

30. Logistic Regression (Practical) (Binary Classification) (Polynomial Input)

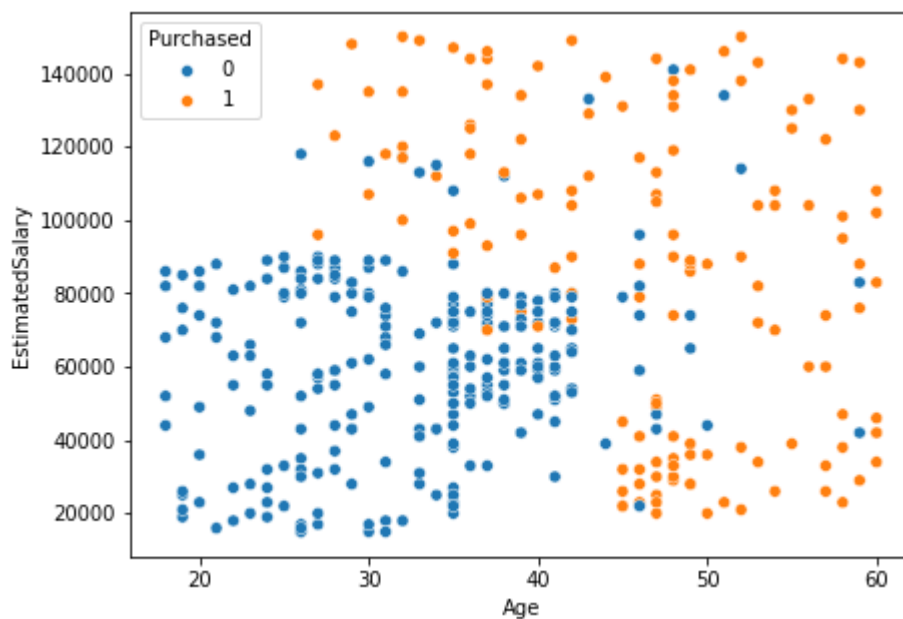
```
In [1]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [5]: dataset = pd.read_csv(r'Data/Social_Network_Ads_2.csv')
dataset.head(3)
```

```
Out[5]:
```

	Age	EstimatedSalary	Purchased
0	19	19000	0
1	35	20000	0
2	26	43000	0

```
In [7]: plt.figure(figsize=(7,5))
sns.scatterplot(x="Age", y="EstimatedSalary", data=dataset, hue="Purchased")
plt.show()
```



```
In [9]: x = dataset.iloc[:, :-1]
x
```

Out[9]:

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000
...
395	46	41000
396	51	23000
397	50	20000
398	36	33000
399	49	36000

400 rows × 2 columns

```
In [10]: y=dataset["Purchased"]  
y
```

```
Out[10]: 0      0  
1      0  
2      0  
3      0  
4      0  
      ..  
395    1  
396    1  
397    1  
398    0  
399    1  
Name: Purchased, Length: 400, dtype: int64
```

```
In [11]: from sklearn.model_selection import train_test_split
```

```
In [12]: x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.20, random_stat
```

```
In [13]: from sklearn.linear_model import LogisticRegression
```

```
In [14]: lr = LogisticRegression()  
lr.fit(x_train, y_train)
```

```
Out[14]: ▾ LogisticRegression  
LogisticRegression()
```

```
In [16]: lr.score(x_test, y_test)*100
```

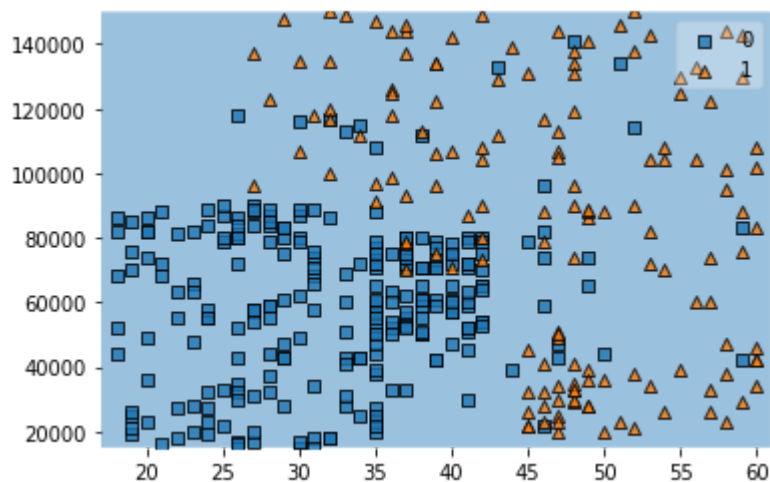
```
Out[16]: 65.0
```

```
In [ ]:
```

```
In [19]: from mlxtend.plotting import plot_decision_regions
```

```
In [20]: plot_decision_regions(x.to_numpy(), y.to_numpy(), clf=lr)
plt.show()
```

C:\Users\rashi\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
warnings.warn(



You can see the data is not linearly separable. so we will classify this data through **polynomial Feature**

```
In [21]: from sklearn.preprocessing import PolynomialFeatures
```

```
In [42]: pf = PolynomialFeatures(degree=3)
pf.fit(x)
pf.transform(x)
```

```

-----
ValueError                                Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_12444\2232260164.py in <module>
      1 pf = PolynomialFeatures(degree=3)
      2 pf.fit(x)
----> 3 pf.transform(x)

~\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\preprocessing\po
lynomial.py in transform(self, X)
      430         # Do as if _min_degree = 0 and cut down array after the
      431         # computation, i.e. use _n_out_full instead of n_output_features
      432         XP = np.empty(
      433             shape=(n_samples, self._n_out_full), dtype=X.dtype, order=se
      434             lf.order
      435         )

ValueError: array is too big; `arr.size * arr.dtype.itemsize` is larger than the max
imum possible size.

```

The output data is in array form so we will convert it into dataset format

```
In [ ]: x = pd.DataFrame(pf.transform(x))
        x.head(5)
```

```
In [ ]: from sklearn.model_selection import train_test_split
```

```
In [35]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.20, random_st
```

```
In [36]: from sklearn.linear_model import LogisticRegression
```

```
In [37]: lrg = LogisticRegression()
        lrg.fit(x_train, y_train)
```

```

C:\Users\rashi\AppData\Local\Programs\Python\Python39\lib\site-packages\sklearn\line
ar_model\_logistic.py:444: ConvergenceWarning: lbfgs failed to converge (status=2):
ABNORMAL_TERMINATION_IN_LNSRCH.

```

Increase the number of iterations (max_iter) or scale the data as shown in:

<https://scikit-learn.org/stable/modules/preprocessing.html>

Please also refer to the documentation for alternative solver options:

https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression

```
n_iter_i = _check_optimize_result(
```

```
Out[37]: ▼ LogisticRegression
        LogisticRegression()
```

```
In [38]: lrg.score(x_test, y_test)*100
```

```
Out[38]: 65.0
```

```
In [ ]:
```

