

TABLA DE CONTENIDOS

JUSTIFICACION

Porque se desarrollo el proyecto

01.

02.

DATASET

Informacion de nuestro set de datos

GENERACION DE VALOR

Porque y a quien le genera valor el proyecto

03.

04.

PROYECTO

Descripcion general del proceso, analisis estadistico, ing. de características, análisis de componentes, etc.

RECOMENDACIONES

Sobre el tipo de modelos que se usaron

05.

06.

CONCLUSIONES

Arquitectura de red y seleccion del modelo

Justificacion del Modelo

Mantenerse un paso por delante de las tendencias y los deseos de los consumidores es la única forma de seguir siendo competitivo. El análisis predictivo permite esto, ayudando a los especialistas en marketing a comprender los comportamientos y tendencias de los consumidores, predecir cambios futuros y planificar sus campañas en consecuencia.

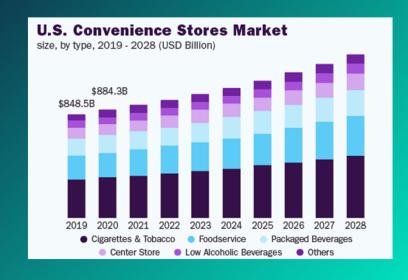
Creemos como equipo que por medio de desarrollar un modelo que nos permita predecir el costo de adquisición de clientes podemos generar una ventaja competitiva para las empresas en las cuales laboramos y generar ese valor adicional que nos permita diferenciarnos de nuestros competidores utilizando los datos como punto de partida para la toma de mejores decisiones buscando maximizar nuestros resultados.

DataSet

El dataset es de una cadena de tiendas de conveniencia de EEUU.

Es la tercera cadena mas grande desde 1988.





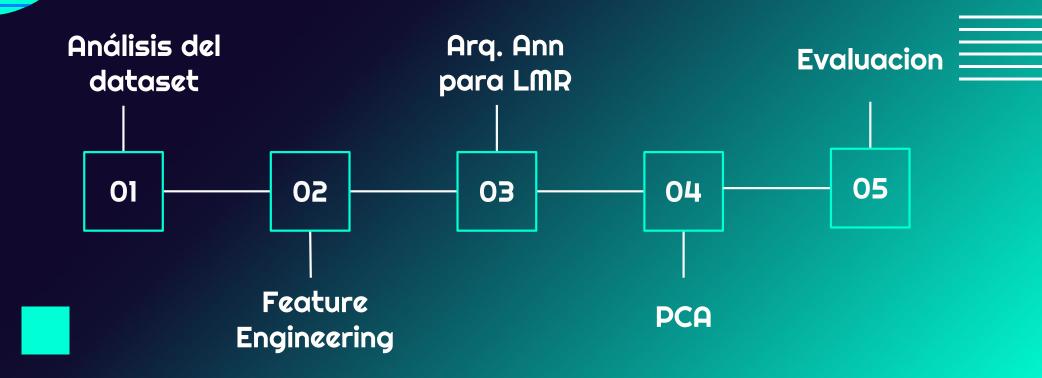


A través de la implementación del modelo, buscaremos aumentar el **valor** para los accionistas, incrementando la rentabilidad sobre el capital invertido, así **como** el **valor** de los bienes o el de los servicios para los consumidores.

Los resultados nos permitirán entender cuales son esos factores y elementos fundamentales que debemos tomar en consideración a la hora de definir la estrategia de captación de clientes y utilizar los recursos y presupuesto de la mejor manera posible.



FLUJO DEL PROYECTO

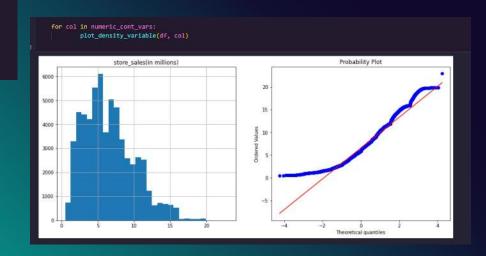


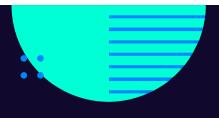
Analisis del dataset

```
def plot_density_variable(df, variable):

   plt.figure(figsize = (15,6))
   plt.subplot(121)
   df[variable].hist(bins=30)
   plt.title(variable)

   plt.subplot(122)
   stats.probplot(df[variable], dist="norm", plot=plt)
   plt.show()
```





Feature Engineering

```
from sklearn.preprocessing import OrdinalEncoder

ord_enc = OrdinalEncoder()

X['food_category'] = ord_enc.fit_transform(X[['food_category']])

X['food_department'] = ord_enc.fit_transform(X[['food_department']])

X['food_family'] = ord_enc.fit_transform(X[['food_family']])

X['promotion_name'] = ord_enc.fit_transform(X[['promotion_name']])

X['sales_country'] = ord_enc.fit_transform(X[['sales_country']])

X['marital_status'] = ord_enc.fit_transform(X[['marital_status']])

X['gender'] = ord_enc.fit_transform(X[['gender']])
```

Arq. ANN para LRM

```
entradas = X_train.shape[1]
lr = 0.01
batch_sz=150

regressor = Sequential()
regressor.add(Dense(input_dim=entradas, units=20, kernel_initializer='uniform', activation='relu'))
regressor.add(Dense(units=40, kernel_initializer='uniform', activation='relu'))
regressor.add(Dense(units=40, kernel_initializer='uniform', activation='relu'))
regressor.add(Dense(units=40, kernel_initializer='uniform', activation='relu'))
regressor.add(Dense(units=40, kernel_initializer='uniform', activation='relu'))
regressor.add(Dense(units=20, kernel_initializer='uniform', activation='relu'))
regressor.add(Dense(units=10, kernel_initializer='uniform', activation='relu'))
regressor.add(Dense(units=1, kernel_initializer='uniform', activation='relu'))
regressor.compile(optimizer=Adam(lr), loss='mean_squared_error', metrics=['mean_squared_error']) #regresion.
result1 = regressor.fit(X_train, y_train, batch_size=batch_sz, validation_split=0.2, verbose=1, epochs=150)
```

```
r2 = r2_score(y_true=y_test, y_pred=predann)
r2

0.964476644848582

from sklearn.metrics import median_absolute_error

mae = median_absolute_error(y_true=y_test, y_pred=predann)
mae

1.7109472656249949

from sklearn.metrics import mean_squared_error

mse = mean_squared_error(y_true=y_test, y_pred=predann)
mse

32.009153933472604
```

PCA

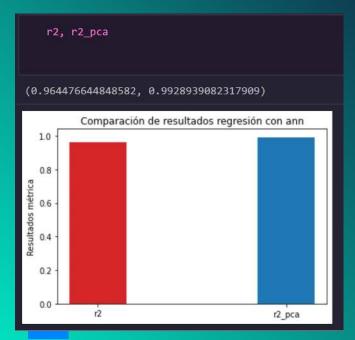
```
pca = PCA(n_components=33)

X_train_pca = pca.fit_transform(X_train)
X_test_pca = pca.transform(X_test)

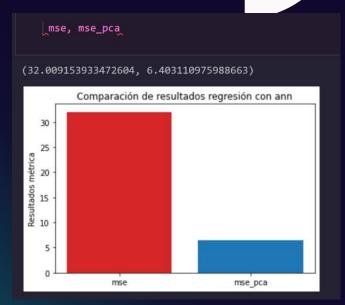
pca.explained_variance_ratio_

array([0.12966831, 0.08571747, 0.07630343, 0.07501033, 0.05258707, 0.05023687, 0.04134245, 0.03418767, 0.02998998, 0.02967268, 0.02923238, 0.02701952, 0.02646267, 0.02618912, 0.02519302, 0.02501044, 0.02475062, 0.02394165, 0.02361968, 0.02282656, 0.02238005, 0.01981667, 0.01956316, 0.0171794, 0.01185166, 0.00944902, 0.00931724, 0.00821287, 0.00766804, 0.00651281, 0.00483017, 0.00185342, 0.00158229])
```

Evaluacion







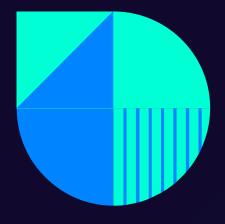
Recomendaciones y Conclusiones

Recomendaciones:

- Agregar gridsearch para optimización de parámetros de red neuranal
- Mayor poder computacional

Conclusiones:

- El resultado confirma que el PCA en conjunto con la red neuronal mejora el resultado significativamente de la variable a predecir.
- El modelo permite optimizar las campañas para la adquisicion de clientes



GRACIAS !!

Tienes alguna duda y/o consulta?

eviau@galileo.edu 15002659@galileo.edu 21000958@galileo.edu







CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon and infographics & images by Freepik

Please keep this slide for attribution