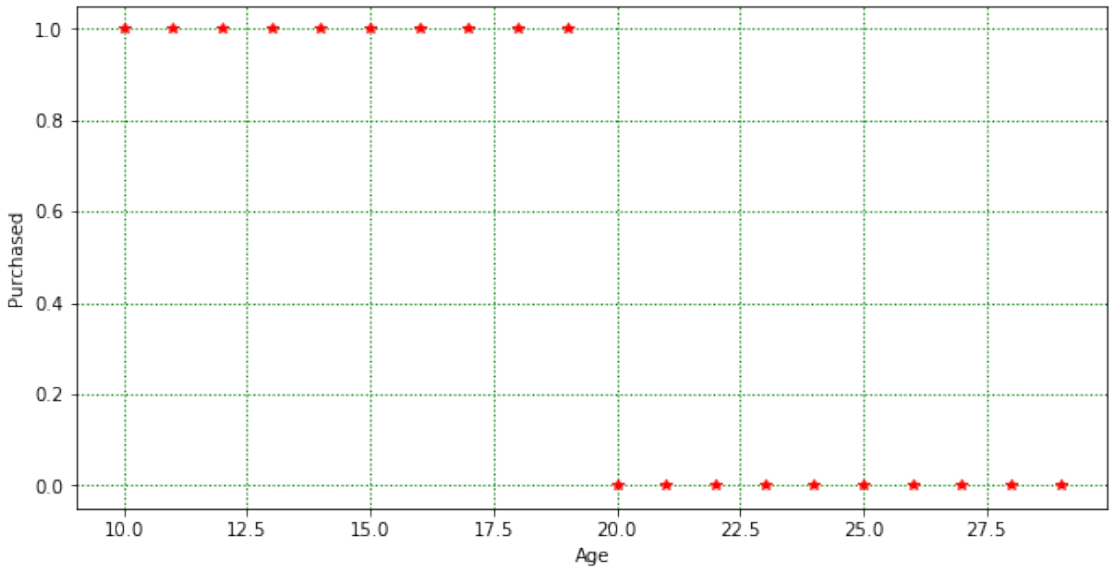
linear and logistic

March 3, 2020

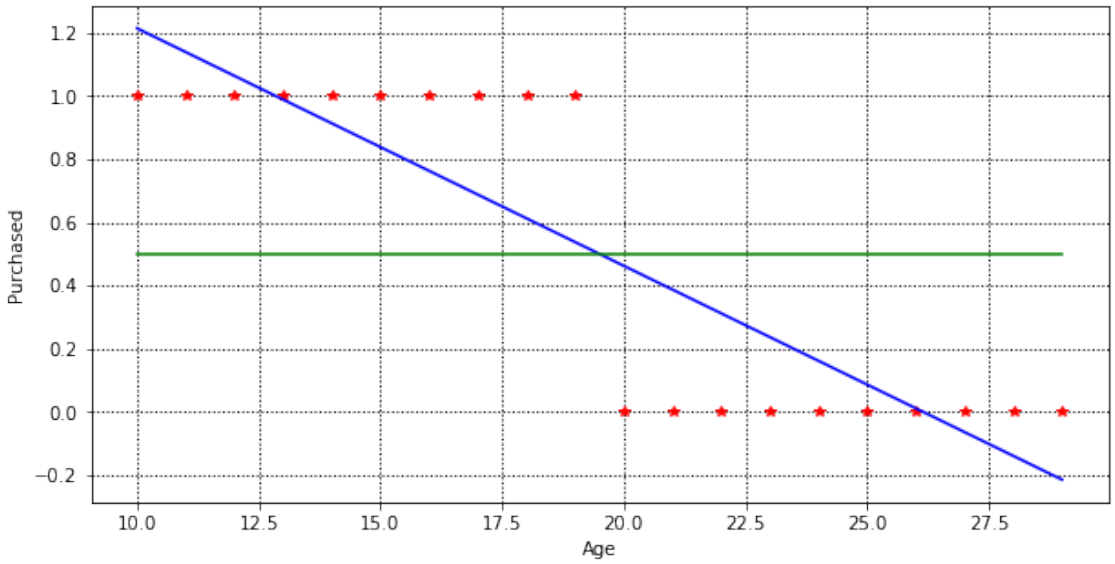
[16]:
```
#linear and logistic regression
# Suja Basnet

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

from sklearn.linear_model import LinearRegression
lin_regression = LinearRegression()

from sklearn.linear_model import LogisticRegression
log_regression = LogisticRegression(solver='lbfgs')

from sklearn.metrics import mean_squared_error, r2_score
```

[17]:
```
x = np.array([i for i in range(10,30)])
y = np.concatenate([np.ones(10), np.zeros(10)])
```

[18]:
```
fig = plt.figure(figsize=(10,5))
xlabel = 'Age'
ylabel = 'Purchased'
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.grid(color='green', linestyle=':', linewidth=1)
_ = plt.plot(x, y, '*r')
```



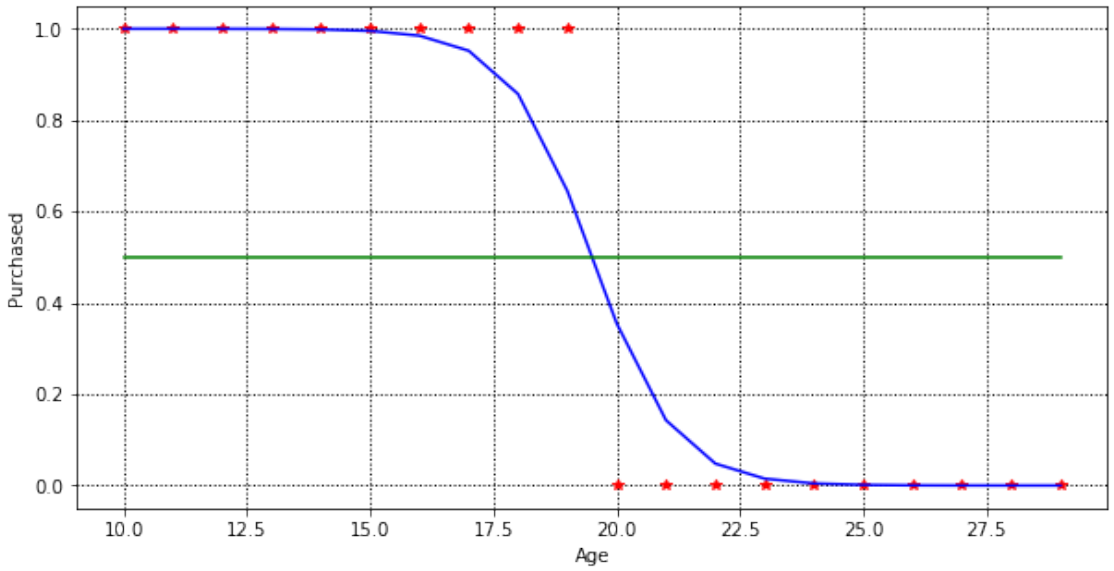
[19]:
```
#linear regression

_ = lin_regression.fit(pd.DataFrame(x), y)

lin_y_pred_1 = lin_regression.predict(pd.DataFrame(x))

line_point_5 = x * 0 + .5

fig = plt.figure(figsize=(10,5))
xlabel = 'Age'
ylabel = 'Purchased'
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.grid(color='k', linestyle=':', linewidth=1)
plt.plot(x, y, '*r')
plt.plot(x, lin_y_pred_1, '-b')
_ = plt.plot(x, line_point_5,'-g')
```



[20]:
```
  #Logistic Regression
test_x = np.array([i for i in range(10,35,5)])
test_y_pred = lin_regression.predict(pd.DataFrame(test_x))
test_y_pred
```

[20]:
```
array([ 1.21428571,  0.83834586,  0.46240602,  0.08646617, -0.28947368])
```

[21]:
```
_ = log_regression.fit(pd.DataFrame(x), y)

y_pred = log_regression.predict_proba(pd.DataFrame(x))
log_y_pred_1 = [item[1] for item in y_pred]

fig = plt.figure(figsize=(10,5))
xlabel = 'Age'
ylabel = 'Purchased'
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.grid(color='k', linestyle=':', linewidth=1)
plt.plot(x, y, '*r')
plt.plot(x, log_y_pred_1, '-b')
_ = plt.plot(x, line_point_5,'-g')
```

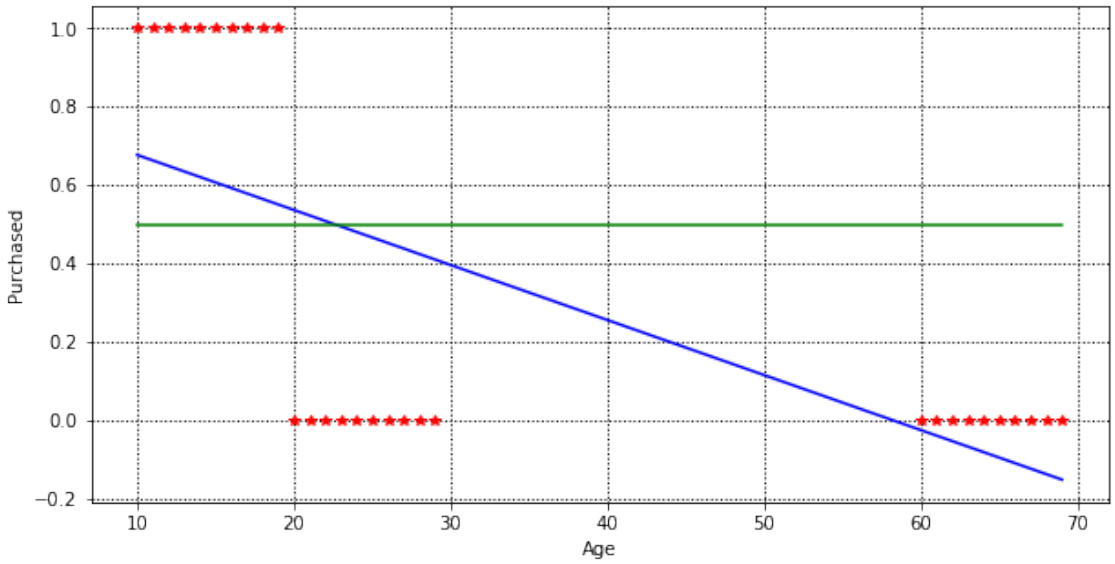


[22]:
```
print("Linear regression R2: ", r2_score(y, lin_y_pred_1))
print("Logistic regression R2: ", r2_score(y, log_y_pred_1))
print("Linear regression RMSE: ", mean_squared_error(y, lin_y_pred_1))
print("Logistic regression RMSE: ", mean_squared_error(y, log_y_pred_1))
```

```
Linear regression R2:  0.7518796992481203
Logistic regression R2:  0.9404089597242656
Linear regression RMSE:  0.062030075187969935
Logistic regression RMSE:  0.014897760068933596
```

[23]:
```
x = np.append(x, np.array([i for i in range(60,70)]))
y = np.append(y, np.zeros(10))
```

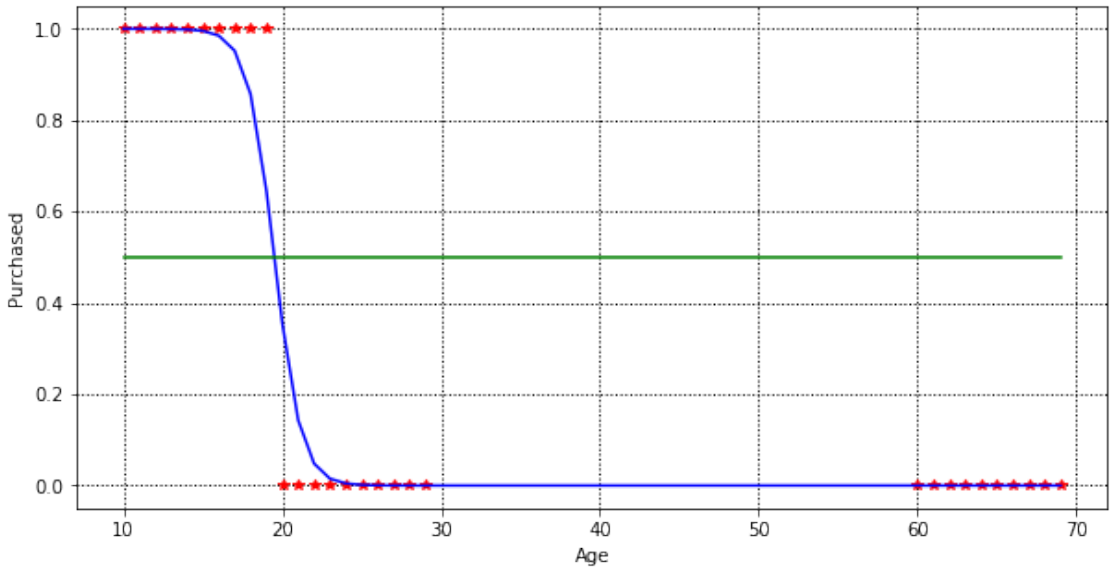
[24]:
```
_ = lin_regression.fit(pd.DataFrame(x), y)

lin_y_pred_2 = lin_regression.predict(pd.DataFrame(x))

line_point_5 = x * 0 + .5

fig = plt.figure(figsize=(10,5))
xlabel = 'Age'
ylabel = 'Purchased'
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.grid(color='k', linestyle=':', linewidth=1)
plt.plot(x, y, '*r')
plt.plot(x, lin_y_pred_2, '-b')
_ = plt.plot(x, line_point_5,'-g')
```



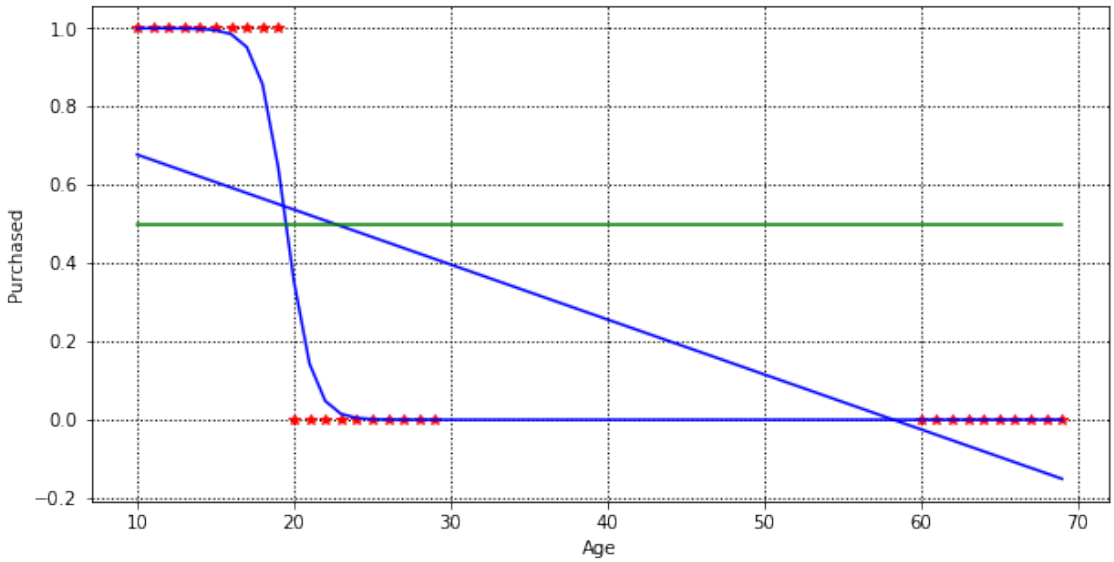
[26]:
```
test_x = np.array([i for i in range(18,26)])
test_y_pred = lin_regression.predict(pd.DataFrame(test_x))
test_y_pred
```

[26]:
```
array([0.56495292, 0.55091537, 0.53687781, 0.52284026, 0.50880271,
       0.49476516, 0.48072761, 0.46669006])
```

[28]:
```
_ = log_regression.fit(pd.DataFrame(x), y)

y_pred = log_regression.predict_proba(pd.DataFrame(x))
log_y_pred_2 = [item[1] for item in y_pred]

fig = plt.figure(figsize=(10,5))
xlabel = 'Age'
ylabel = 'Purchased'
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.grid(color='k', linestyle=':', linewidth=1)
plt.plot(x, y, '*r')
plt.plot(x, log_y_pred_2, '-b')
_ = plt.plot(x, line_point_5,'-g')
```



[32]:
```
fig = plt.figure(figsize=(10,5))
xlabel = 'Age'
ylabel = 'Purchased'
plt.xlabel(xlabel)
plt.ylabel(ylabel)
plt.grid(color='k', linestyle=':', linewidth=1)
plt.plot(x, y, '*r')
plt.plot(x, lin_y_pred_2, '-b')
plt.plot(x, log_y_pred_2, '-b')
_ = plt.plot(x, line_point_5,'-g')
```



[33]:
```
test_y_pred = log_regression.predict_proba(pd.DataFrame(test_x))
test_y_pred
```

[33]:
```
array([[0.14286332, 0.85713668],
       [0.35497559, 0.64502441],
       [0.64502249, 0.35497751],
       [0.85713565, 0.14286435],
       [0.95194543, 0.04805457],
       [0.98494151, 0.01505849],
       [0.99539093, 0.00460907],
       [0.99859958, 0.00140042]])
```

[34]:
```
print("Linear regression R2: ", r2_score(y, lin_y_pred_2))
print("Logistic regression R2: ", r2_score(y, log_y_pred_2))
print("Linear regression RMSE: ", mean_squared_error(y, lin_y_pred_2))
print("Logistic regression RMSE: ", mean_squared_error(y, log_y_pred_2))
```

```
Linear regression R2:  0.42112651342340734
Logistic regression R2:  0.9553066567250715
Linear regression RMSE:  0.12863855257257611
Logistic regression RMSE:  0.009931854061095221
```