

ELI5 Interpretation model

```
In [1]: pip install eli5
```

Collecting eli5Note: you may need to restart the kernel to use updated packages.

```
  Downloading eli5-0.10.1-py2.py3-none-any.whl (105 kB)
Requirement already satisfied: scipy in d:\python\lib\site-packages (from eli5) (1.4.1)
Requirement already satisfied: six in d:\python\lib\site-packages (from eli5) (1.15.0)
Collecting graphviz
  Downloading graphviz-0.15-py2.py3-none-any.whl (18 kB)
Collecting tabulate>=0.7.7
  Downloading tabulate-0.8.7-py3-none-any.whl (24 kB)
Requirement already satisfied: scikit-learn>=0.18 in d:\python\lib\site-packages (from eli5) (0.23.1)
Requirement already satisfied: jinja2 in d:\python\lib\site-packages (from eli5) (2.11.2)
Requirement already satisfied: numpy>=1.9.0 in d:\python\lib\site-packages (from eli5) (1.18.5)
Requirement already satisfied: attrs>16.0.0 in d:\python\lib\site-packages (from eli5) (19.3.0)
Requirement already satisfied: joblib>=0.11 in d:\python\lib\site-packages (from scikit-learn>=0.18->eli5) (0.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in d:\python\lib\site-packages (from scikit-learn>=0.18->eli5) (2.1.0)
Requirement already satisfied: MarkupSafe>=0.23 in d:\python\lib\site-packages (from jinja2->eli5) (1.1.1)
Installing collected packages: graphviz, tabulate, eli5
Successfully installed eli5-0.10.1 graphviz-0.15 tabulate-0.8.7
```

```
In [3]: import pandas as pd
import numpy as np
import seaborn as sns
import eli5 as eli
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
```

D:\Python\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.metrics.scorer module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.metrics. Anything that cannot be imported from sklearn.metrics is now part of the private API.

```
warnings.warn(message, FutureWarning)
```

D:\Python\lib\site-packages\sklearn\utils\deprecation.py:143: FutureWarning: The sklearn.feature_selection.base module is deprecated in version 0.22 and will be removed in version 0.24. The corresponding classes / functions should instead be imported from sklearn.feature_selection. Anything that cannot be imported from sklearn.feature_selection is now part of the private API.

```
warnings.warn(message, FutureWarning)
```

```
In [28]: data = pd.read_csv(r'C:\Users\Jaswanth Reddy\Downloads\Social_Network_Ads.csv')
data.head(50)
```

Out[28]:

| | User ID | Gender | Age | EstimatedSalary | Purchased |
|----|----------|--------|-----|-----------------|-----------|
| 0 | 15624510 | Male | 19 | 19000 | 0 |
| 1 | 15810944 | Male | 35 | 20000 | 0 |
| 2 | 15668575 | Female | 26 | 43000 | 0 |
| 3 | 15603246 | Female | 27 | 57000 | 0 |
| 4 | 15804002 | Male | 19 | 76000 | 0 |
| 5 | 15728773 | Male | 27 | 58000 | 0 |
| 6 | 15598044 | Female | 27 | 84000 | 0 |
| 7 | 15694829 | Female | 32 | 150000 | 1 |
| 8 | 15600575 | Male | 25 | 33000 | 0 |
| 9 | 15727311 | Female | 35 | 65000 | 0 |
| 10 | 15570769 | Female | 26 | 80000 | 0 |
| 11 | 15606274 | Female | 26 | 52000 | 0 |
| 12 | 15746139 | Male | 20 | 86000 | 0 |
| 13 | 15704987 | Male | 32 | 18000 | 0 |
| 14 | 15628972 | Male | 18 | 82000 | 0 |
| 15 | 15697686 | Male | 29 | 80000 | 0 |
| 16 | 15733883 | Male | 47 | 25000 | 1 |
| 17 | 15617482 | Male | 45 | 26000 | 1 |
| 18 | 15704583 | Male | 46 | 28000 | 1 |
| 19 | 15621083 | Female | 48 | 29000 | 1 |
| 20 | 15649487 | Male | 45 | 22000 | 1 |
| 21 | 15736760 | Female | 47 | 49000 | 1 |
| 22 | 15714658 | Male | 48 | 41000 | 1 |
| 23 | 15599081 | Female | 45 | 22000 | 1 |

| | User ID | Gender | Age | EstimatedSalary | Purchased |
|----|----------|--------|-----|-----------------|-----------|
| 24 | 15705113 | Male | 46 | 23000 | 1 |
| 25 | 15631159 | Male | 47 | 20000 | 1 |
| 26 | 15792818 | Male | 49 | 28000 | 1 |
| 27 | 15633531 | Female | 47 | 30000 | 1 |
| 28 | 15744529 | Male | 29 | 43000 | 0 |
| 29 | 15669656 | Male | 31 | 18000 | 0 |
| 30 | 15581198 | Male | 31 | 74000 | 0 |
| 31 | 15729054 | Female | 27 | 137000 | 1 |
| 32 | 15573452 | Female | 21 | 16000 | 0 |
| 33 | 15776733 | Female | 28 | 44000 | 0 |
| 34 | 15724858 | Male | 27 | 90000 | 0 |
| 35 | 15713144 | Male | 35 | 27000 | 0 |
| 36 | 15690188 | Female | 33 | 28000 | 0 |
| 37 | 15689425 | Male | 30 | 49000 | 0 |
| 38 | 15671766 | Female | 26 | 72000 | 0 |
| 39 | 15782806 | Female | 27 | 31000 | 0 |
| 40 | 15764419 | Female | 27 | 17000 | 0 |
| 41 | 15591915 | Female | 33 | 51000 | 0 |
| 42 | 15772798 | Male | 35 | 108000 | 0 |
| 43 | 15792008 | Male | 30 | 15000 | 0 |
| 44 | 15715541 | Female | 28 | 84000 | 0 |
| 45 | 15639277 | Male | 23 | 20000 | 0 |
| 46 | 15798850 | Male | 25 | 79000 | 0 |
| 47 | 15776348 | Female | 27 | 54000 | 0 |
| 48 | 15727696 | Male | 30 | 135000 | 1 |
| 49 | 15793813 | Female | 31 | 89000 | 0 |

```
In [18]: data.isna().sum()
```

```
Out[18]: User ID      0
Gender      0
Age         0
EstimatedSalary  0
Purchased   0
dtype: int64
```

```
In [19]: #split the dataset into independent and dependent variables
```

```
X = data.iloc[:, [1, 2, 3]]
```

```
y = data.iloc[:, [4]]
```

```
print("Independent Variables(Predictors) :n", X.head(), "nDependent Variable(Target):n", y.head())
```

```
Independent Variables(Predictors) :n  Gender  Age  EstimatedSalary
0   Male   19           19000
1   Male   35           20000
2  Female  26           43000
3  Female  27           57000
4   Male   19           76000 nDependent Variable(Target):n  Purchased
0           0
1           0
2           0
3           0
4           0
```

```
In [20]: encoder = LabelEncoder()
```

```
X.Gender = encoder.fit_transform(X.Gender)
```

```
# Splitting the dataset into the Training and Testing set
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25)
```

D:\Python\lib\site-packages\pandas\core\generic.py:5303: SettingWithCopyWarning:

A value is trying to be set on a copy of a slice from a DataFrame.

Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
self[name] = value
```

```
In [21]: # Decision Tree Classifier(A decision tree is basically a time series mixed with probability of up and down move  
classifier_dtc = DecisionTreeClassifier()  
classifier_dtc.fit(X_train, y_train)
```

```
Out[21]: DecisionTreeClassifier()
```

```
In [30]: # Predicting the Test set results  
y_pred = classifier_dtc.predict(X_test)
```

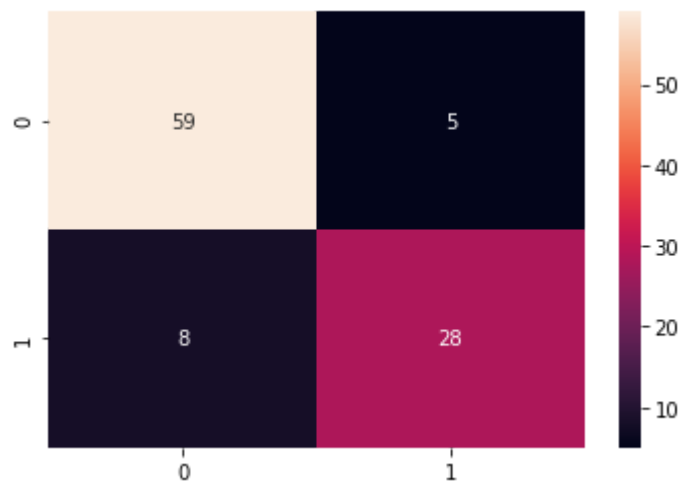
```
In [23]: # Making the Confusion Matrix
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
clf_rpt = classification_report(y_test,y_pred)
print("classification report :", clf_rpt)
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm,annot=True)
```

```
classification report :                precision    recall  f1-score   support

      0      0.88      0.92      0.90         64
      1      0.85      0.78      0.81         36

 accuracy          0.87         100
 macro avg      0.86      0.85      0.86         100
weighted avg      0.87      0.87      0.87         100
```

```
Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x226b299b4c0>
```



Model Interpretation

```
In [24]: # To know how the model is giving high accuracy and to find each weight that contributes  
eli.explain_weights(classifier_dtc)
```

```
Out[24]:
```

| Weight | Feature |
|--------|---------|
| 0.5398 | x1 |
| 0.4251 | x2 |
| 0.0351 | x0 |

```
In [25]: # To debug and check how the outcome for a particular row  
eli.explain_prediction(classifier_dtc , np.array(X_test)[1])
```

```
Out[25]: y=0 (probability 1.000) top features
```

| Contribution? | Feature |
|---------------|---------|
| +0.643 | <BIAS> |
| +0.428 | x1 |
| -0.071 | x2 |


```
In [26]: # Comparing what combination of features and values lead to a particular prediction, we can use show_prediction(  
eli.show_prediction(classifier_dtc, X_test.iloc[1],  
                    feature_names=list(X.columns),  
                    show_feature_values=True)
```

Out[26]: **y=0** (probability **1.000**) top features

| Contribution? | Feature | Value |
|---------------|-----------------|-----------|
| +0.643 | <BIAS> | 1.000 |
| +0.428 | Age | 39.000 |
| -0.071 | EstimatedSalary | 71000.000 |

```
In [27]: #A prediction is basically the sum of positive features inclusive of bias.  
eli.show_prediction(classifier_dtc, X_test.iloc[20],  
                    feature_names=list(X.columns),  
                    show_feature_values=True)
```

Out[27]: **y=0** (probability **1.000**) top features

| Contribution? | Feature | Value |
|---------------|-----------------|-----------|
| +0.643 | <BIAS> | 1.000 |
| +0.239 | Age | 21.000 |
| +0.117 | EstimatedSalary | 88000.000 |

In []:

In []: