



UTPL

La Universidad Católica de Loja

Maestría en Inteligencia Artificial Aplicada

Proyecto Final Grupo 10

Análisis de datos y visualización

Docente: PhD. Janneth Alexandra Chicaiza Espinoza

Integrantes:

- Rafael Guillermo Castro Merino**
- Santiago Andrés Mendieta Carrión**

Número de filas: 500

Número de atributos/variables: 9 variables (3 categóricas y 6 numéricas).

Información de las variables:

customer name: nombre del cliente

customer e-mail: correo electrónico del cliente

country: país de origen y de residencia del cliente

gender: género del cliente (0 para Femenino, 1 para Masculino)

age: edad del cliente

annual Salary: salario anual del cliente

credit card debt: deudas en la tarjeta de crédito del cliente

net worth: patrimonio neto del cliente(activos menos pasivos)

car purchase amount: monto de compra del automóvil que realiza el cliente

Valores nulos: Ninguno

Autor: Mohd Shahnawaz Aadil

URL: <https://www.kaggle.com/datasets/mohdshahnawazaadil/sales-prediction-dataset/data>

- Mediante un análisis inicial descubrimos que el dataset proporcionado no tiene valores nulos, es decir que no existen valores faltantes
- También podemos observar que existen 4 variables categóricas:
Customer name, customer e-mail, country y gender. Gender ya estaba convertida a valores binarios para clasificar el sexo de cliente
- Para las variables numéricas contamos con 5 variables:
Age, annual salary, credit card debt, net worth y car purchase amount

Resumen general de los atributos del dataset:

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 500 entries, 0 to 499  
Data columns (total 9 columns):  
#   Column                Non-Null Count  Dtype  
---  -  
0   customer name         500 non-null    object  
1   customer e-mail       500 non-null    object  
2   country               500 non-null    object  
3   gender                500 non-null    int64  
4   age                   500 non-null    float64  
5   annual Salary         500 non-null    float64  
6   credit card debt      500 non-null    float64  
7   net worth             500 non-null    float64  
8   car purchase amount   500 non-null    float64  
dtypes: float64(5), int64(1), object(3)  
memory usage: 35.3+ KB
```

Automatizacion del EDA Univariado:

skippy summary







Data Summary

dataframe	Values
Number of rows	500
Number of columns	9

Data Types

Column Type	Count
float64	5
string	3
int32	1

number

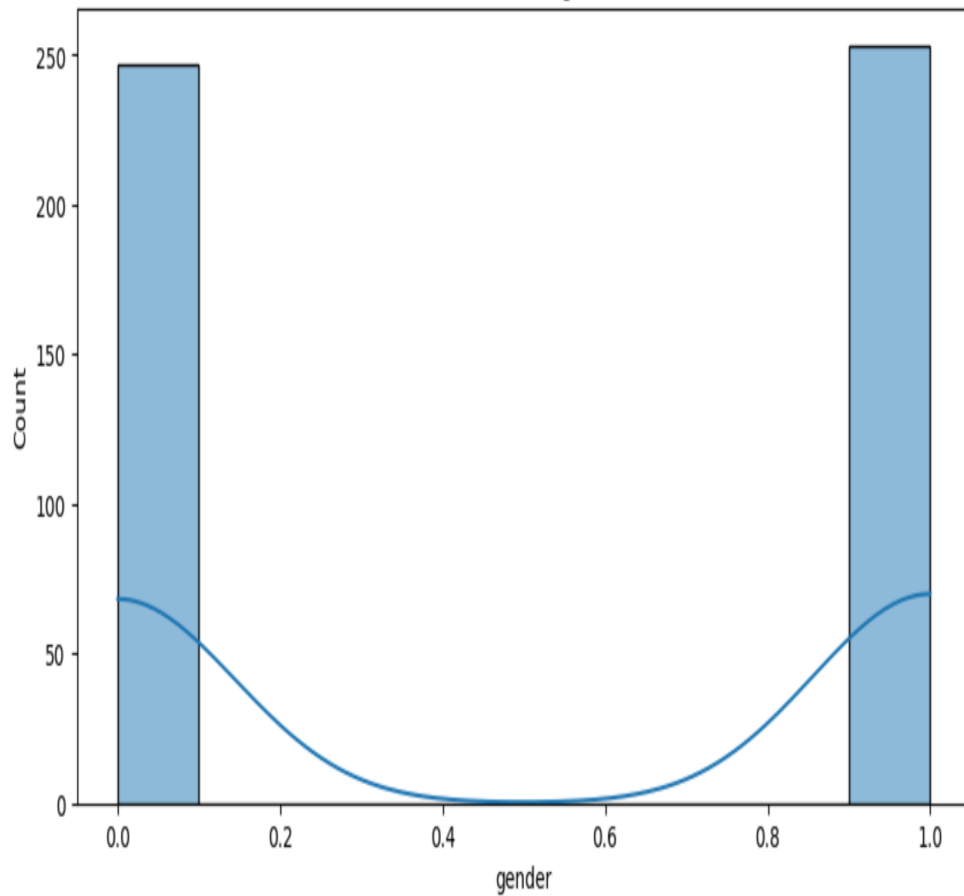
column_name	NA	NA %	mean	sd	p0	p25	p50	p75	p100	hist
gender	0	0	0.506	0.5005	0	0	1	1	1	
age	0	0	46.24	7.979	20	40.95	46.05	51.61	70	
annual Salary	0	0	62130	11700	20000	54390	62920	70120	100000	
credit card debt	0	0	9608	3489	100	7398	9655	11800	20000	
net worth	0	0	431500	173500	20000	299800	426800	557300	1000000	
car purchase amount	0	0	44210	10770	9000	37630	44000	51250	80000	

string

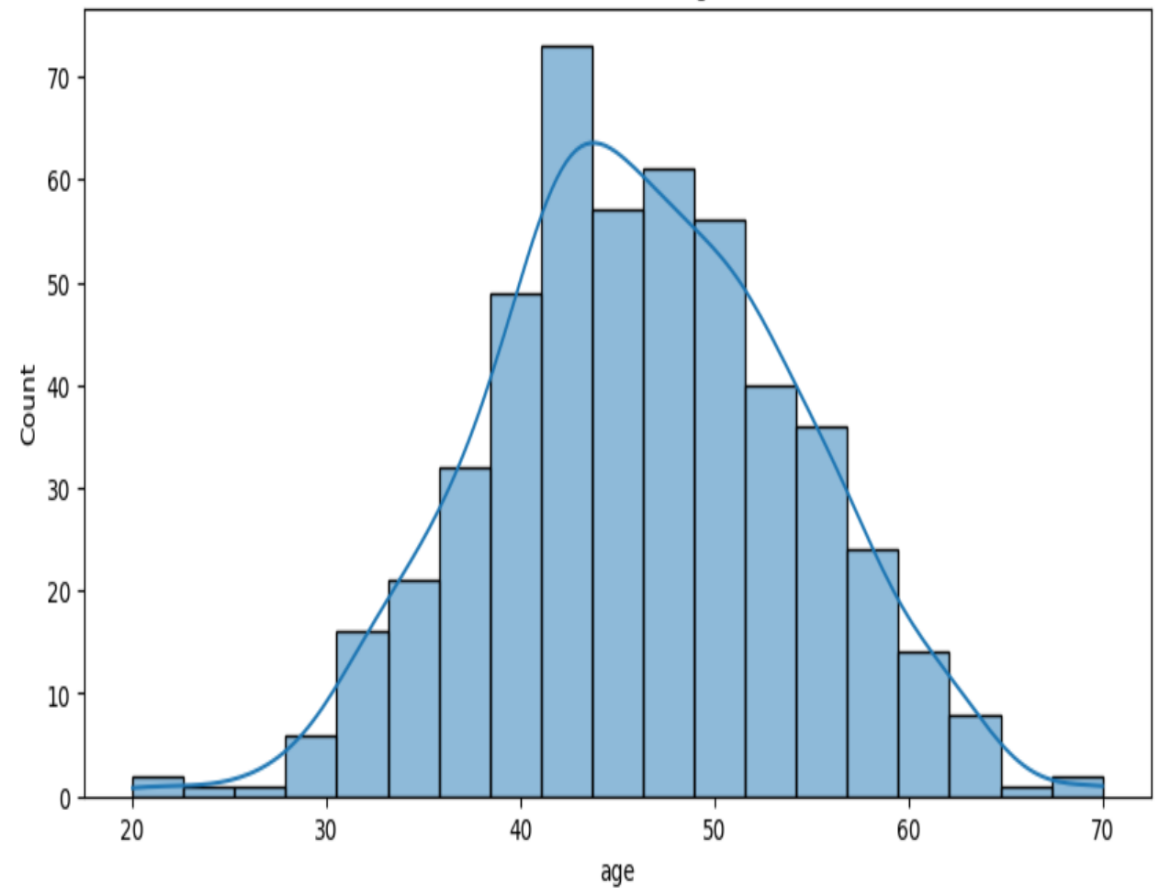
column_name	NA	NA %	words per row	total words
customer name	0	0	2.2	1099
customer e-mail	0	0	1	500
country	0	0	1.5	748

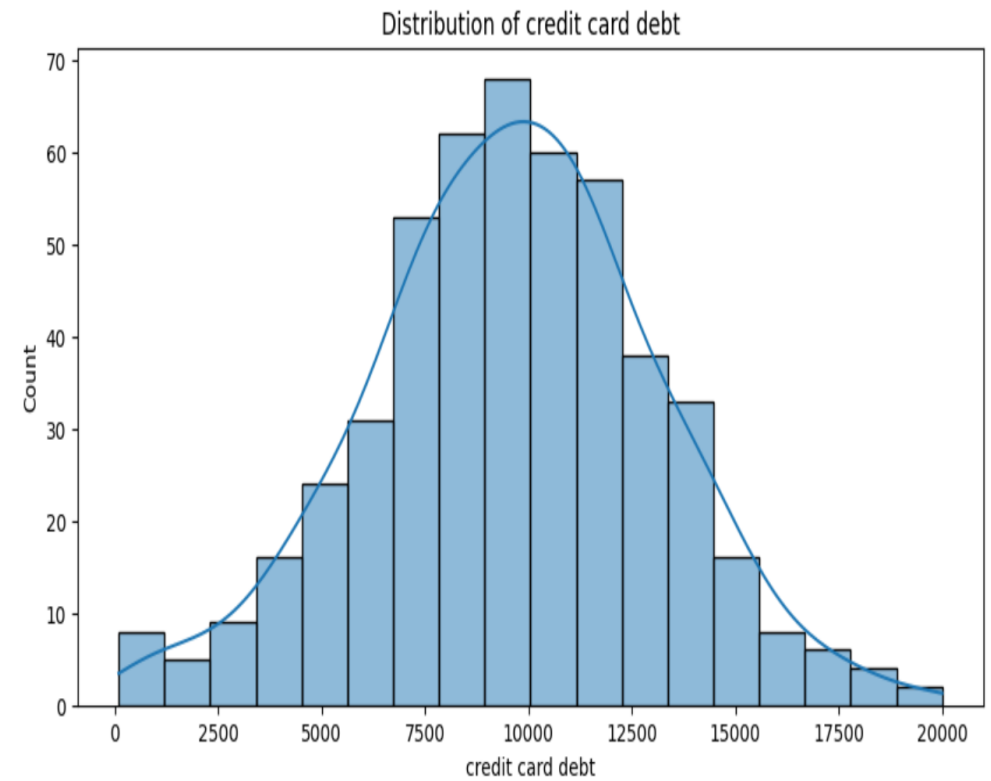
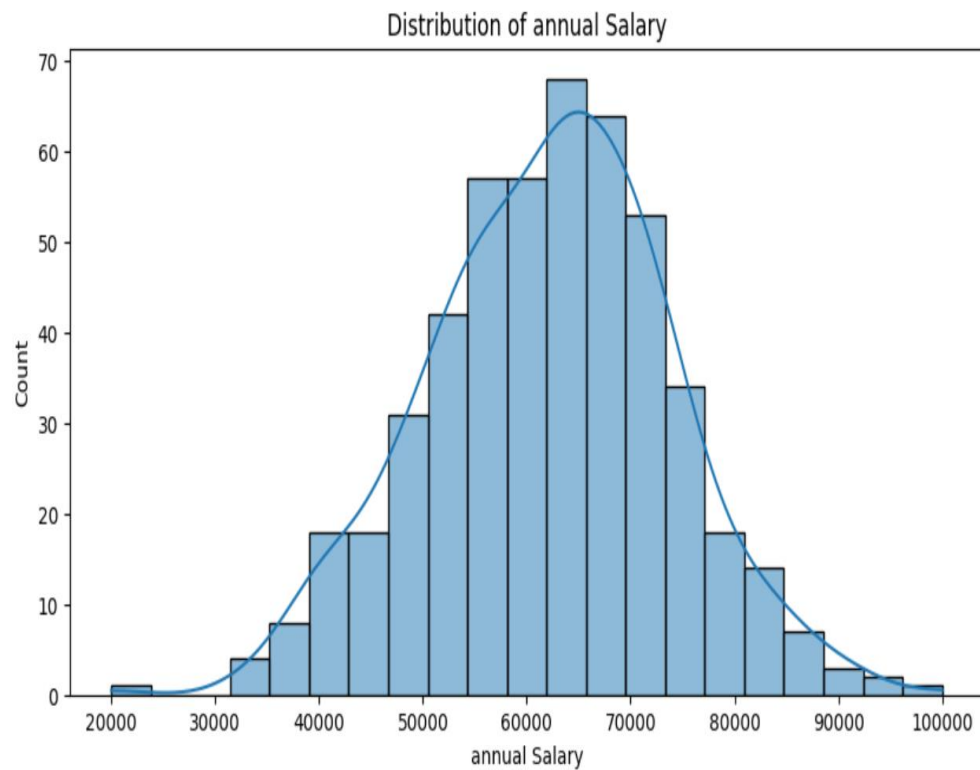
End

Distribution of gender

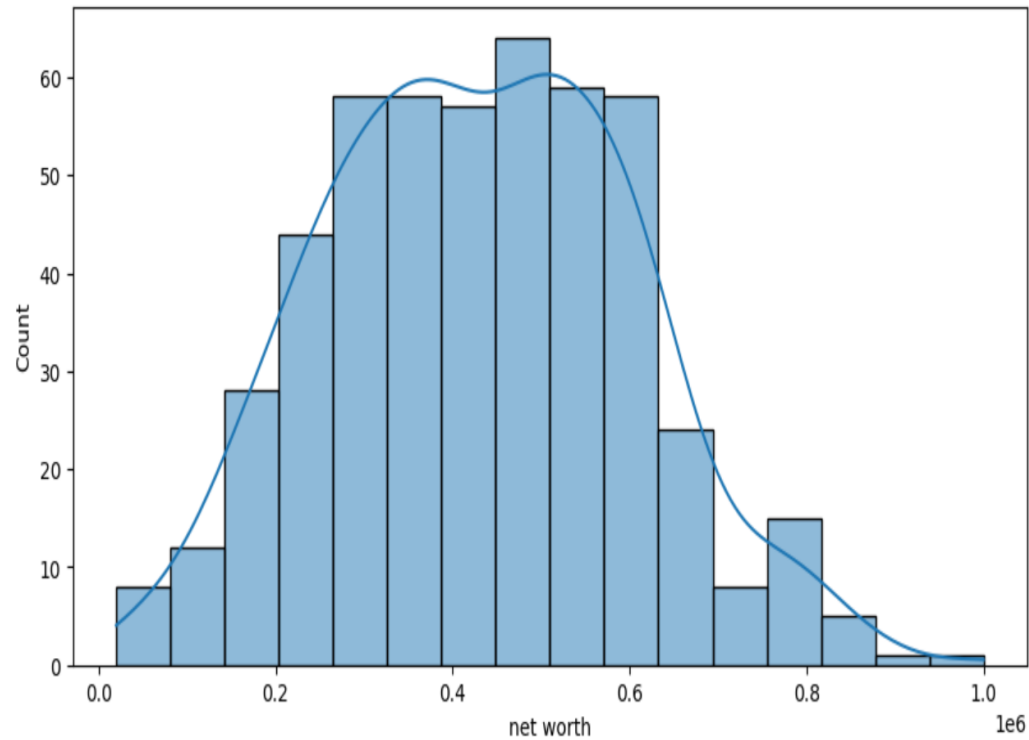


Distribution of age

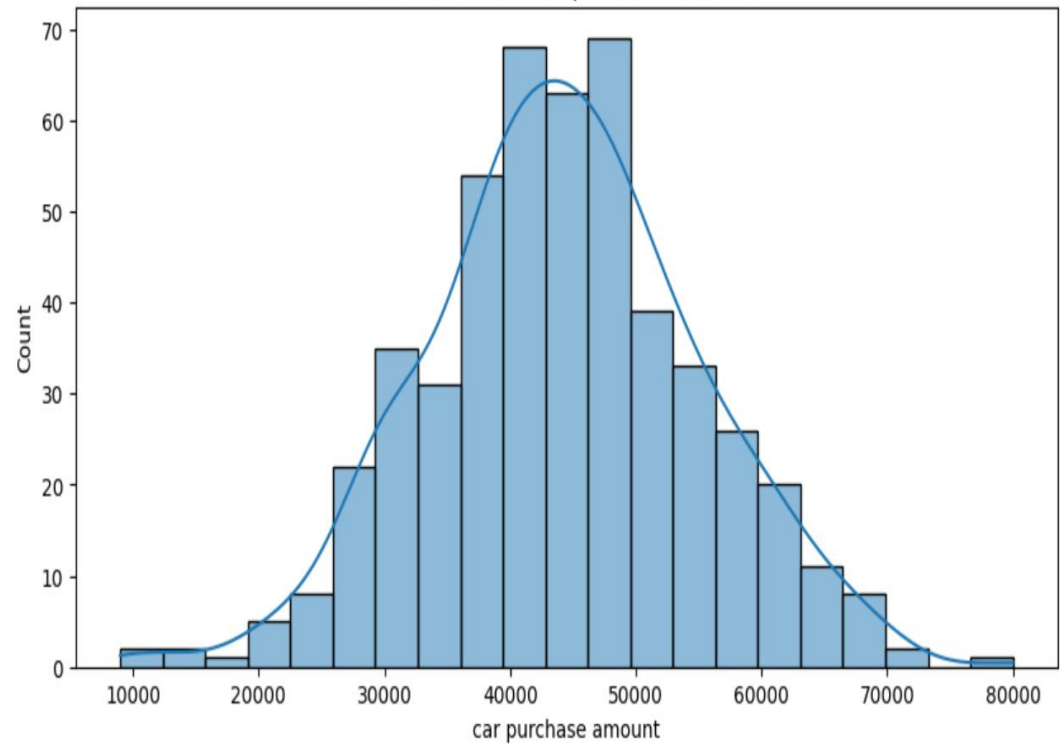


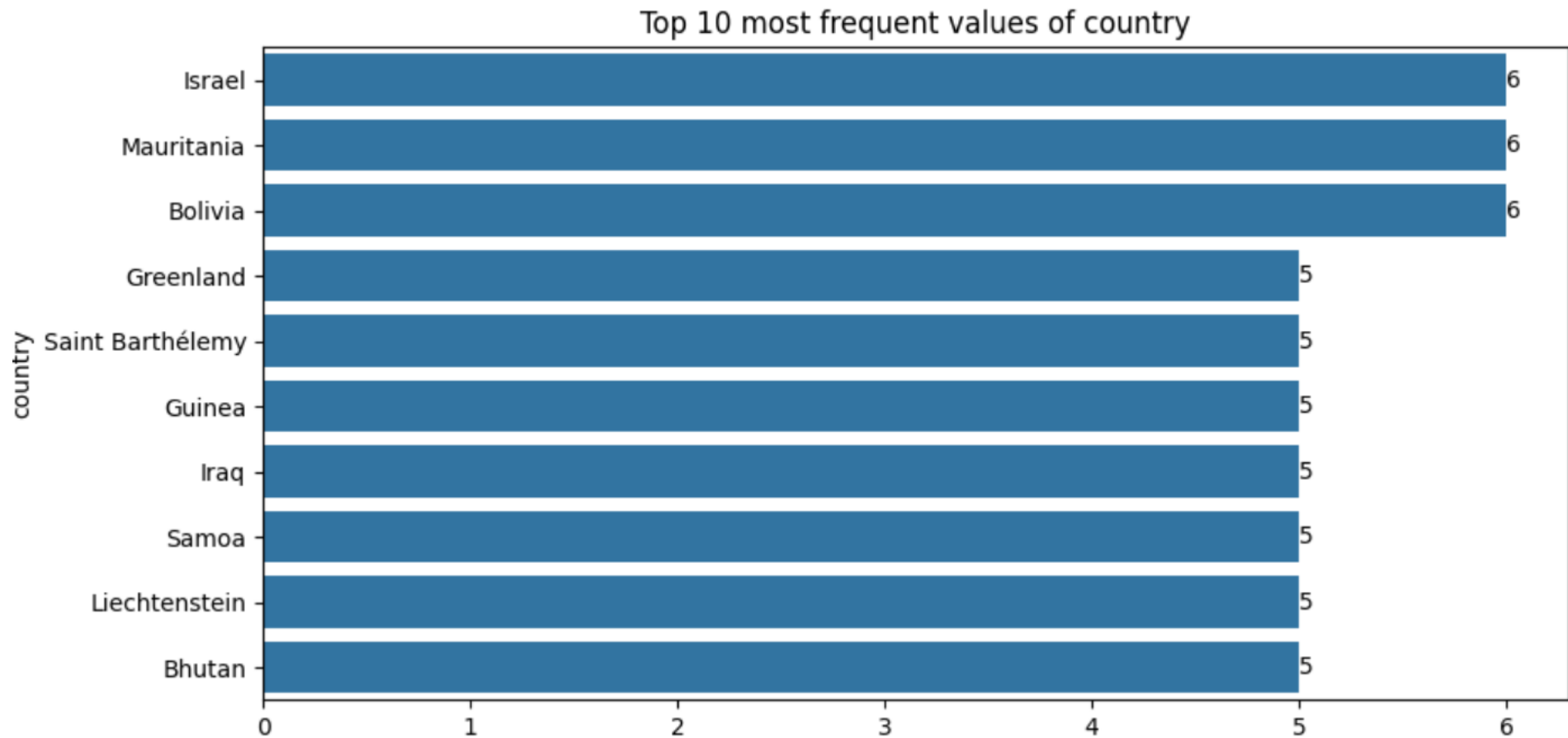


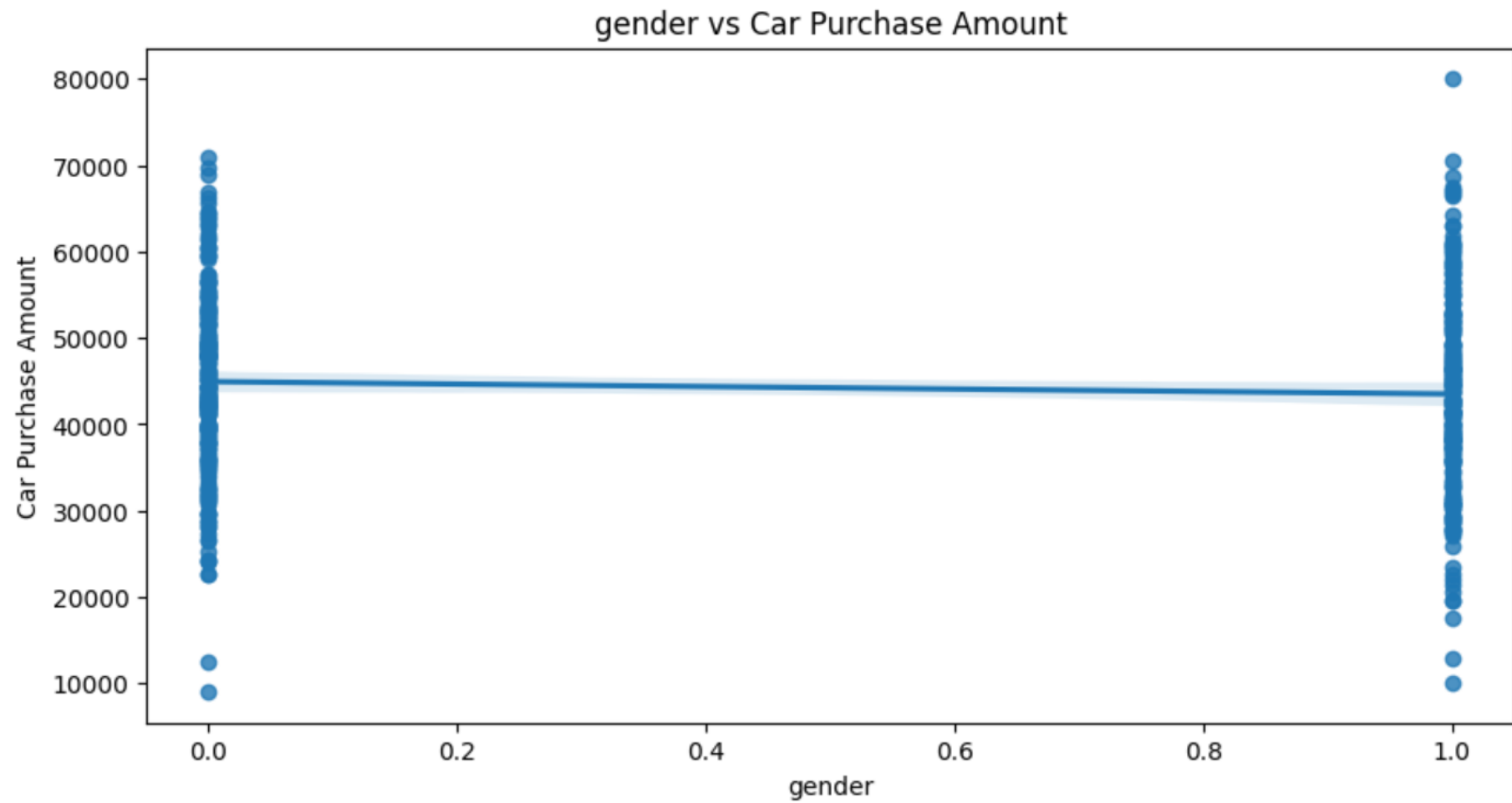
Distribution of net worth

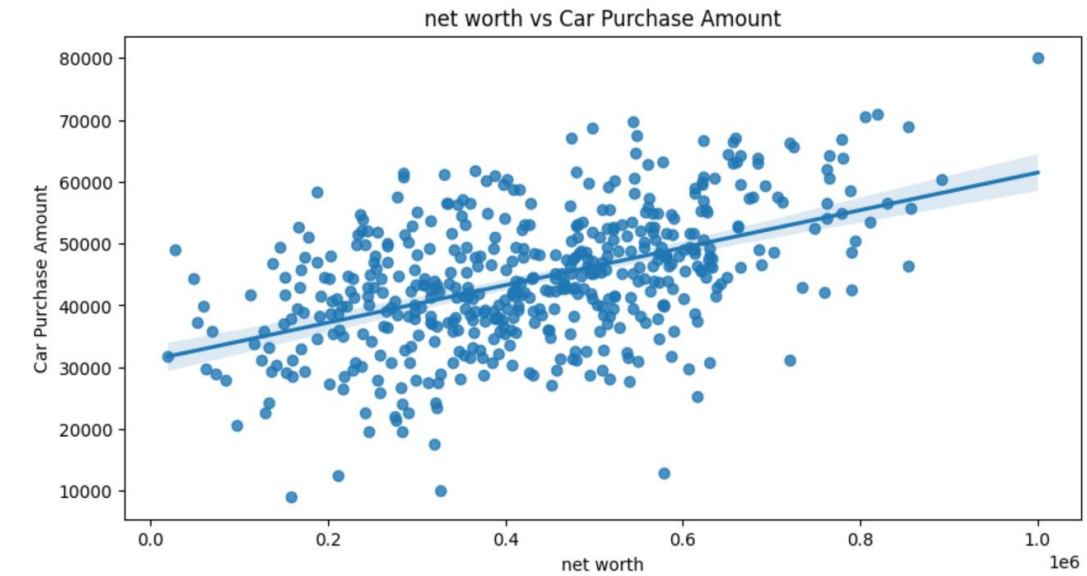
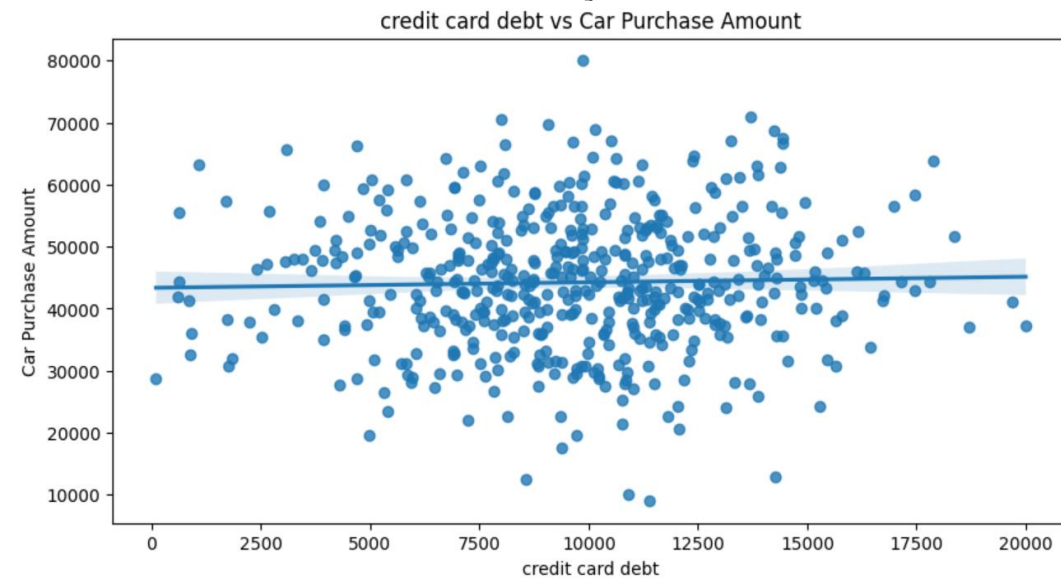
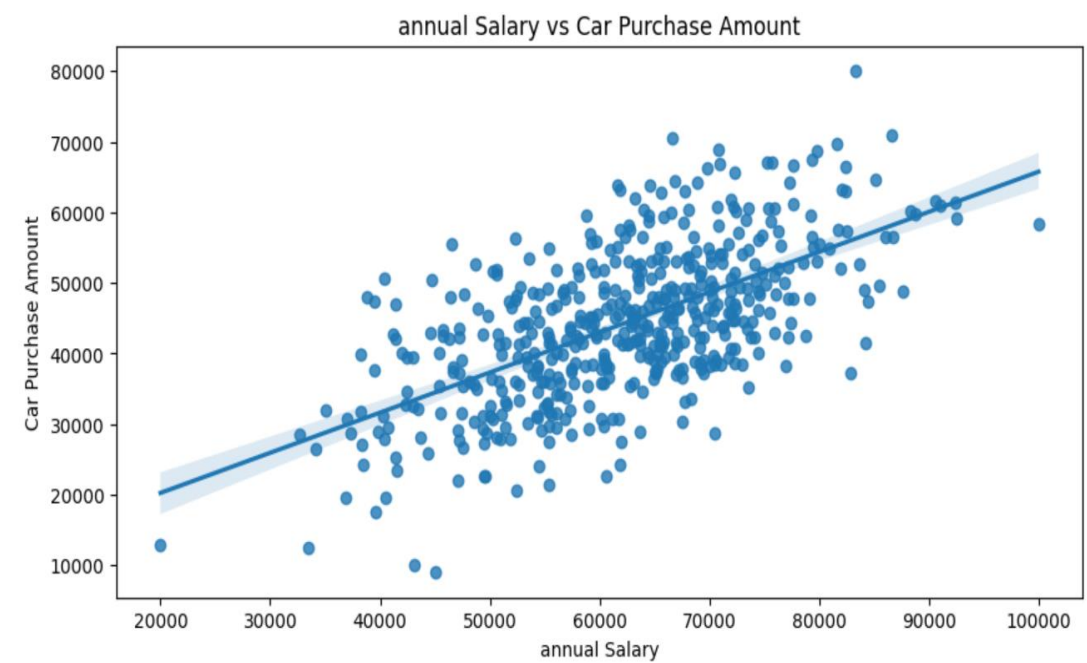
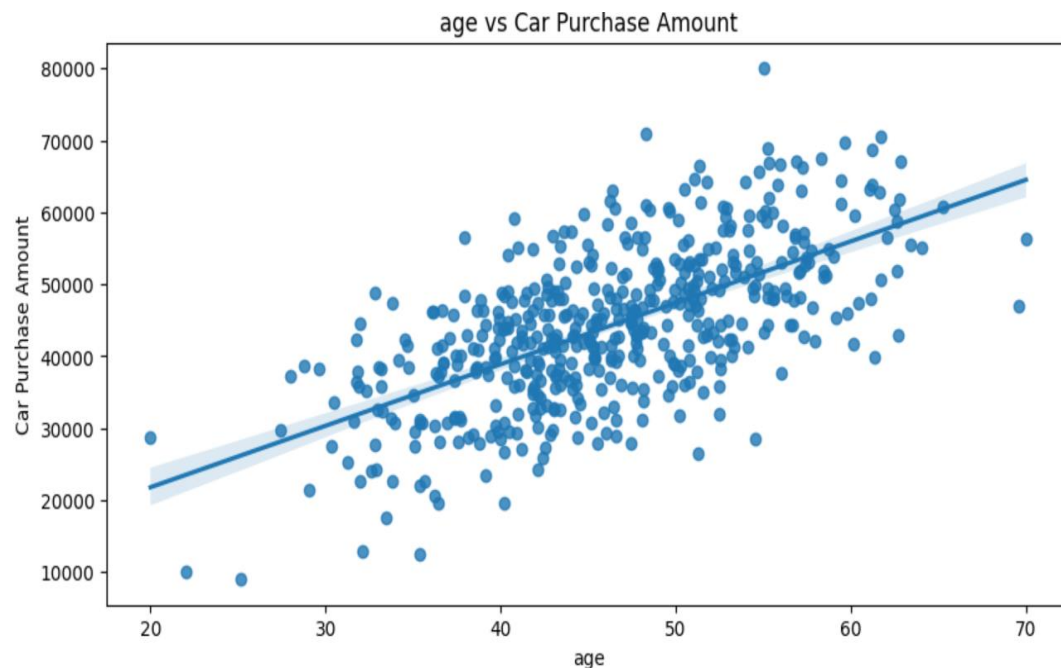


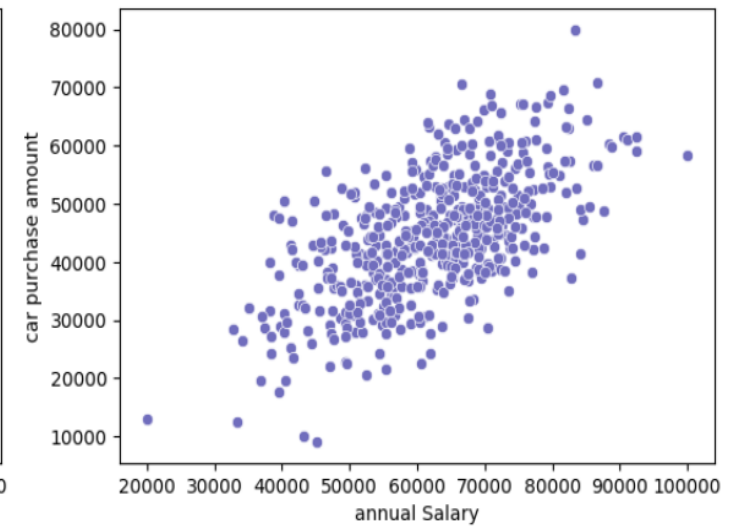
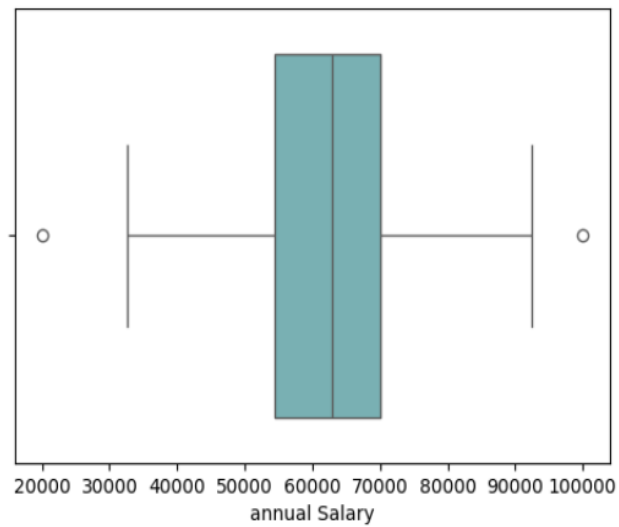
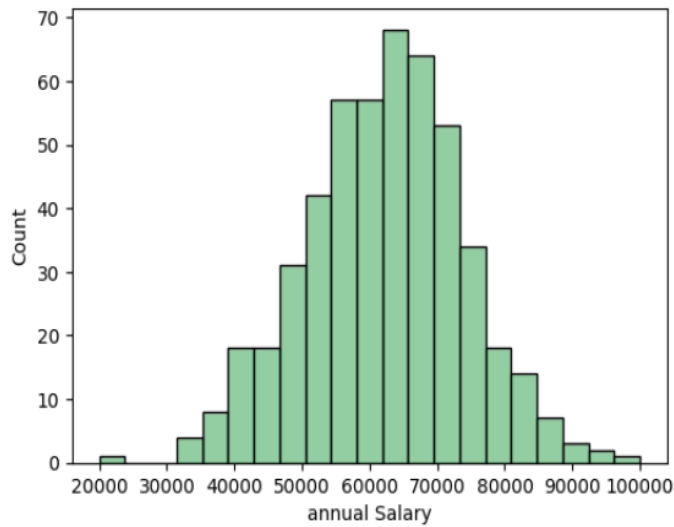
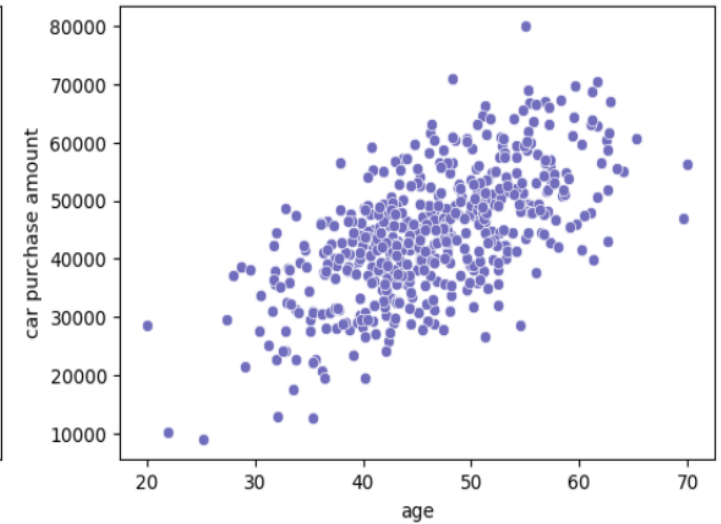
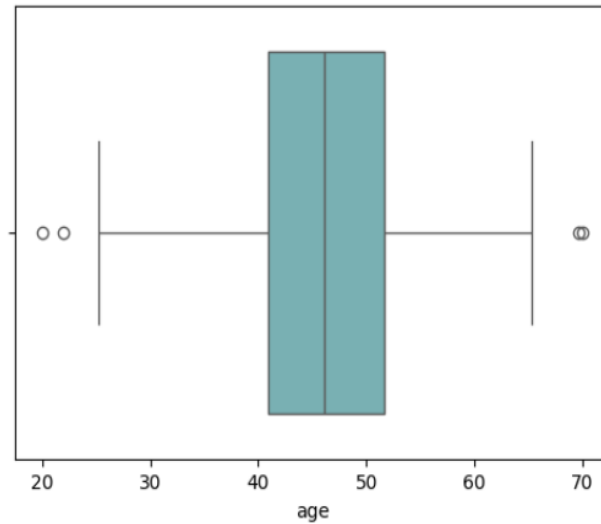
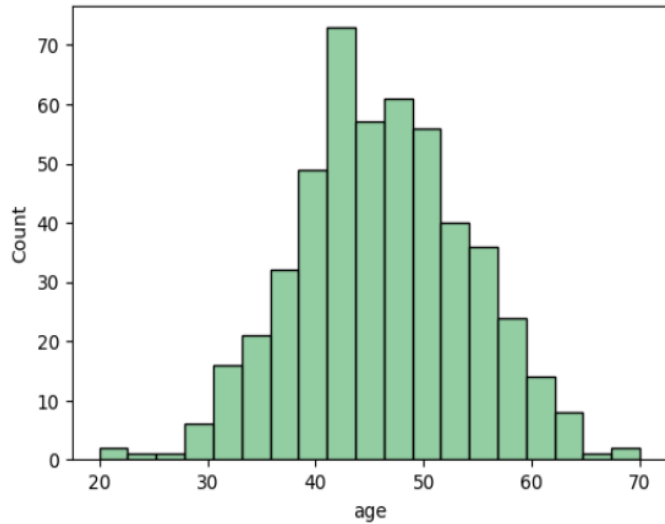
Distribution of car purchase amount

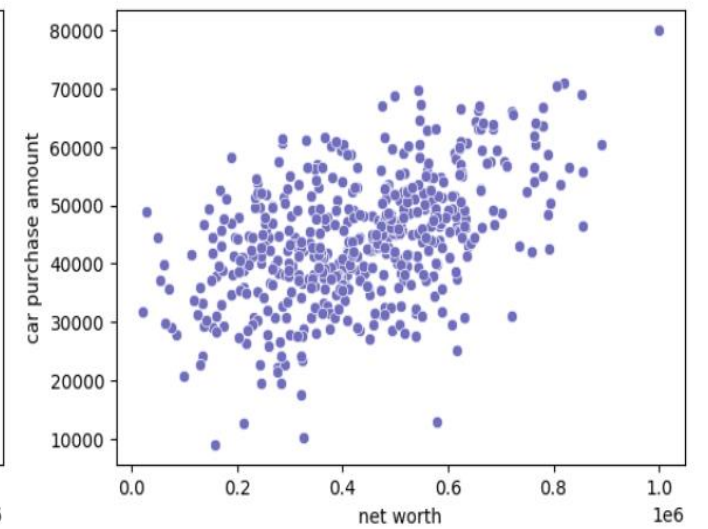
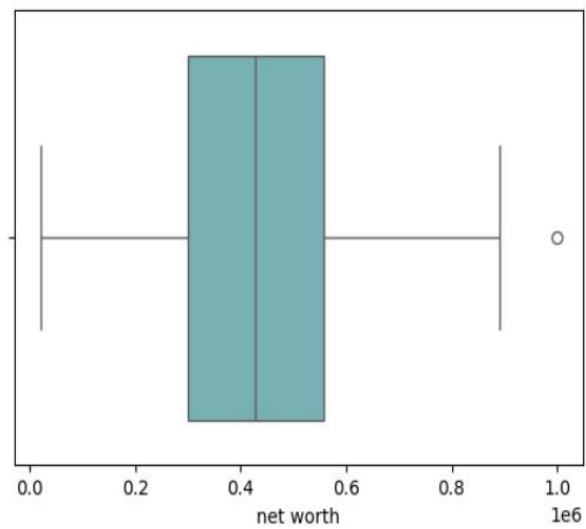
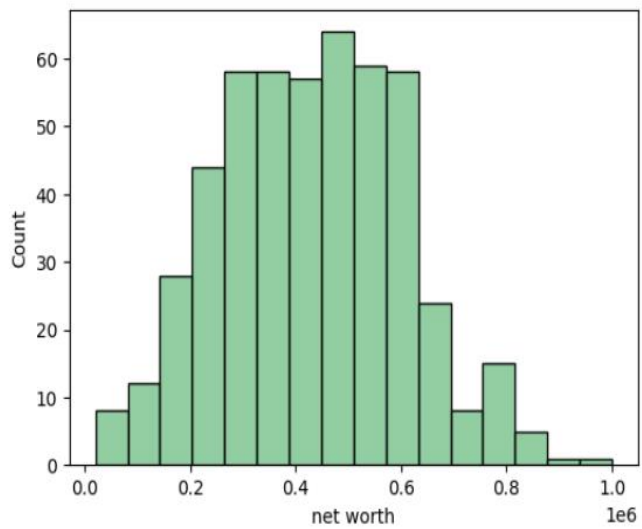
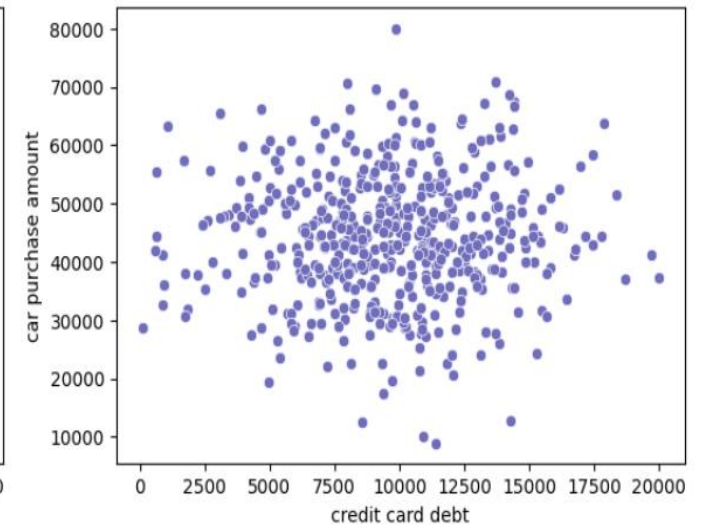
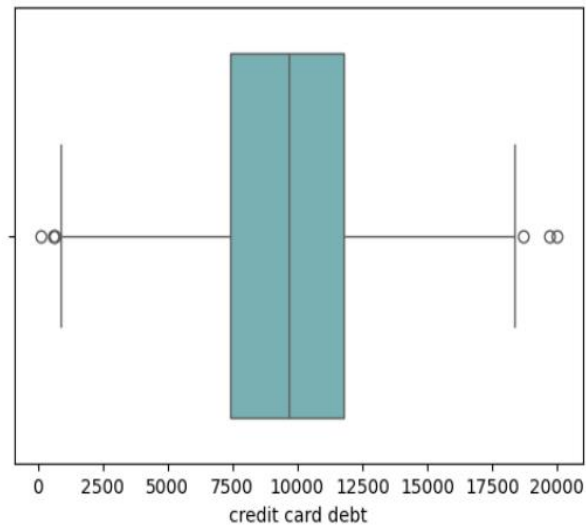
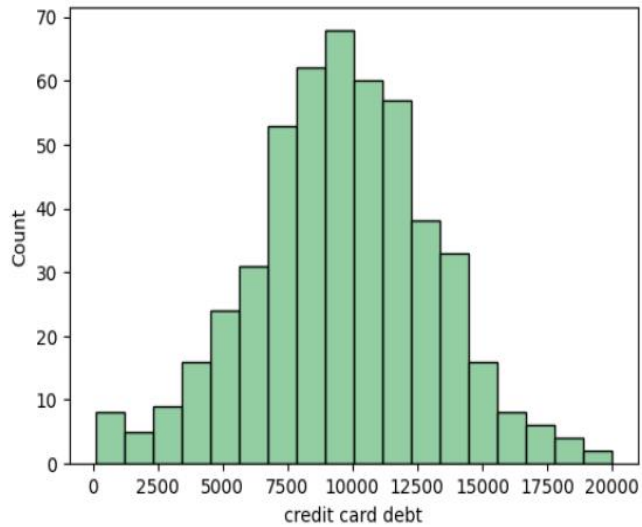


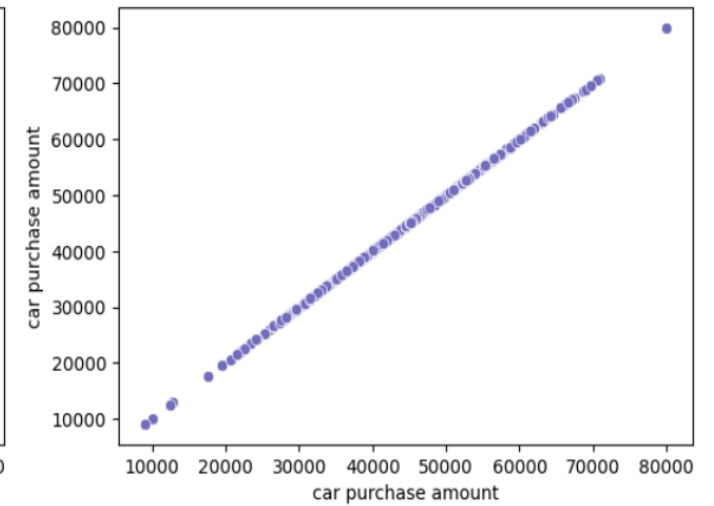
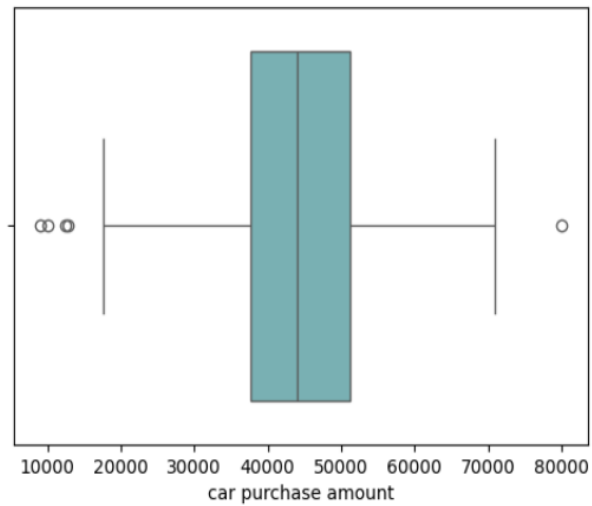
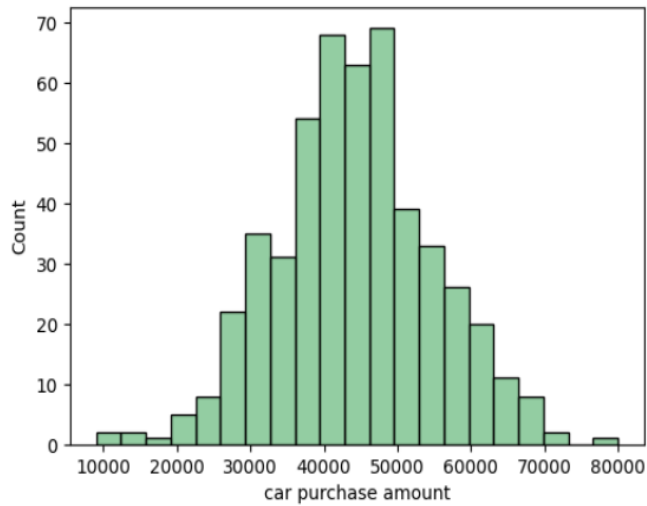


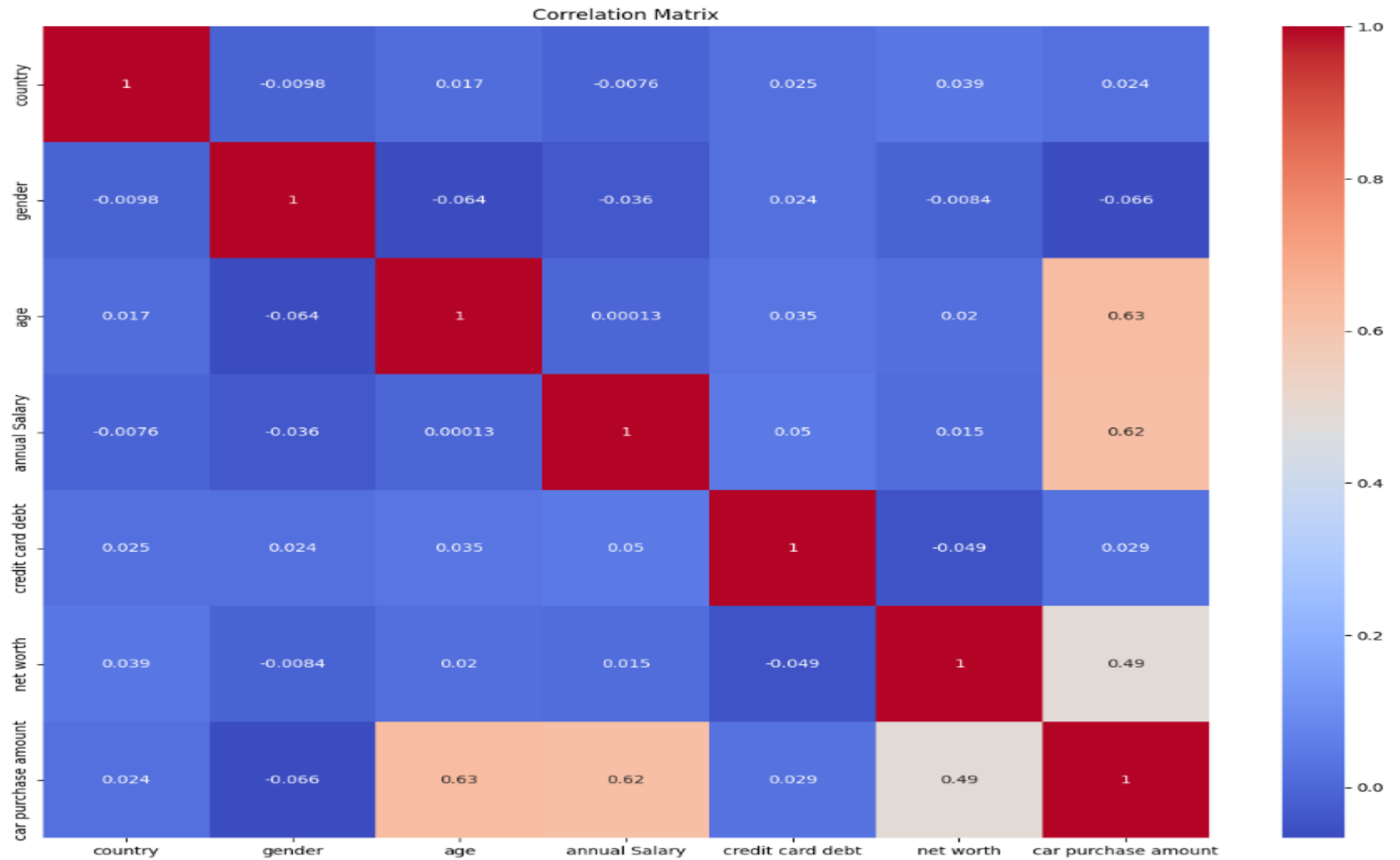












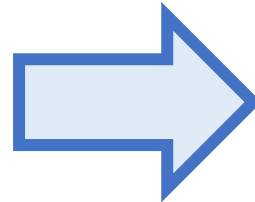
3

Preprocesamiento

- **Eliminación de outliers:** Se eliminan en total 15 filas del dataframe. Lo cual corresponde al 3,09% del mismo.

Cantidad y porcentaje de outliers por cada variable

	Columna	Cantidad Outliers	Porcentaje Outliers
0	age	4	0.8
1	annual Salary	2	0.4
2	credit card debt	7	1.4
3	net worth	1	0.2
4	car purchase amount	5	1.0



Dataframe obtenido

	country	gender	age	annual Salary	credit card debt	net worth	car purchase amount
0	Bulgaria	0	41.851720	62812.09301	11609.380910	238961.2505	35321.45877
1	Belize	0	40.870623	66646.89292	9572.957136	530973.9078	45115.52566
2	Algeria	1	43.152897	53798.55112	11160.355060	638467.1773	42925.70921
3	Cook Islands	1	58.271369	79370.03798	14426.164850	548599.0524	67422.36313
4	Brazil	1	57.313749	59729.15130	5358.712177	560304.0671	55915.46248
...
480	Nepal	0	41.462515	71942.40291	6995.902524	541670.1016	48901.44342
481	Zimbabwe	1	37.642000	56039.49793	12301.456790	360419.0988	31491.41457
482	Philippines	1	53.943497	68888.77805	10611.606860	764531.3203	64147.28888
483	Botswana	1	59.160509	49811.99062	14013.034510	337826.6382	45442.15353
484	marlal	1	46.731152	61370.67766	9391.341628	462946.4924	45107.22566

485 rows × 7 columns



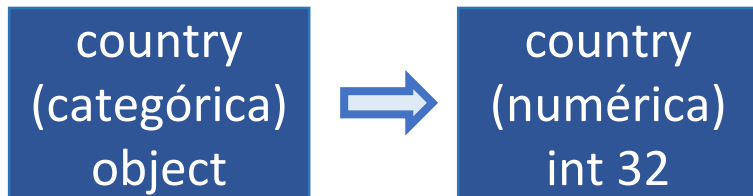
Dimensión del dataframe

3

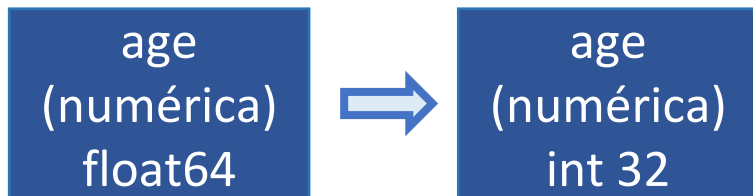
Preprocesamiento

- **Modificación de variables:** Se transforma la variable 'country' en numérica, y se redondea la variable 'age'.

.LabelEncoder()



.round()



Dataframe obtenido

	country	gender	age	annual Salary	credit card debt	net worth	car purchase amount
0	27	0	42	62812.09301	11609.380910	238961.2505	35321.45877
1	17	0	41	66646.89292	9572.957136	530973.9078	45115.52566
2	1	1	43	53798.55112	11160.355060	638467.1773	42925.70921
3	41	1	58	79370.03798	14426.164850	548599.0524	67422.36313
4	26	1	57	59729.15130	5358.712177	560304.0671	55915.46248
...
480	127	0	41	71942.40291	6995.902524	541670.1016	48901.44342
481	207	1	38	56039.49793	12301.456790	360419.0988	31491.41457
482	143	1	54	68888.77805	10611.606860	764531.3203	64147.28888
483	24	1	59	49811.99062	14013.034510	337826.6382	45442.15353
484	208	1	47	61370.67766	9391.341628	462946.4924	45107.22566

Variables modificadas

4

Exploración inicial

- **Importancia de las variables:** Mediante el modelo 'decision tree' se explora la importancia de cada variable.

Entrenamiento

```
#####
# Training
#####
dtr = DecisionTreeRegressor(random_state = 10)
dtr.fit(X_train, y_train)
```

DecisionTreeRegressor

DecisionTreeRegressor(random_state=10)

feature_importances_

```
#####
# Importancia de las variables al momento de crear el modelo
#####
importances_sk = dtr.feature_importances_
feature_importances_df=pd.DataFrame(x_columns, columns=['Columna'])
feature_importances_df['Importancia']=importances_sk
feature_importances_df.sort_values(by='Importancia', ascending=False,
feature_importances_df.reset_index(drop=True, inplace=True)
feature_importances_df
```

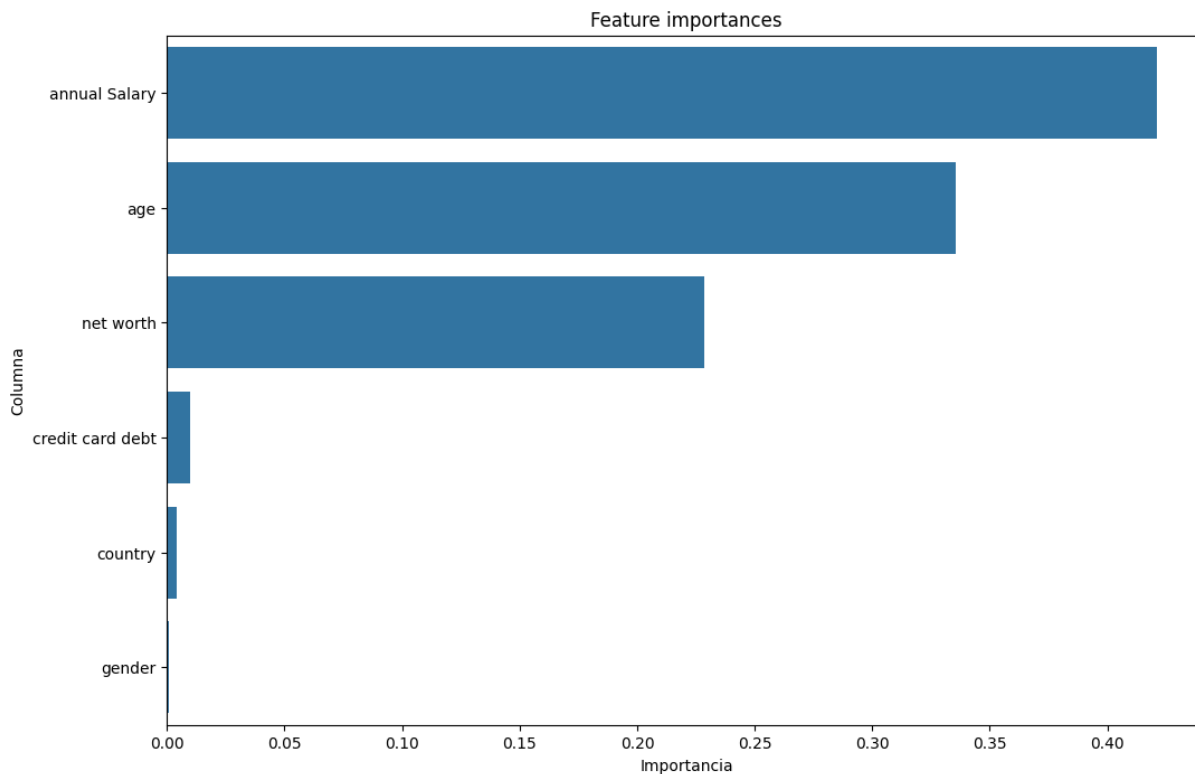
Parámetros

```
#####
# Imprimir parámetros usados por el modelo
#####
print('Max features:', dtr.max_features_)
print('Depth:', dtr.get_depth())
print('N_leaves:', dtr.get_n_leaves())
print('Min samples leaf:', dtr.min_samples_leaf)
print('Min samples split :', dtr.min_samples_split)
```

Max features: 6
Depth: 14
N_leaves: 388
Min samples leaf: 1
Min samples split : 2

	Columna	Importancia
0	annual Salary	0.421085
1	age	0.335691
2	net worth	0.228583
3	credit card debt	0.009735
4	country	0.004277
5	gender	0.000628

Gráfico de barras sobre la importancia de las variables



Variables de mucha importancia

neth
worth

age

annual Salary

Variables de poca importancia

gender

country

credit card
debt

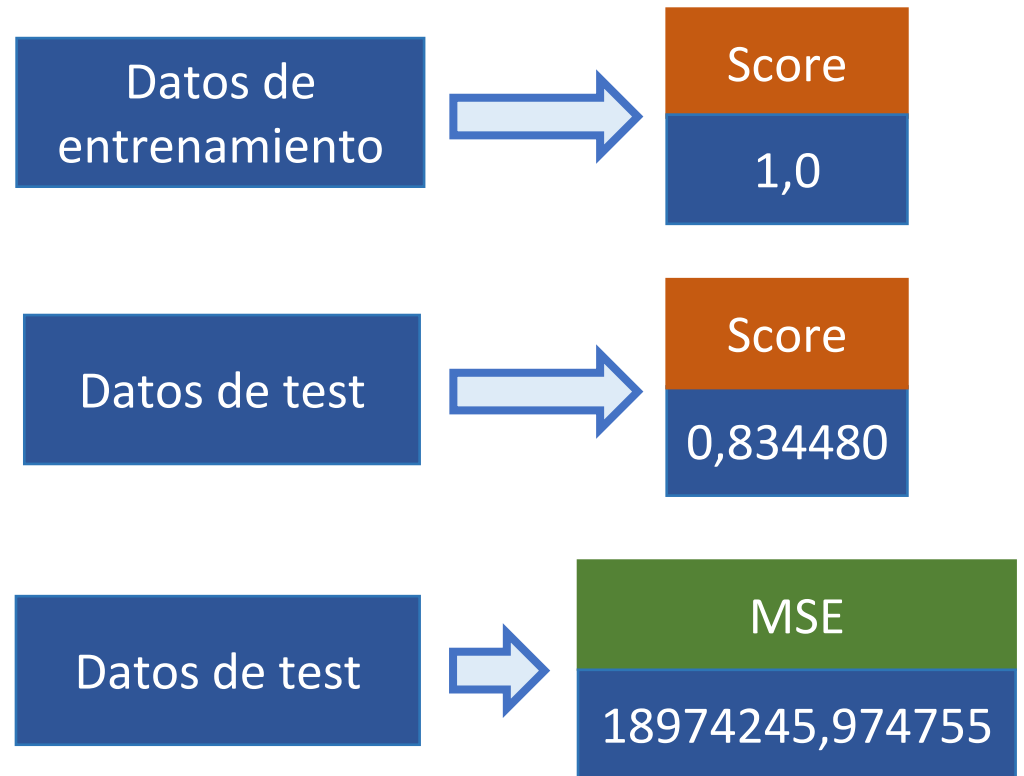
4

Exploración inicial

Evaluación

```
#####  
# Evaluación  
#####  
  
# Train dataset:  
dt_score_train_0 = dtr.score(X_train, y_train) #Precision en  
print('Accuracy_score on train dataset : ', dt_score_train_0)  
  
# Test dataset:  
dt_score_0 = dtr.score(X_test, y_test)  
dt_mse_0 = np.mean((y_test - y_test_predicted)**2)  
  
print("Accuracy_score of model: %f" % dt_score_0) #Precision  
print("Mean Squared Error: %f" % dt_mse_0) #MSE del modelo  
  
Accuracy_score on train dataset : 1.0  
Accuracy_score of model: 0.834480  
Mean Squared Error: 18974245.974755
```

Modelo de exploración: Decision Tree



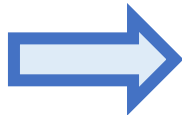
- **Modelo 1 (Decision Tree):** Para el entrenamiento del modelo se utilizó GridSearch y Cross Validation.

Variables predictoras

age

annual Salary

net worth



Variable a predecir

car purchase amount

Entrenamiento

```
#####  
# Training  
#####  
dtr = DecisionTreeRegressor(random_state=10,  
                             max_features=1.0)  
  
params = {'min_samples_split': range(2, 20, 2), #range(start, stop, step)  
          'min_samples_leaf': range(2, 20, 2),  
          'max_depth': range(5, 15)}  
  
grid_search = GridSearchCV(dtr, param_grid=params, cv=10)  
grid_search.fit(X_train, y_train) #Entrenamiento
```

```
GridSearchCV  
└─ estimator: DecisionTreeRegressor  
    └─ DecisionTreeRegressor
```

Parámetros

Max_features: 1

Min_samples_leaf: 2

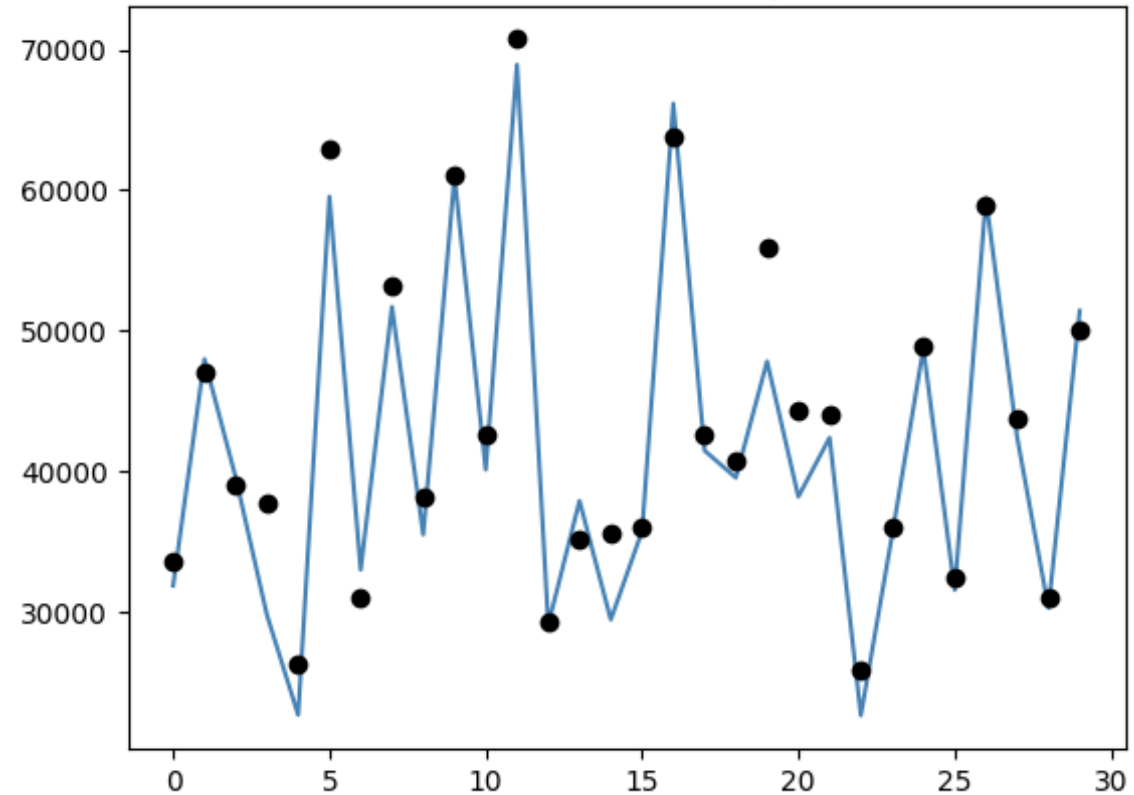
Max_Depth: 14

Min_samples_split: 2

Tabla: valor real vs predicho

	Valor real	Valor predicho
152	31837.22537	33498.401830
380	47970.76767	46972.798880
262	39549.13039	38965.699270
358	29754.66271	37675.933607
312	22630.25982	26183.412870
474	59538.40327	62883.551285
441	32967.20191	30914.966435
199	51683.60859	53159.055930
421	35475.00344	38103.885565
359	60960.83428	61073.324360

Gráfico: valor real vs predicho



- **Modelo 2 (Random Forest):** Para el entrenamiento del modelo se utilizó GridSearch y Cross Validation.

Variables predictoras

age

annual Salary

net worth

Variable a predecir

car purchase amount

Entrenamiento

```
#####  
# Training  
#####  
rf = RandomForestRegressor(n_estimators= 100,  
                           random_state = 10,  
                           max_features=1.0,  
                           oob_score=True)  
  
params = {'max_depth':range(9,13),  
          'min_samples_leaf':range(1,3),  
          'min_samples_split':range(2,10,2)  
}  
grid_search = GridSearchCV(rf,param_grid=params,cv=10)  
grid_search.fit(X_train, y_train) #Entrenamiento
```

```
GridSearchCV ⓘ ?  
└─ estimator: RandomForestRegressor  
    └─ RandomForestRegressor ⓘ
```

Parámetros

Max_features: 1

Min_simples_leaf: 1

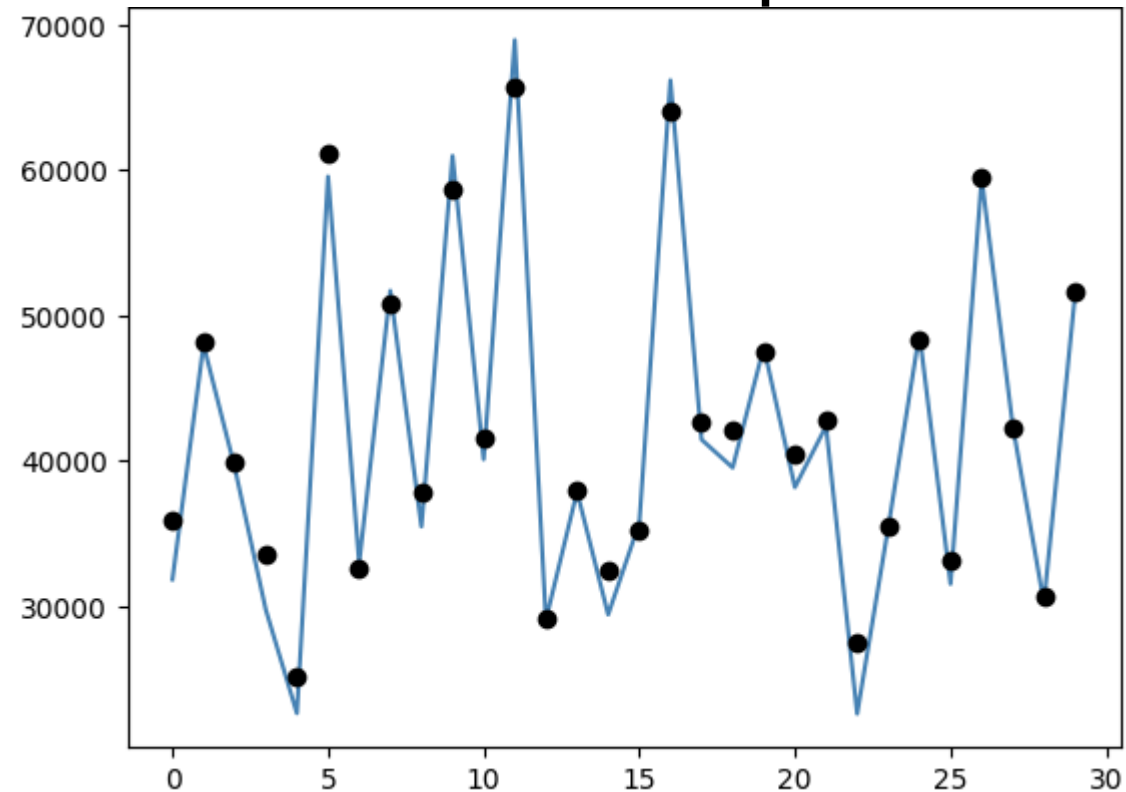
Max_Depth: 12

Min_simples_split: 2

Tabla: valor real vs predecido

	Valor real	Valor predecido
152	31837.22537	35893.424074
380	47970.76767	48144.450085
262	39549.13039	39929.035494
358	29754.66271	33603.334493
312	22630.25982	25184.504242
474	59538.40327	61113.856475
441	32967.20191	32639.844663
199	51683.60859	50724.391144
421	35475.00344	37769.925924
359	60960.83428	58711.548071

Gráfico: valor real vs predecido



6

Evaluación de modelos

- Para la evaluación de los modelos se calcularon las siguientes métricas: precisión del dataset de entrenamiento, precisión del modelo y el MSE del modelo.

Modelo 1: Decision Tree

```
#####  
# Evaluación  
#####  
  
# Train dataset:  
dt_score_train = best_model.score(X_train, y_train)  
print('Accuracy_score on train dataset : ', dt_score_train)  
  
# Test dataset:  
dt_score = best_model.score(X_test, y_test)  
dt_mse = np.mean((y_test - y_test_predicted)**2)  
  
print("Accuracy_score of model: %f" % dt_score)  
print("Mean Squared Error: %f" % dt_mse)
```

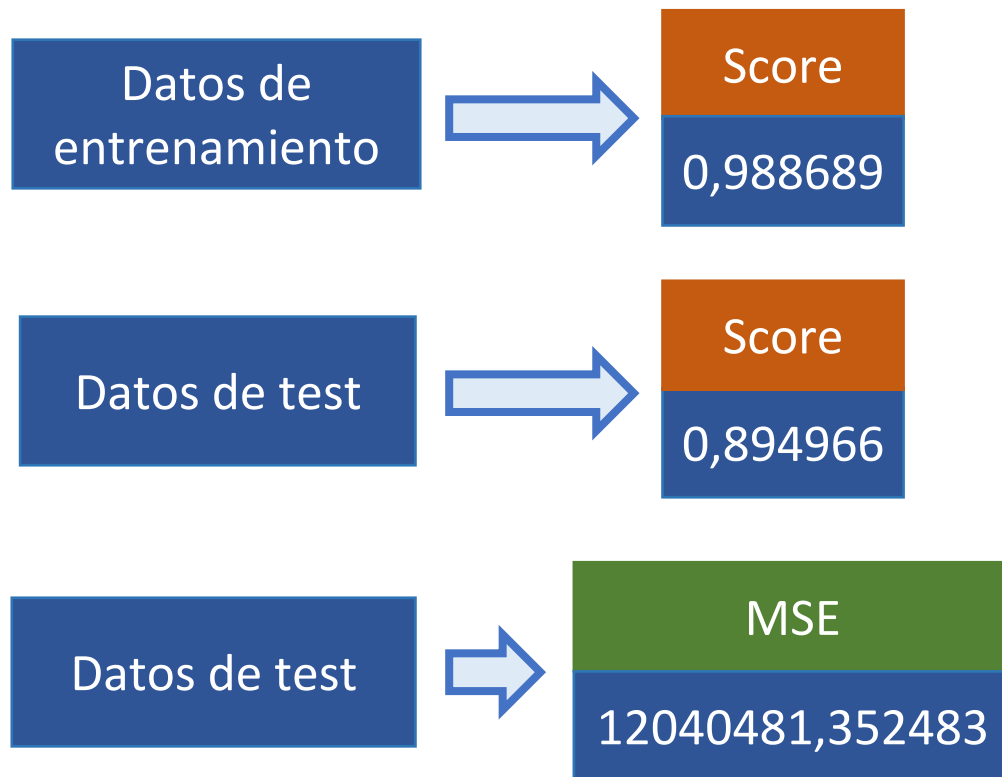
```
Accuracy_score on train dataset : 0.9886890530032699  
Accuracy_score of model: 0.894966  
Mean Squared Error: 12040481.352483
```

Modelo 2: Random Forest

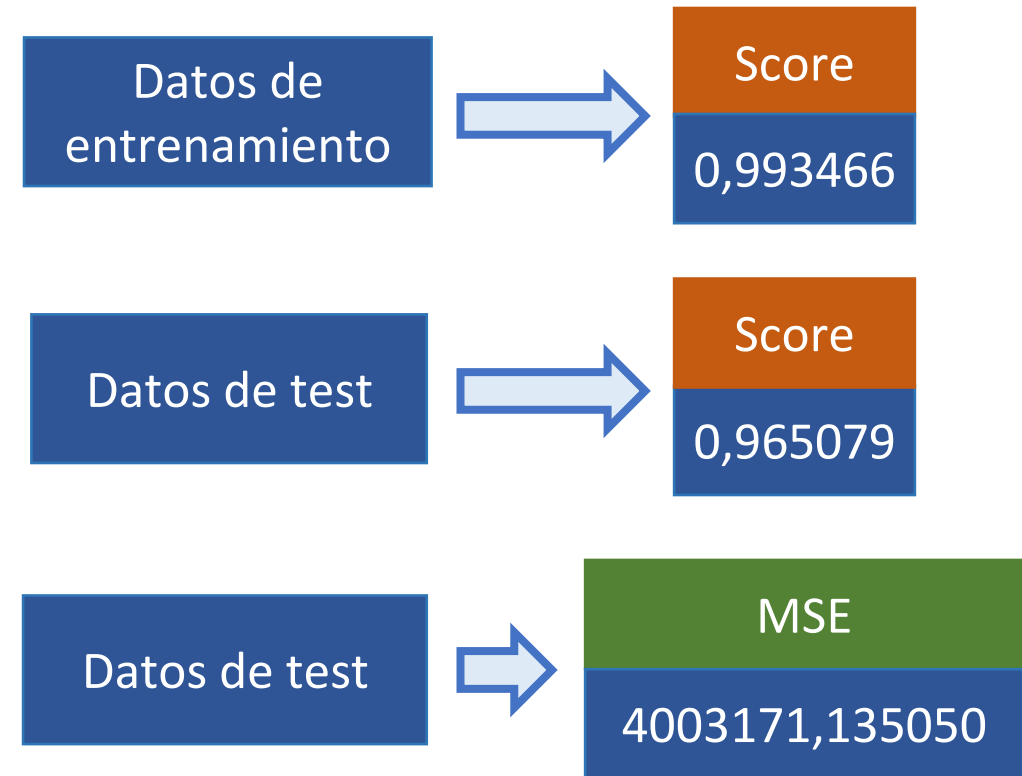
```
#####  
# Evaluación  
#####  
  
# Train dataset:  
rf_score_train = best_model.score(X_train, y_train)  
print('Accuracy_score on train dataset : ', rf_score_train)  
  
# Test dataset:  
rf_score = best_model.score(X_test, y_test)  
rf_mse = np.mean((y_test - y_test_predicted)**2)  
  
print("Accuracy_score of model: %f" % rf_score)  
print("Mean Squared Error: %f" % rf_mse)
```

```
Accuracy_score on train dataset : 0.993466622042856  
Accuracy_score of model: 0.965079  
Mean Squared Error: 4003171.135050
```

Modelo 1: Decision Tree



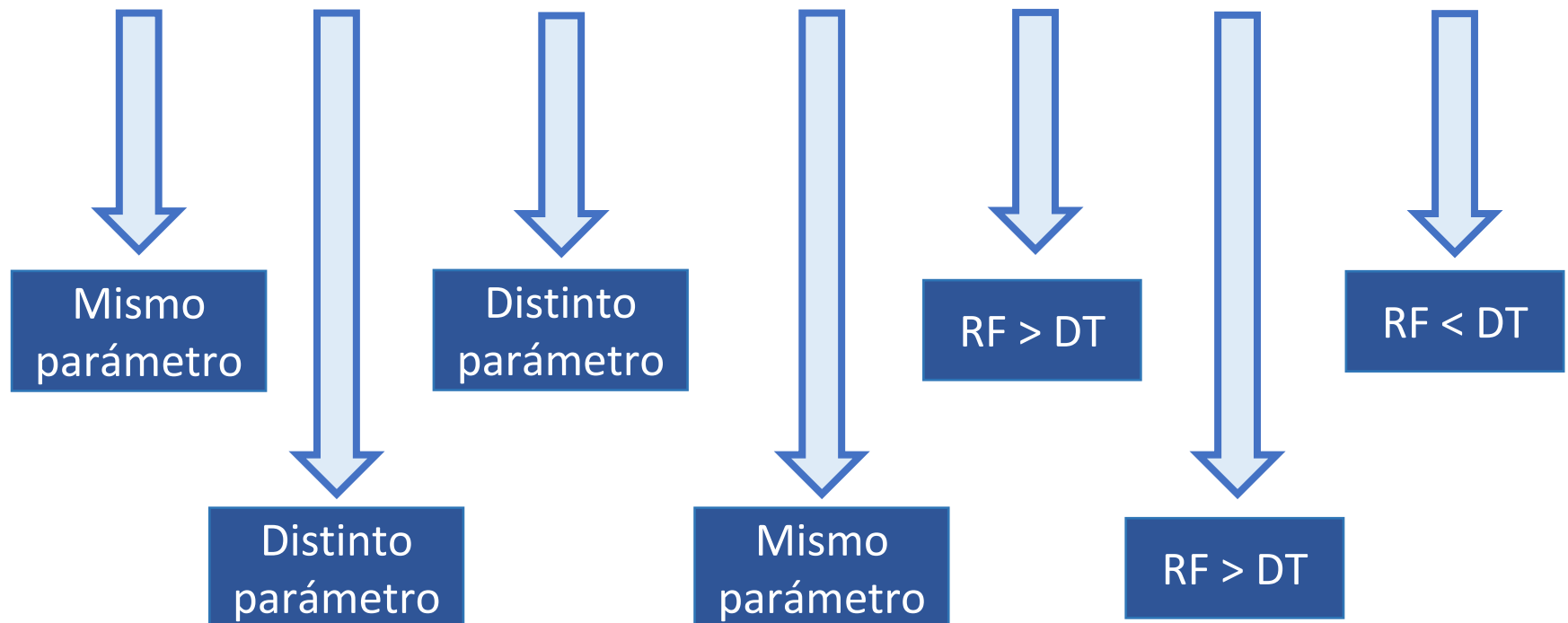
Modelo 2: Random Forest



- Se elaboró un dataframe para exponer los resultados obtenidos de los dos modelos realizados: Decision Tree (DT) y Random Forest (RF).

Dataframe de los resultados

Modelos de evaluación		P1: Max Features	P2: Max Depth	P3: Min Samples Leaf	P4: Min Samples Split	Precisión train	Precisión modelo	MSE
1	Modelo 1: Decision Tree	1	14	2	2	0.988689	0.894966	12040481.352483
2	Modelo 2: Random Forest	1	12	1	2	0.993467	0.965079	4003171.13505



Conclusión 1

- Se concluye que la variable género (gender), el país (country) y la deuda en la tarjeta de crédito (credit card debt) influyen muy poco en el monto de compra de automóviles (car purchase amount).

Conclusión 2

- Se puede afirmar que el patrimonio neto (net worth), la edad (age) y los ingresos anuales (annual Salary) son las características que se deben tomar en cuenta para poder predecir el monto de compra de automóviles (car purchase amount).

Conclusión 3

- Se concluye que el modelo 2, correspondiente a Random Forest es el que mejor predice la variable 'car purchase amount', alcanzando una precisión bastante alta.

Gracias por su
Atención



UTPL
La Universidad Católica de Loja