Marco Fuentes – 18188
CC3074– Minería de Datos

Laboratorio 3: Classification

Información del dataset

El dataset consiste en un conjunto de datos publicitarios falsos, que indica si un usuario de Internet en particular hizo clic en un anuncio en el sitio web de una empresa. Este conjunto de datos contiene las siguientes características:

- **'Daily Time Spent on Site'**: tiempo del consumidor en el sitio en minutos
- **'Age'**: edad cliente en años
- **'Area Income'**: promedio renta del área geográfica del consumidor
- **'Daily Internet Usage'**: promedio minutos al día que el consumidor está en Internet
- **'Ad Topic Line'**: título del anuncio
- **'City'**: ciudad del consumidor
- **'Male'**: si el consumidor era hombre o no
- **'Country'**: país del consumidor
- **'Timestamp'**: hora a la que el consumidor hizo clic en el anuncio o en la ventana cerrada
- **'Clicked on Ad'**: 0 o 1 indicado si el cliente realiza clic en el anuncio

Hipótesis u objetivo

El objetivo de este laboratorio es construir un modelo que sea capaz de predecir si un usuario hará clic o no en un anuncio dependiendo las características de cada usuario.

Solución y exploración

La solución se divide en dos partes: la exploración de la data, y el modelo para las predicciones.

En la parte de exploración, se corrieron algunos comandos para identificar columnas de la tabla y sus tipos de datos correspondientes:

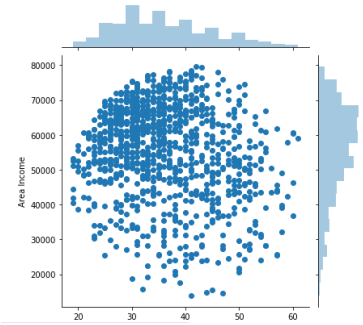
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
Daily Time Spent on Site    1000 non-null float64
Age                         1000 non-null int64
Area Income                 1000 non-null float64
Daily Internet Usage        1000 non-null float64
Ad Topic Line               1000 non-null object
City                        1000 non-null object
Male                        1000 non-null int64
Country                     1000 non-null object
Timestamp                   1000 non-null object
Clicked on Ad               1000 non-null int64
dtypes: float64(3), int64(3), object(4)
memory usage: 78.2+ KB
```

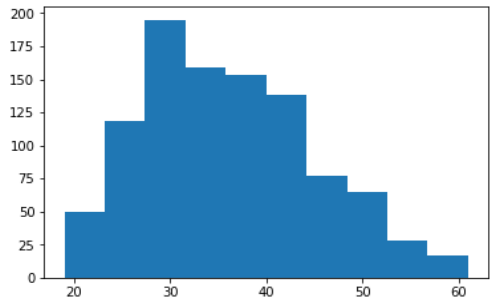
Posteriormente, se corrigieron inconsistencias en los tipos de datos y se continuó la exploración comenzando con la distribución de edades, ingresos mensuales y tiempo promedio en línea :

```
sns.jointplot(data=data, x="Age", y="Area Income")
```

```
<seaborn.axisgrid.JointGrid at 0x7feed4a6da90>
```

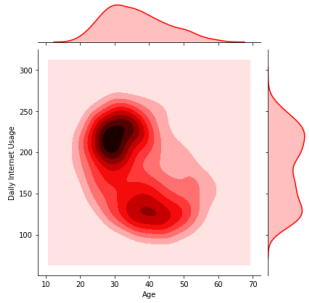



```
plt.hist(data['Age'])
plt.show()
```



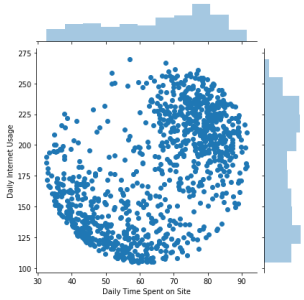
```
sns.jointplot(data=data, x="Age", y="Daily Internet Usage", color="red", kind="kde")
```

```
<seaborn.axisgrid.JointGrid at 0x7feed48d5290>
```

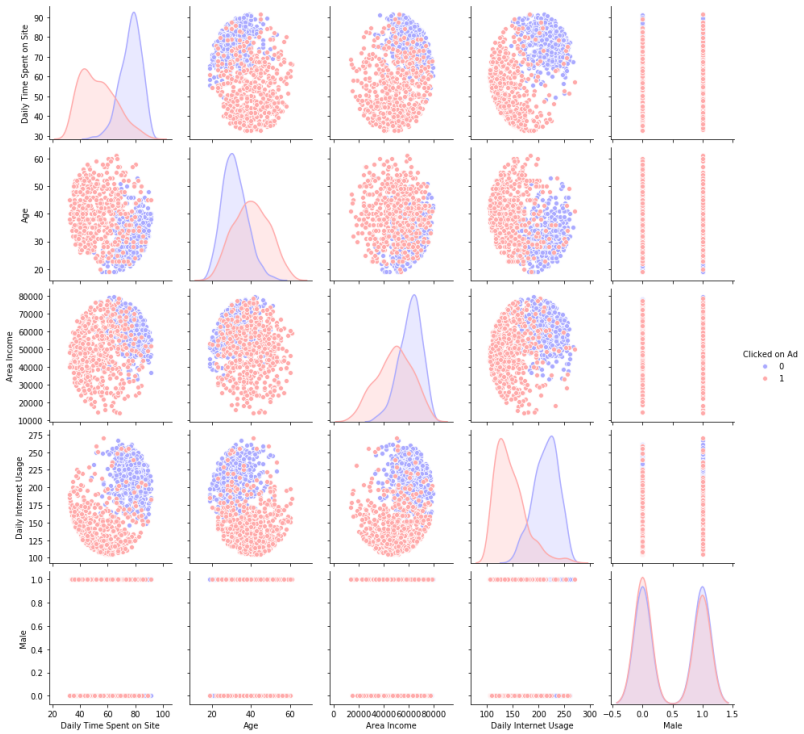



```
sns.jointplot(data=data, x="Daily Time Spent on Site", y="Daily Internet Usage")
```

```
<seaborn.axisgrid.JointGrid at 0x7feed47a0390>
```



Finalmente, se separaron las poblaciones basándose en si el usuario hizo clic en el anuncio o no:



Finalmente, en la parte del modelo, se usó el paquete sklearn de Python para entrenar un modelo de regresión logística basándose en el dataset previamente descrito.

Resultados: puntuales y respaldados por gráficos, tablas y visualizaciones en general

El modelo de clasificación entrenado obtuvo una precisión mayor al 85% en sus predicciones tanto positivas como negativas:

```
print(classification_report(y_test, clf.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.92      0.90      0.91       183
           1       0.88      0.90      0.89       147

   micro avg       0.90      0.90      0.90       330
   macro avg       0.90      0.90      0.90       330
weighted avg       0.90      0.90      0.90       330
```

Con la siguiente confussion matrix:

```
confusion_matrix(y_test, clf.predict(X_test))
```

```
array([[165,  18],
       [ 14, 133]])
```

Sin embargo, si en lugar de un modelo de regresión logística, se usa un Random Forest, los resultados mejoran a 95% de precisión:

```
In [30]: clf = RandomForestClassifier(max_depth=2, random_state=0).fit(X_train, y_train)
```

```
/home/marco/.local/lib/python2.7/site-packages/sklearn/ensemble/forest.py:246: FutureWarning: The default value of n_estimators
will change from 10 in version 0.20 to 100 in 0.22.
  "10 in version 0.20 to 100 in 0.22.", FutureWarning)
```

```
In [31]: print(classification_report(y_test, clf.predict(X_test)))
```

```
              precision    recall  f1-score   support

           0       0.96      0.96      0.96       183
           1       0.95      0.95      0.95       147

   micro avg       0.95      0.95      0.95       330
   macro avg       0.95      0.95      0.95       330
weighted avg       0.95      0.95      0.95       330
```

Con la siguiente confussion matrix:

```
confusion_matrix(y_test, clf.predict(X_test))
```

```
array([[175,   8],
       [  8, 139]])
```