ANZ Analysis: Correlation between annual salary and age

In this analysis, we build a simple regression model and decision-tree model to predict the annual salary for each customer using the attribute age.

Data Cleansing

We need to remove duplicates and handle the missing values.

```
import pandas as pd
import numpy as np

data = pd.read_excel('~/Downloads/ANZ/ANZ synthesised transaction dataset.xlsx', delimiter='/t')

print (data.head())
print (data.isnull().sum())

data.fillna(data.mean(), inplace=True)
data[['bpay_biller_code', 'merchant_id', 'merchant_suburb', 'merchant_state', 'merchant_long_lat']].fillna(0, inplace=True)

print (data.isnull().sum())

writer = pd.ExcelWriter('cleaned.xlsx')
data.to_excel(writer)
writer.save()print('Dataframe is written succesfully to excel file yo')
```

Code explanation:

1. `data.fillna(data.mean(), inplace=True)`: we fill the missing values using average of each integer/float columns.
2. `data[['bpay_biller_code', 'merchant_id', 'merchant_suburb', 'merchant_state', 'merchant_long_lat']].fillna(0, inplace=True)`: since these columns are string type, we can not find the average value of these. So we fill the missing with `0`.
3. This is optional, I choose to save the cleaned data into excel document.

Feature Engineering

In this part, we need to select from raw data into a form which is easier to interpret.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_excel('~/Downloads/ANZ/cleaned.xlsx', index='customer_id')

data.drop_duplicates(['customer_id'], keep='first', inplace=True)
data['annual_salary'] = data['amount'] *  52

salary = data[data['txn_description'] =='PAY/SALARY']
age_salary = salary[['age', 'annual_salary']]
```

Code explanation:

1. `data.drop_duplicates(['customer_id'], keep='first', inplace=True)`: we need to select distinct customer first.
2. `data['annual_salary'] = data['amount'] * 52`: in this data, customer received certain amount of salary by weekly, so we need to multiply with 52 weeks to get the annual salary.
3. `salary = data[data['txn_description'] =='PAY/SALARY']`: create a subset `age_salary` which contains **PAY/SALARY** category of transactions and slice the column `age` and `annual_salary`.

Simple Regression Model

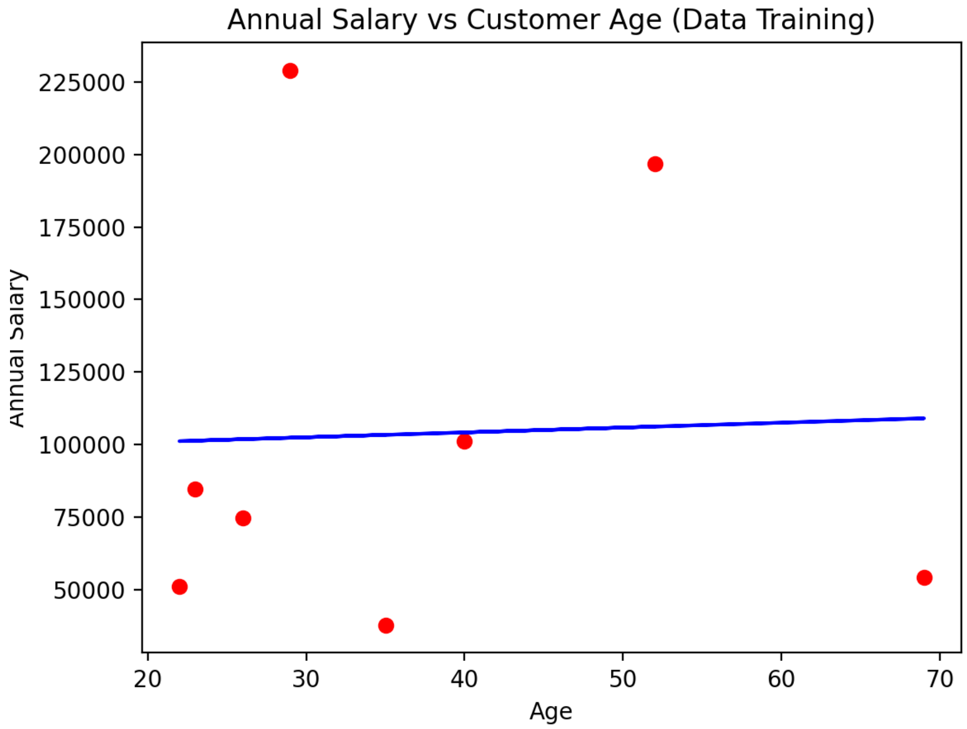
First, we build a simple regression model with the data above.

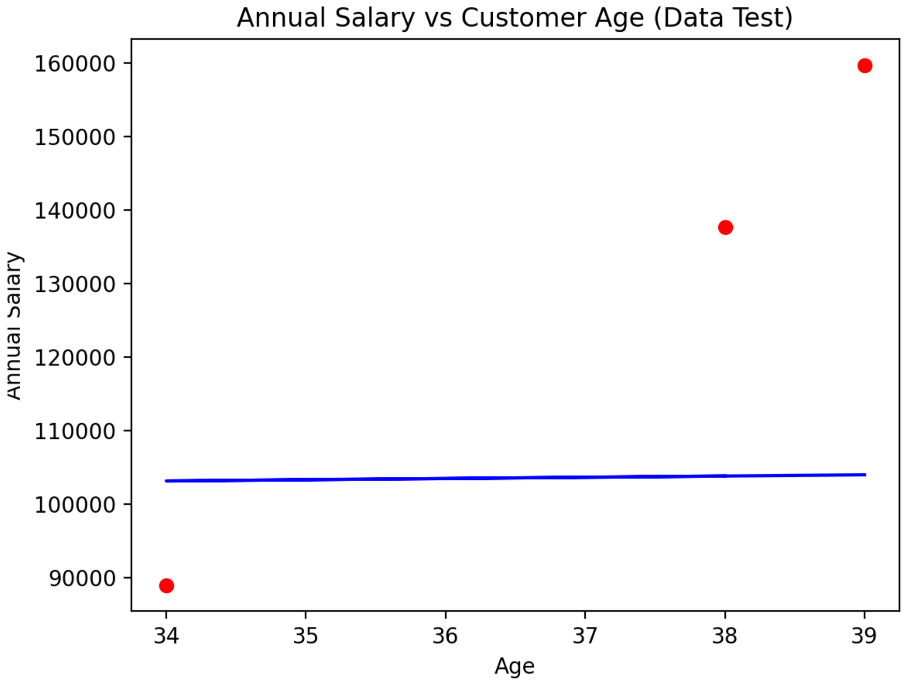
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error

X = age_salary.iloc[:, :-1].values
y = age_salary.iloc[:, 1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size  =  1/5, random_state=0)

regressor = LinearRegression()
regressor.fit(X_train, y_train)

ax_train = plt
ax_train.scatter(X_train, y_train, color='red')
ax_train.plot(X_train, regressor.predict(X_train), color='blue')
ax_train.title('Annual Salary vs Customer Age (Training Set)')
ax_train.xlabel('Age')
ax_train.ylabel('Annual Salary')
ax_train.show()

ax_test = plt
ax_test.scatter(X_test, y_test, color='red')
ax_test.plot(X_test, regressor.predict(X_test), color='blue')
ax_test.title('Annual Salary vs Customer Age (Test Set)')
ax_test.xlabel('Age')
ax_test.ylabel('Annual Salary')
ax_test.show()

#y_pred = regressor.predict([[27]])
y_pred = regressor.predict(X_test)

#evaluate
lin_mse = mean_squared_error(y_pred, y_test)
lin_rmse = np.sqrt(lin_mse)
lin_mae = mean_absolute_error(y_pred, y_test)

print('Slope: ', regressor.coef_)
print('Intercept: ', regressor.intercept_)
print('Liner Regression R squared: %.4f'  % regressor.score(X_test, y_test))
print('Liner Regression RMSE: %.4f'  % lin_rmse)
print('Liner Regression MAE: %.4f'  % lin_mae)
```

Code explanation:

1. `x`: the first column which contains Age array
   `y`: the last column which contains Annual Salary array
2. `test_size=1/5`: we will split our dataset into 2 parts (data training and data test) and the ratio of **data test** compare to dataset is 1/5.
3. `regressor = LinearRegression()`: our training model will implement the Linear Regression.
   `regressor.fit`: in this line, we pass the `X_train` which contains value of **Age** and `y_train` which contains values of **Annual Salary** to form up the model. This is the training process.
4. Now we can use it to predict *any values of X depends on y* or *any values of y depends on X*. First, we want to know Annual Salary for age 27: `y_pred = regressor.predict([[27]])`. Then we apply to all X_test.
5. The model need to evaluate, so we calculate *Root Mean Squared Error* and *Mean Absolute Error*. Here are the values:

```
Slope: [167.53325482]
Intercept: 97475.68457173448
Liner Regression R squared: -0.6984
Liner Regression RMSE: 38498.1724
Liner Regression MAE: 34563.7152
```


Annual Salary vs Customer Age (Data Training)
Annual Salary
225000
200000
175000
150000
125000
100000
75000
50000
20
30
40
50
60
70
Age


Annual Salary vs Customer Age (Data Test)
Annual Salary
160000
150000
140000
130000
120000
110000
100000
90000
34
35
36
37
38
39
Age

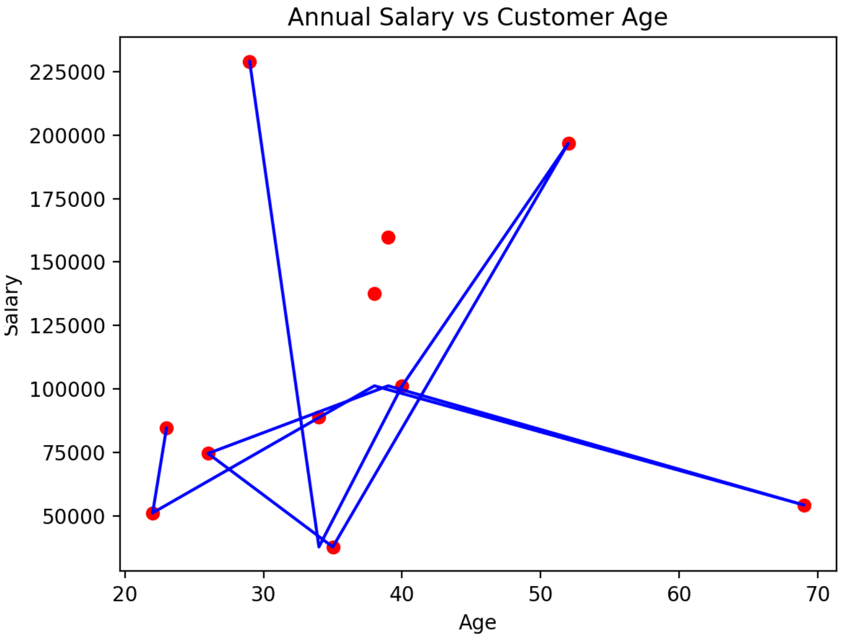
Decision Tree Model

Next we will build Decision Tree model.

```
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_squared_error

X = age_salary.iloc[:, :-1].values
y = age_salary.iloc[:, 1].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size  =  0.20, random_state=0)

regressor = DecisionTreeRegressor()
regressor.fit(X_train, y_train)

#y_pred = regressor.predict([[27]])
y_pred = regressor.predict(X_test)

plt.scatter(X, y, color  =  'red')
plt.plot(X, regressor.predict(X), color  =  'blue')
plt.title('Annual Salary vs Customer Age')
plt.xlabel('Age')
plt.ylabel('Salary')
plt.show()

#evaluate
lin_mse = mean_squared_error(y_pred, y_test)
lin_rmse = np.sqrt(lin_mse)
print('Liner Regression RMSE: %.4f'  % lin_rmse)
```

Code explanation:

1. `regressor = DecisionTreeRegressor()` `regressor.fit(X_train, y_train)` where we use decision tree model.
2. We also have the data test 20% `test_size = 0.20`
3. The RMSE for decision tree: `49587.8135`


Annual Salary vs Customer Age
Salary
Age