11

May 16, 2023

[1]:
```
import pandas as pd
```

[2]:
```
df = pd.read_csv('cust_seg.csv')
df.head(5)
```

```
/var/folders/_0/nmpfpzw134n12j0c0z6jtrw80000gn/T/ipykernel_6311/3036801543.py:1:
DtypeWarning: Columns (16) have mixed types. Specify dtype option on import or
set low_memory=False.
  df = pd.read_csv('cust_seg.csv')
```

[2]:

	Unnamed: 0	fecha_dato	ncodpers	ind_empleado	pais_residencia	sexo	age	\
0	0	2015-01-28	1375586	N	ES	H	35	
1	1	2015-01-28	1050611	N	ES	V	23	
2	2	2015-01-28	1050612	N	ES	V	23	
3	3	2015-01-28	1050613	N	ES	H	22	
4	4	2015-01-28	1050614	N	ES	V	23	

	fecha_alta	ind_nuevo	antiguedad	…	ind_hip_fin_ult1	ind_plan_fin_ult1	\
0	2015-01-12	0.0	6	…	0	0	
1	2012-08-10	0.0	35	…	0	0	
2	2012-08-10	0.0	35	…	0	0	
3	2012-08-10	0.0	35	…	0	0	
4	2012-08-10	0.0	35	…	0	0	

	ind_pres_fin_ult1	ind_reca_fin_ult1	ind_tjcr_fin_ult1	ind_valo_fin_ult1	\
0	0	0	0	0	
1	0	0	0	0	
2	0	0	0	0	
3	0	0	0	0	
4	0	0	0	0	

	ind_viv_fin_ult1	ind_nomina_ult1	ind_nom_pens_ult1	ind_recibo_ult1
0	0	0.0	0.0	0
1	0	0.0	0.0	0
2	0	0.0	0.0	0
3	0	0.0	0.0	0
4	0	0.0	0.0	0

```
[5 rows x 48 columns]
```

[ ]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.cluster import KMeans


# Preprocess date columns
df['fecha_dato'] = pd.to_datetime(df['fecha_dato'])
df['fecha_alta'] = pd.to_datetime(df['fecha_alta'])
df['month'] = df['fecha_dato'].dt.month
df['year'] = df['fecha_dato'].dt.year
df['alta_month'] = df['fecha_alta'].dt.month
df['alta_year'] = df['fecha_alta'].dt.year

# Drop original date columns
df.drop(['fecha_dato', 'fecha_alta'], axis=1, inplace=True)

# Identify columns with non-numeric values (categorical variables)
categorical_cols = ['ind_empleado', 'pais_residencia', 'sexo']

# Perform one-hot encoding for categorical variables
df_encoded = pd.get_dummies(df, columns=categorical_cols)


# Split the data into features (X) and target variable (y)
X = df.drop('ind_recibo_ult1', axis=1)  # Replace 'target_variable' with the␣
 ↪actual column name
y = df['ind_recibo_ult1']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)

# Model 1: Logistic Regression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
logreg_predictions = logreg.predict(X_test)

# Model 2: Random Forest
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
rf_predictions = rf.predict(X_test)

# Model 3: Gradient Boosting Machines (GBM)
gbm = GradientBoostingClassifier()
```

```
gbm.fit(X_train, y_train)
gbm_predictions = gbm.predict(X_test)

# Model 4: K-Means Clustering
kmeans = KMeans(n_clusters=5)  # Specify the desired number of clusters
kmeans.fit(X)
cluster_labels = kmeans.labels_

# Evaluate the models, perform further analysis, and use the results for␣
 ↪customer segmentation
# You can analyze the predictions, feature importance, clustering labels, etc.␣
 ↪based on your specific requirements
```