Imports and loading dataset

```
import numpy as np
import pandas as pd
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeClassifier, DecisionTreeRegressor
print("Importing modules and loading dataset...\n")
```

```
Importing modules and loading dataset...
```

Reading in dataframe

```
df = pd.read_excel("/content/drive/My Drive/Internship/7. ANZ internship/TASK 1/ANZ synthesised tran
df.head()
```

	status	card_present_flag	bpay_biller_code	account	currency	long_lat	txn_descript
0	authorized	1.0	NaN	ACC-1598451071	AUD	153.41 -27.95	
1	authorized	0.0	NaN	ACC-1598451071	AUD	153.41 -27.95	SALES-
2	authorized	1.0	NaN	ACC-1222300524	AUD	151.23 -33.94	
3	authorized	1.0	NaN	ACC-1037050564	AUD	153.10 -27.66	SALES-
4	authorized	1.0	NaN	ACC-1598451071	AUD	153.41 -27.95	SALES-

Modifying data to obtain salaries for each customer

Amount column shows annual salary

```
df_salaries = df[df["txn_description"] == "PAY/SALARY"].groupby("customer_id").mean()
df_salaries.head()
```

	card_present_flag	merchant_code	balance	age	amount
customer_id					
CUS-1005756958	NaN	0.0	4718.665385	53	970.47
CUS-1117979751	NaN	0.0	11957.202857	21	3578.65
CUS-1140341822	NaN	0.0	5841.720000	28	1916.51
CUS-1147642491	NaN	0.0	8813.467692	34	1711.39
CUS-1196156254	NaN	0.0	23845.717143	34	3903.73

```
salaries = []

for customer_id in df["customer_id"]:
    salaries.append(int(df_salaries.loc[customer_id]["amount"]))

df["annual_salary"] = salaries


df_cus = df.groupby("customer_id").mean()
df_cus.head()
```

	card_present_flag	merchant_code	balance	age	amount	annual_salary
customer_id						
CUS-1005756958	0.812500	0.0	2275.852055	53	222.862603	970
CUS-1117979751	0.826923	0.0	9829.929000	21	339.843700	3578
CUS-1140341822	0.815385	0.0	5699.212250	28	212.632500	1916
CUS-						

PREDICTIVE ANALYTICS:

Linear regression

```
N_train = int(len(df_cus)*0.8)
X_train = df_cus.drop("annual_salary", axis=1).iloc[:N_train]
Y_train = df_cus["annual_salary"].iloc[:N_train]
X_test = df_cus.drop("annual_salary", axis=1).iloc[N_train:]
Y_test = df_cus["annual_salary"].iloc[N_train:]


linear_reg = LinearRegression()


linear_reg.fit(X_train, Y_train)
linear_reg.score(X_train, Y_train)
```

```
0.23295376366257825
```

```
linear_reg.predict(X_test)
```

```
array([1993.98473311, 2867.39066481, 1944.95959591, 1806.85984885,
       2226.35045442, 2075.34697175, 1813.02987337, 5388.67435983,
       1902.35351608, 2191.90445145, 1713.48134178, 2854.40519949,
       2094.77781158, 3815.34342881, 2249.92922822, 1768.80816189,
       2095.02988288, 1515.18425875, 1782.72752537, 2481.2898546 ])
```

```
linear_reg.score(X_test, Y_test)
```

```
-0.31694234980747504
```

Decision Tree - Classification and Regression

```
df_cat = df[["txn_description", "gender", "age", "merchant_state", "movement"]]
```

```
pd.get_dummies(df_cat).head()
```

	age	txn_description_INTER BANK	txn_description_PAY/SALARY	txn_description_PAYMENT	txn_descri
0	26	0	0	0	
1	26	0	0	0	
2	38	0	0	0	
3	40	0	0	0	
4	26	0	0	0	

```
N_train = int(len(df)*0.8)
X_train = pd.get_dummies(df_cat).iloc[:N_train]
Y_train = df["annual_salary"].iloc[:N_train]
X_test = pd.get_dummies(df_cat).iloc[N_train:]
Y_test = df["annual_salary"].iloc[N_train:]
```

Classification

```
decision_tree_class = DecisionTreeClassifier()
```

```
decision_tree_class.fit(X_train, Y_train)
decision_tree_class.score(X_train, Y_train)
```

```
0.7882499481004774
```

```
decision_tree_class.predict(X_test)
```

```
array([1013, 1043, 4132, ..., 4054, 1043,  996])
```

```
decision_tree_class.score(X_test, Y_test)
```

```
0.755500207555002
```

Regression

```
decision_tree_reg = DecisionTreeRegressor()
```

```
decision_tree_reg.fit(X_train, Y_train)
decision_tree_reg.score(X_train, Y_train)
```

```
0.7468978726536879
```

```
decision_tree_reg.predict(X_test)
```

```
array([1226.42857143, 1043.        , 4132.        , ..., 3345.04761905,
       1043.        , 1626.        ])
```

```
decision_tree_reg.score(X_test, Y_test)
```

```
0.6729642931765837
```