# Data Intake Report

Name: Flask deployment
Report date: 30/10/2021
Internship Batch:LISUM04
Version:<1.0>
Data intake by:Betty Wairegi
Data intake reviewer:
Data storage location: C:\3D Objects\

**Tabular data details:**

| | |
|---|---|
| **Total number of observations** | 1000 |
| **Total number of files** | 1 |
| **Total number of features** | 10 |
| **Base format of the file** | .csv |
| **Size of the data** | 49.3KB |

**Note: Replicate same table with file name if you have more than one file.**

**Proposed Approach:**

- The data had no missing values and the task is to deploy a ML model using flask as a web app.

**Data selected.**

```
marketing=pd.read_csv('3D Objects/DirectMarketing.csv')
marketing.head(10)
```

| | Age | Gender | OwnHome | Married | Location | Salary | Children | History | Catalogs | AmountSpent |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Old | Female | Own | Single | Far | 47500 | 0 | High | 6 | 755 |
| 1 | Middle | Male | Rent | Single | Close | 63600 | 0 | High | 6 | 1318 |
| 2 | Young | Female | Rent | Single | Close | 13500 | 0 | Low | 18 | 296 |
| 3 | Middle | Male | Own | Married | Close | 85600 | 1 | High | 18 | 2436 |
| 4 | Middle | Female | Own | Single | Close | 68400 | 0 | High | 12 | 1304 |
| 5 | Young | Male | Own | Married | Close | 30400 | 0 | Low | 6 | 495 |
| 6 | Middle | Female | Rent | Single | Close | 48100 | 0 | Medium | 12 | 782 |
| 7 | Middle | Male | Own | Single | Close | 68400 | 0 | High | 18 | 1155 |
| 8 | Middle | Female | Own | Married | Close | 51900 | 3 | Low | 6 | 158 |
| 9 | Old | Male | Own | Married | Far | 80700 | 0 | NaN | 18 | 3034 |

```
marketing.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   Age          1000 non-null   object
 1   Gender       1000 non-null   object
 2   OwnHome      1000 non-null   object
 3   Married      1000 non-null   object
 4   Location     1000 non-null   object
 5   Salary       1000 non-null   int64 
 6   Children     1000 non-null   int64 
 7   History      697 non-null    object
 8   Catalogs     1000 non-null   int64 
 9   AmountSpent  1000 non-null   int64 
dtypes: int64(4), object(6)
memory usage: 78.2+ KB
```

## Encoding of data

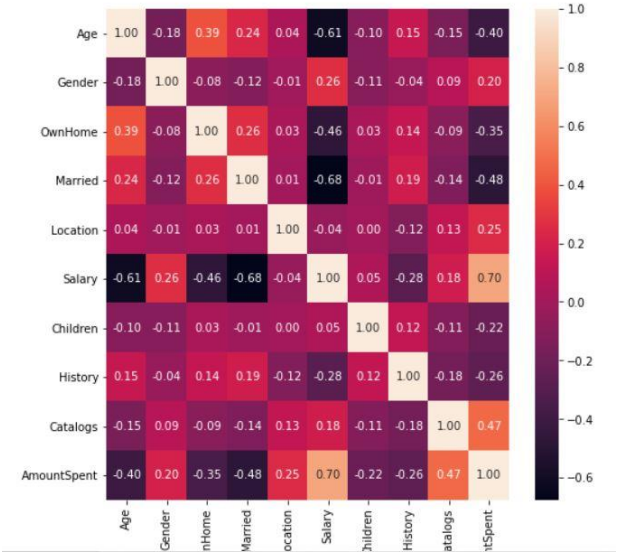
The data was encoded particularly the categorical variables to a binary format using LabelEncoding.

```
marketing.head(10)
```

| | Age | Gender | OwnHome | Married | Location | Salary | Children | History | Catalogs | AmountSpent |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 0 | 1 | 1 | 47500 | 0 | 0 | 6 | 755 |
| **1** | 0 | 1 | 1 | 1 | 0 | 63600 | 0 | 0 | 6 | 1318 |
| **2** | 2 | 0 | 1 | 1 | 0 | 13500 | 0 | 1 | 18 | 296 |
| **3** | 0 | 1 | 0 | 0 | 0 | 85600 | 1 | 0 | 18 | 2436 |
| **4** | 0 | 0 | 0 | 1 | 0 | 68400 | 0 | 0 | 12 | 1304 |
| **5** | 2 | 1 | 0 | 0 | 0 | 30400 | 0 | 1 | 6 | 495 |
| **6** | 0 | 0 | 1 | 1 | 0 | 48100 | 0 | 2 | 12 | 782 |
| **7** | 0 | 1 | 0 | 1 | 0 | 68400 | 0 | 0 | 18 | 1155 |
| **8** | 0 | 0 | 0 | 0 | 0 | 51900 | 3 | 1 | 6 | 158 |
| **9** | 1 | 1 | 0 | 0 | 1 | 80700 | 0 | 3 | 18 | 3034 |

## Correlation of variables with target variable

```
#correlation map
plt.figure(figsize=(8,8))
sns.heatmap(marketing.corr(),annot=True,fmt='.2f');
```



## Spliting data into train and test

```
# target and data to be used
X=marketing[['Salary','OwnHome','Gender','Children','History','Catalogs','Location','Married','Age']]
Y=marketing['AmountSpent']
```

```
from sklearn.model_selection import train_test_split
```

```
train_x,test_x,train_y,test_y=train_test_split(X,Y,train_size=0.7)
train_x.shape
```

```
(700, 9)
```

## Creating model

```
from sklearn.linear_model import LinearRegression
regression=LinearRegression()
```

```
regression_fit=regression.fit(train_x,train_y)
```

```
predict_regression=regression.predict(test_x)
```

## Saving the model

```
import joblib
joblib.dump(predict_regression,'linear.pkl')
linearmodel=joblib.load(open('linear.pkl','rb'))
```

## Deploying on flask

```
from flask import request
app=Flask(__name__,template_folder='template')
@app.route('/')
def home():
     return render_template('flaskdummy.html')

@app.route('/predict',methods=['POST','GET'])
def prediction():
    if request.method == 'POST':
        g=request.args.get('gender')
        o=request.args.get('own')
        l=request.args.get('location')
        s=request.args.get('status')
        sa=request.args.get('salary')
        a=request.args.get('age')
        c=request.args.get('cat')
        ch=request.args.get('child')
        b=request.args.get('buy')

        test = [g,o,l,s,sa,a,c,ch,b]
        predicted = linearmodel.predict(test)
        return render_template('flaskdummy.html',pred="Probability of depression is {}".format(predicted))


if __name__== '__main__' :
    app.run()
```

```
 * Serving Flask app '__main__' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```