferentes clases de los atributos nominales, ordinales y booleanos. El resultado es una relación de uno a muchos con estadísticas como, por ejemplo:

- La media
- El máximo
- El mínimo
- Percentiles
- Desviación estándar
- Entre otras.

Así, por ejemplo, dos nuevas características que se crearon fue la asimetría de la presión arterial para los pacientes que fuman y para los que no fuman.

	currentSmoker.SKEW(df2.age)	age	currentSmoker
index			
2	0.503753	48	1
3	0.503753	61	1
4	0.503753	46	1
7	0.503753	45	1
9	0.503753	43	1

Asimetría de la edad en personas que fuman

	currentSmoker.SKEW(df2.age)	age	currentSmoker
index			
0	-0.038574	39	0
1	-0.038574	46	0
5	-0.038574	43	0
6	-0.038574	63	0
8	-0.038574	52	0

Asimetría en la edad de las personas que NO fuman

La asimetría recordemos que es una medida de la distribución de probabilidad de una variable aleatoria con valores reales sobre su media. Así pues, tenemos nuevas columnas creadas a partir de los datos originales y realizando transformaciones y aplicaciones estadísticas para posteriormente evaluar si estas nuevas variables agregan valor y mejoran el desempeño de modelos de predicción y clasificación

Estas operaciones no son difíciles de programar, pero es un proceso que lleva mucho tiempo pues cada nueva característica requiere varios pasos para su construcción. Por este motivo se usó la librería FeatureTools para Python, como se menciona en el título del enfoque, la cual permitió automatizar esta tarea.

Con este método se crearon 368 nuevas variables para un total de 383 variables en el conjunto.

b) Eliminación de variables altamente correlacionadas

Se contaba entonces con un dataset enriquecido con nueva información, pero con alta dimensionalidad. Por lo tanto, el siguiente paso consistió en determinar las variables de mayor relevancia y descartar el resto, para esto entonces se utilizó en principio la matriz de correlación, que explicaremos a continuación y posteriormente un modelo con regularización.

Se calculó entonces la matriz de correlación sobre el conjunto con los nuevos atributos con el fin de filtrar las variables con alta colinealidad. Los coeficientes de correlación se calcularon en valor absoluto, pues se deseaban filtrar las variables independientemente de si su relación era directa o inversa.

Para realizar el filtrado se definió un umbral de corte de 0.95, es decir, se eliminaron las variables con un coeficiente de Pearson mayor a 0.95 en valor absoluto. Este valor se definió buscando que el filtro no fuera muy agresivo y sólo se eliminaran las variables con una colinealidad muy alta y definida, pues adicional a este método de selección de variables, como se mencionó anteriormente, se aplicaría un segundo método basado en los pesos de un clasificador SVM y se buscaba que dicho clasificador contara con más información.

En este filtrado de colinealidad se removieron 298 variables, así que el conjunto se redujo a 85 características.

c) Selección de variables basada en los pesos de un clasificador SVM lineal con regularización L1.

La regularización consiste en agregar una penalización a los diferentes parámetros del modelo de aprendizaje automático para reducir la libertad del modelo. En la regularización del modelo SVM lineal, la penalización se aplica sobre los coeficientes que multiplican cada una de las características. En particular, la regularización Lasso o L1 tiene la propiedad de reducir algunos de los coeficientes a cero. Este concepto se puede usar como un indicador de la relevancia de la variable, de esta forma los coeficientes que se reduzcan a cero pueden eliminarse del modelo.

Antes de entrenar el modelo SVM lineal, y con el fin de garantizar un buen desempeño del clasificador, se verificó el balance de las clases de la variable de salida. En esta validación se encontró que la variable estaba desbalanceada, 15% de las observaciones eran 1 y el 85% eran 0.

Se usó entonces una técnica de oversampling sobre los datos de entrenamiento, de tal manera que se contara con la misma cantidad de observaciones en cada clase. Para esto se usó el método SMOTE (Synthetic Minority Oversampling Technique²) mediante la implementación de la librería `imblearn.over_sampling`.

Una vez se tuvo un conjunto de entrenamiento con las clases balanceadas se entrenó el clasificador SVM lineal con regularización L1. De este modelo se seleccionaron las variables cuyos coeficientes fueran mayores que 0 y se obtuvieron entonces las siguientes 22 variables:

² Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16, 321-357.

Male	Age	Education	CurrentSmoker	cigsPerDay
BPMeds	prevalenceStroke	Diabetes	totChol	sysBP
diaBP	BMI	heartRate	Glucose	Percentile(cigsPerDay)
Percentile(sysBP)	Percentile(BMI)	Percentile(glucose)	educationSTD(totChol)	educationMAX(glucose)
educationMIN(glucose)	prevalenceHyp			

Después de finalizar la ingeniería de características se obtuvo un conjunto con 22 variables, 15 del conjunto inicial y 7 creadas a partir de los mismos datos (Subrayados y negrilla).

3.3. Implementación de técnicas de Ingeniería de Características utilizando enfoque mixto entre Backward Elimination y Feature Tools.

Esta implementación la decidimos realizar a partir de unos resultados que se tenían en una solución de un usuario de Kaggle y que habían dado buenos desempeños en un modelo. A continuación, explicamos en que consiste este enfoque mixto.

En primer lugar, vamos a replicar la solución presentada en Kaggle donde realizan una selección de características utilizando Backward Elimination a partir del valor p de un modelo de regresión logística, en el cual al final de la iteración deja las variables significativas bajo la mirada del valor p. Posteriormente utilizamos este conjunto de variables significativas para crear, a través de Feature Tools, más variables con el conjunto de transformaciones que se presento anteriormente, utilizando por ejemplo media, moda, máximo, mínimo, etc. Pero en esta ocasión no utilizamos todas las variables del dataset sino las que nos dieron significativas a partir de la solución presentada en Kaggle.

La solución presentada en Kaggle³ fue de la siguiente:

Inicia eliminando del conjunto de datos la variable “*Education*”, posteriormente elimina todos los registros que contengan valores nulos, tal cual lo realizamos en este trabajo. Luego, agrega

³ <https://www.kaggle.com/dileep070/logistic-regression>

una columna de 1 al conjunto de datos, esta se llamará “Constante” y será utilizada para el modelo de regresión logística más adelante.

En ese momento ajusta un modelo de regresión logística para todos los datos y presenta los resultados:

Optimization terminated successfully. Current function value: 0.376795 Iterations 7						
Dep. Variable:	TenYearCHD	No. Observations:	3656			
Model:	Logit	Df Residuals:	3641			
Method:	MLE	Df Model:	14			
Date:	Tue, 03 Mar 2020	Pseudo R-squ.:	0.1171			
Time:	23:02:03	Log-Likelihood:	-1377.6			
converged:	True	LL-Null:	-1560.3			
Covariance Type:	nonrobust	LLR p-value:	2.412e-69			
	coef	std err	z	P> z	[0.025	0.975]
const	-8.4895	0.695	-12.221	0.000	-9.851	-7.128
Sex_male	0.5540	0.109	5.082	0.000	0.340	0.768
age	0.0642	0.007	9.676	0.000	0.051	0.077
currentSmoker	0.0713	0.157	0.455	0.649	-0.236	0.378
cigsPerDay	0.0180	0.006	2.887	0.004	0.006	0.030
BPMeds	0.1576	0.234	0.673	0.501	-0.301	0.616
prevalentStroke	0.7060	0.489	1.444	0.149	-0.253	1.665
prevalentHyp	0.2332	0.138	1.689	0.091	-0.037	0.504
diabetes	0.0430	0.315	0.136	0.892	-0.575	0.661
totChol	0.0023	0.001	2.009	0.045	5.53e-05	0.004
sysBP	0.0157	0.004	4.139	0.000	0.008	0.023
diaBP	-0.0046	0.006	-0.718	0.473	-0.017	0.008
BMI	0.0081	0.013	0.635	0.525	-0.017	0.033
heartRate	-0.0031	0.004	-0.731	0.465	-0.011	0.005
glucose	0.0071	0.002	3.185	0.001	0.003	0.012

Resultados Reg. Log. Solución kaggle

Luego construye una función que realizará el proceso de Backward Elimination a partir del valor p ajustado a la regresión logística. Esta función es programada por el usuario que presenta la solución, nosotros la replicamos a nuestros datos:

```
def back_feature_elim (data_frame, dep_var, col_list):
    while len(col_list)>0 :
        model=sm.Logit(dep_var,data_frame[col_list])
        result=model.fit(dis=0)
        largest_pvalue=round(result.pvalues,3).nlargest(1)
        if largest_pvalue[0]<(0.05):
            return result
            break
        else:
            col_list=col_list.drop(largest_pvalue.index)
```

Función Backward elimination en Python

Los resultados al correr la función con todos los datos y al utilizar la función Backward Elimination se presentan a continuación:

	CI 95%(2.5%)	CI 95%(97.5%)	Odds Ratio	pvalue
const	0.000043	0.000275	0.000108	0.000
Sex_male	1.421955	2.161623	1.753206	0.000
age	1.054747	1.081654	1.068116	0.000
cigsPerDay	1.011102	1.027789	1.019412	0.000
totChol	1.000072	1.004483	1.002275	0.043
sysBP	1.013411	1.021985	1.017689	0.000
glucose	1.004002	1.010623	1.007307	0.000

Observemos que estas variables que se presentan tienen un P-value < 0.05, con lo cual fueron seleccionadas para utilizarse en la modelación posterior.

En este punto el usuario realiza un nuevo modelo de regresión logística con estas variables y presenta el resultado del desempeño:

	precision	recall	f1-score	support
0	0.86	1.00	0.93	623
1	0.80	0.11	0.19	109
accuracy			0.86	732
macro avg	0.83	0.55	0.56	732
weighted avg	0.86	0.86	0.82	732

Resultados de la reg. Log. Con variables significativas - Kaggle

Observemos que el modelo tiene un buen accuracy, 86%, sin embargo, la medida de recall tiene un valor bajo, y recordemos que este problema busca clasificar la presencia o no de una enfermedad, lo cual es muy importante tener en cuenta el recall.

Nosotros vamos a utilizar estas 6 variables sin contar la constante para generar nuevos atributos y buscar mejorar más adelante el desempeño de los modelos de clasificación binaria. En ese orden de ideas repetimos los mismos pasos del enfoque 1:

a) Creación de nuevos atributos mediante transformaciones y agregados.

A partir de la librería Feature Tools y con las nuevas variables generamos los nuevos atributos. De esta manera pasamos a tener 6 variables a contar con 103.

b) Eliminación de variables altamente correlacionadas.

De la misma manera que en el enfoque con todas las variables, utilizamos un umbral de 95% para considerar una colinealidad entre dos variables. De esta manera al aplicar este umbral de corte en la matriz de correlación el nuevo número de variables es de 93.

c) Selección de variables basada en los pesos de un clasificador SVM lineal con regularización L1.

Realizamos el mismo modelo con regularización L1 aplicado al conjunto de las 93 variables. El resultado de este proceso nos deja como resultado un dataset de 21 variables. Este dataset será entonces el que utilizaremos para crear un modelo de clasificación binaria y medir los desempeños.

Las variables con las que quedo este enfoque fueron las siguientes:

- age
- Sex_male
- cigsPerDay
- totChol
- sysBP
- glucose
- PERCENTILE(cigsPerDay)
- PERCENTILE(sysBP)
- PERCENTILE(glucose)

3.4. Entrenamiento y comparación del desempeño de un clasificador

Para validar si los métodos de ingeniería de características que se usaron repercutían en un mejor modelo de Machine Learning se decidió entrenar 2 modelos de clasificación: un Random Forest y una Regresión Logística.

Cada clasificador se entrenó con los 2 conjuntos de datos (el conjunto original y el conjunto con los nuevos atributos). Los resultados se presentarán en la sección de pruebas análisis de resultados de los modelos.

4. Pruebas y análisis de resultados

Ahora bien, tenemos entonces 3 conjuntos de datos que son resultado de este ejercicio:

- Dataset con variables originales: **16 variables**
- Dataset enfoque 1 Feature Tools: **21 variables**
- Dataset enfoque mixto Kaggle-Feature Tools: **21 variables**

Para cada uno de estos datasets vamos a realizar un modelo de clasificación binaria y vamos a realizar un análisis de las métricas de clasificación de los modelos a partir de los datos de train y test. Puntualmente nos vamos a enfoque en las métricas de test. Para esto, como se menciono anteriormente entrenamos 2 modelos a cada Dataset, el primero fue regresión logística y el segundo un random forest.

A cada modelo se le midió el desempeño utilizando tres métricas de rendimiento, el accuracy, el recall y el F1 score. y los resultados fueron los siguientes:

Clasificador	Desempeño con los atributos originales	Desempeño con los nuevos atributos	Desempeño con enfoque mixto
Random Forest	<ul style="list-style-type: none">• Accuracy: 0.81• Recall: 0.31• F1-score: 0.34	<ul style="list-style-type: none">• Accuracy: 0.81• Recall: 0.28• F1-score: 0.31	<ul style="list-style-type: none">• Accuracy: 0.79• Recall: 0.23• F1-score: 0.25
Regresión Logística	<ul style="list-style-type: none">• Accuracy: 0.58• Recall: 0.35• F1-score: 0.20	<ul style="list-style-type: none">• Accuracy: 0.68• Recall: 0.67• F1-score: 0.39	<ul style="list-style-type: none">• Accuracy: 0.67• Recall: 0.66• F1-score: 0.38

Es importante resaltar la importancia de seleccionar bien la métrica adecuada del problema, en nuestro caso, como se mencionó en párrafos anteriores, el problema busca clasificar bajo unos

parámetros si un paciente tendrá o no una enfermedad. Esto hace que el enfoque de la métrica sea muy importante, no es lo mismo decir que el paciente tendrá la enfermedad cuando realmente no la tenía a equivocarme al decir que no la tiene cuando realmente la tiene, lo cual sería muy negativo. Esto adicionando el problema de desbalanceo de clases en este tipo de problema hace que un análisis de metricas como el accuracy no sea el más adecuado.

De los resultados obtenidos tenemos diferentes valores para las metricas, realmente rendimientos muy altos como se quisieran obtener, pero este proceso buscaba de manera iterativa encontrar las mejores variables para lograr una predicción acertada. Lo que sí es positivo es que se lograron desempeños buenos en las medidas de recall comparado con la solución que nos presentaron en Kaggle, donde originalmente es de 0.11 y con estas nuevas características logró subir hasta 0.67.

Los resultados de la regresión logística tuvieron un desempeño superior en todos los enfoques en la métrica de Recall, y con los enfoques de Feature Tools original y mixto esta regresión también logra un mejor desempeño en F1 score, que también es positivo. Por lo cual bajo estas condiciones y resultados podemos decir que la regresión logística tiene un mejor desempeño en la clasificación de nuestro problema con la métrica deseada.

Al continuar analizando los resultados de la regresión logística, observamos que los enfoque de Feature Tools 1 y mixto tienen desempeños muy similares, lo cual en el random forest no es así, aun teniendo una pequeña diferencia, teniendo mejor desempeño el random forest con el enfoque Feature Tools 1.

5. Cuaderno Jupyter

El Notebook con la implementación puede encontrarse en:

https://github.com/santiagooc/CM0891-Aprendizaje-Automatico/blob/master/01_Ingenieria_Caracteristicas.ipynb