Couger550TermProject_Final

September 20, 2023

1 Who's Clicking on Your Ads?

Corbin Couger

4/20/2023

1.1 Milestone 1

1.1.1 The Idea

In this project I want to aid in solving a relavent, expensive, and prominent problem that many different business's face, ads! The main way companies get their name out there and acquire new customers is through ads, which tend to be expensive as well as random. By random, I mean without historic data, the company will do there best to place the add near their expected demographic which can be a rough 'guess' until they get the data to help them get the ad in front of the right customers. This is where I plan to come in... I want to explore a Ad Click dataset I've discovered that will allow me to build a prediction model, most likely logistic regression, to help determine whether a customer made a purchase or not because of the ad they clicked on. My goal is to help those business's get a structure and framework of a model for when they first get that initial dataset built. i.e. once they have their customer ad click data, how do they better direct that ad based on who is buying and drive that customer acquisition cost down. This will be logistic regression model to determine if a user purchased from the company or not after they clicked on the add. This will help determine many questions, the demograpic that the ad is reaching or if the ad might need to go back to the drawing board.

The reason I want to do this is because in the beginning of a companies journey, money can be tight, and now adays with ecommerce and ads driving a lot of a companies expantion and sales, the costs get expensive when running a lot of ads in such a competitive market. There are also so many places to run ads and that will be a problem for a different project, but the main reason is to help make things cheaper for businesses that are just starting out with ads and getting their initial customers. Once that happens, there is often a much quicker growth with now there being ads and word of mouth.

[46]:
```
import pandas as pd
import matplotlib.pyplot as plt

#milestone 3
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score
from sklearn.metrics import roc_curve
```

[2]:
```
ad_df = pd.read_csv('Social_Network_Ads.csv')
ad_df.head(6)
```

[2]:

	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

[3]:
```
ad_df.shape
```

[3]:
```
(400, 5)
```

[4]:
```
ad_df.dtypes
```

[4]:
```
User ID             int64
Gender             object
Age                 int64
EstimatedSalary     int64
Purchased           int64
dtype: object
```

1.1.2 The Data

As shown above, the data I've attained is a compilation of multiple different instances of ad click-thru. The data consists of 5 columns and 400 rows. The columns are described as so:

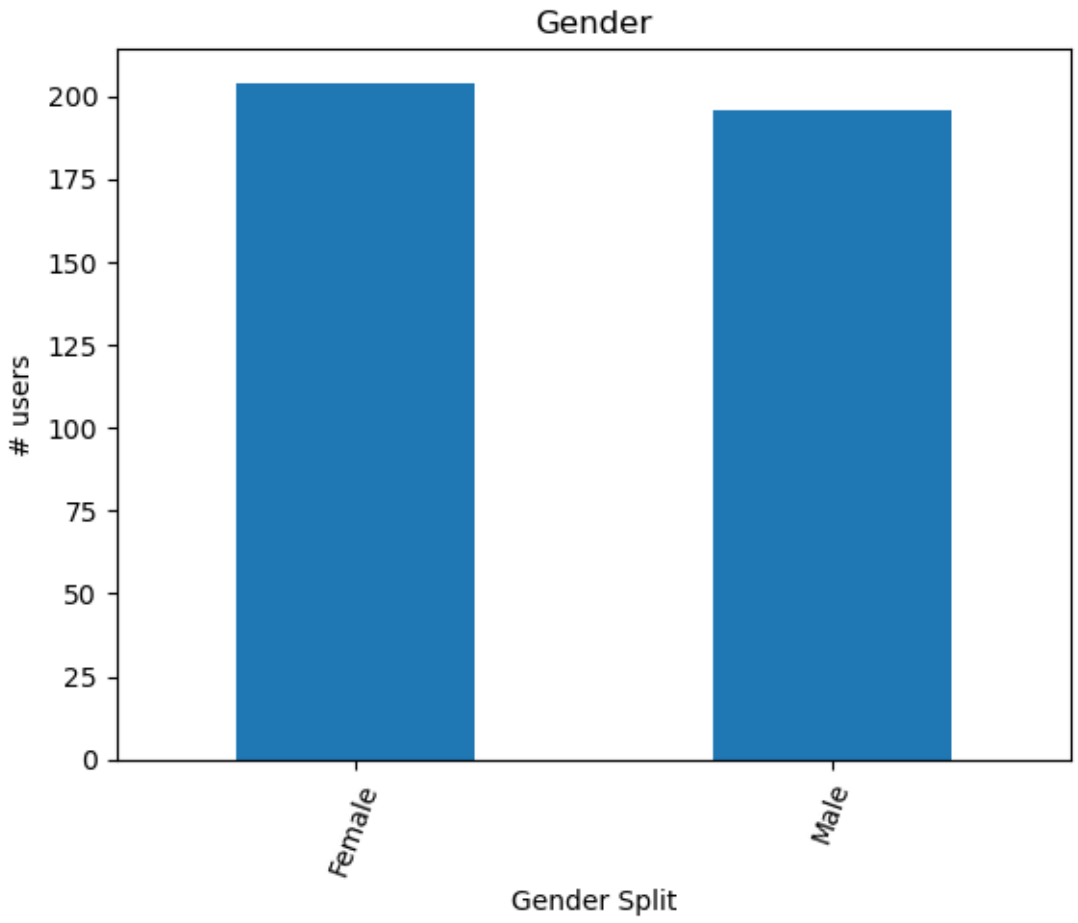
- User ID: unique identifer for each user who saw and clicked on the ad
- Gender: the gender of the user who clicked on the ad
- Age: the age of the user who clicked on the ad
- EstimatedSalary: the estimated salary for the user that clicked on the ad
- Purchased: this will be my dependent variable, whether the user purchased something (1) or not (0) after clicking on the ad.

1.1.3 Graphical Analysis

Now that I've explained my idea(s), I will create and describe a few visualizations with my data to help get a better understanding of what the data looks like.

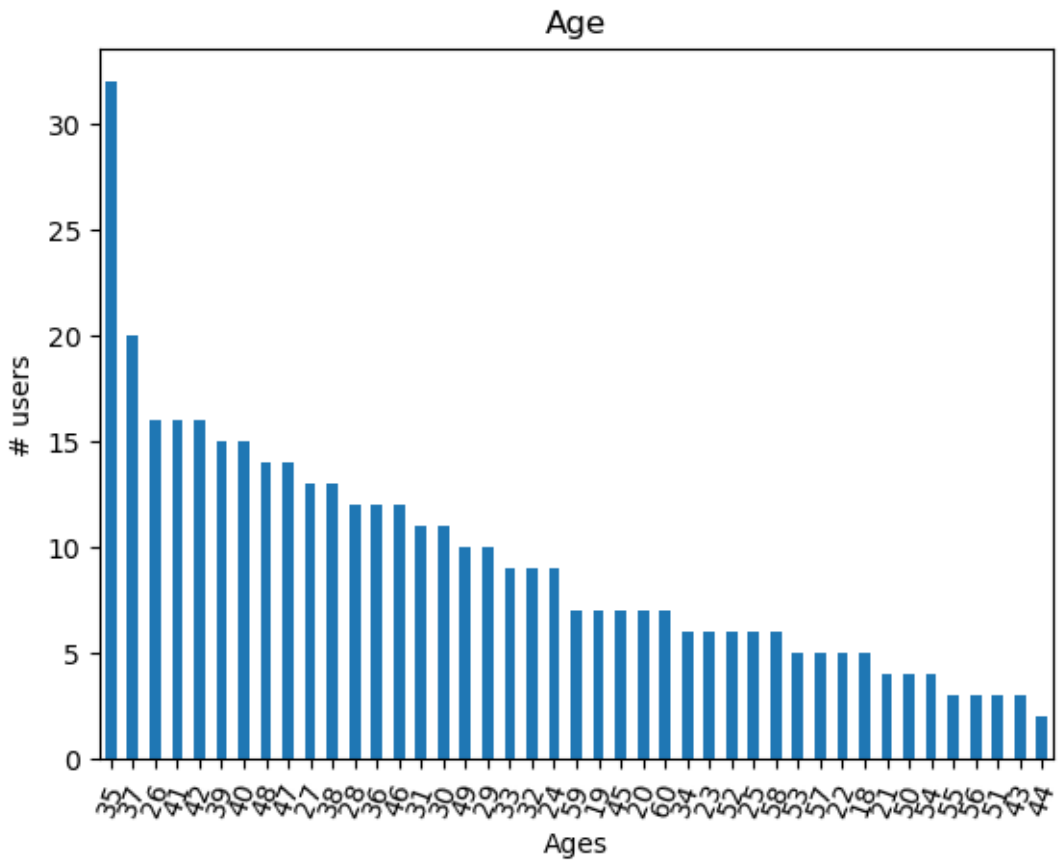
[5]:
```
ad_df['Gender'].value_counts().plot(kind='bar')

plt.xlabel("Gender Split")
plt.xticks(rotation=70)
```

```
plt.ylabel("# users")
plt.title("Gender")
plt.show()
```



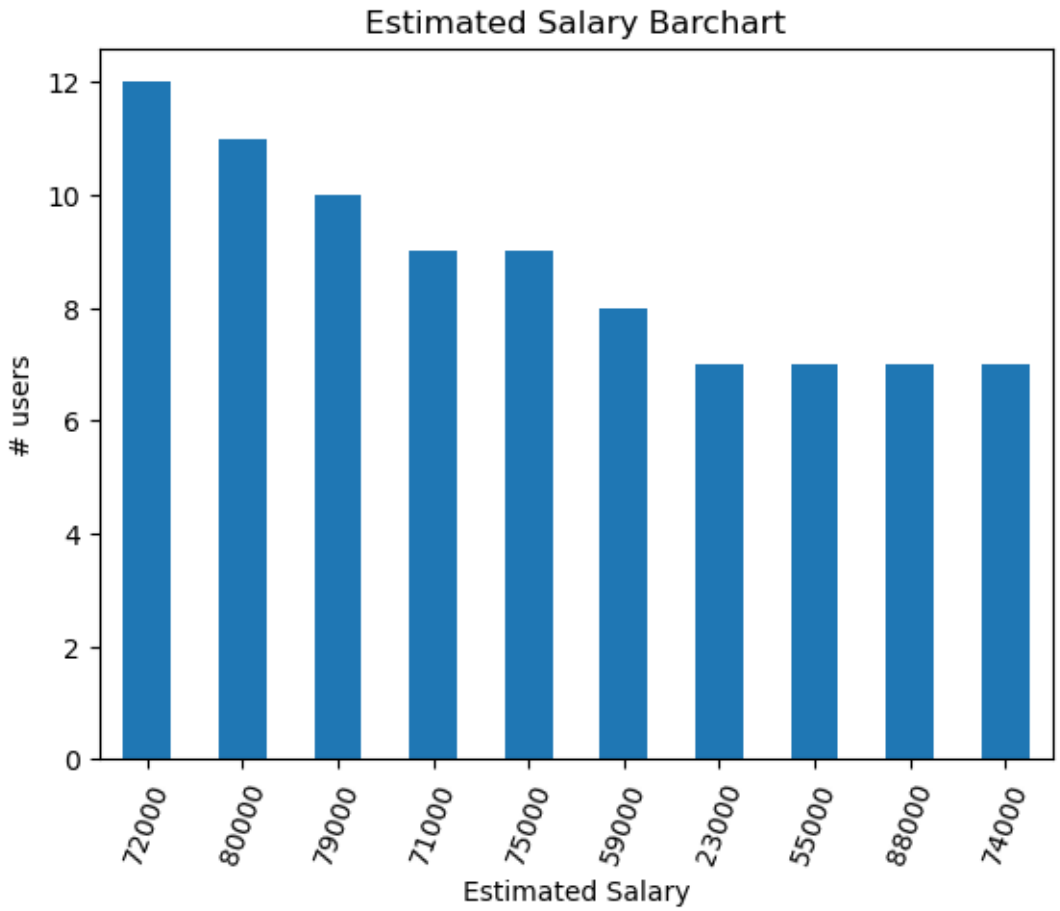
Looking at this plot, I can see that there are more Females clicking this ad than males but not by much. This is interesting and am looking forward to how this will play out in the regression model to see whether or not this is a significant predictor variable.

[6]:
```
ad_df['Age'].value_counts().plot(kind='bar')

plt.xlabel("Ages")
plt.xticks(rotation=70)
plt.ylabel("# users")
plt.title("Age")
plt.show()
```



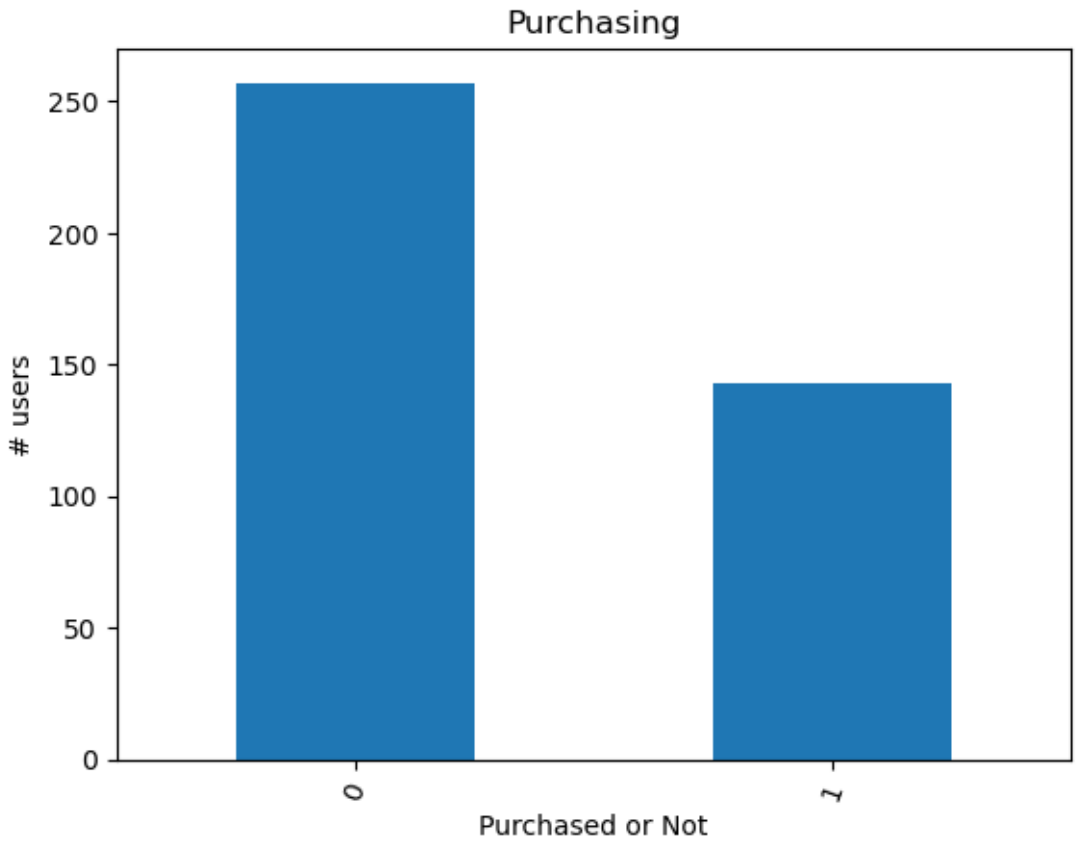
Though this is slightly cluttered, I can tell that the majority of the users that clicked on this add were 35 by almost double of any other age. Now, a company isn't going to solely target one age, but will more-so target an age range. Just based off of this plot, I see that most the users who clicked on the ad were between 35-48. There are only 2 of the top 10 ages that were outside that range.

[7]:
```
ad_df['EstimatedSalary'].value_counts().sort_values(ascending = False).head(10).
  ↪plot(kind='bar')

plt.xlabel("Estimated Salary")
plt.xticks(rotation=70)
plt.ylabel("# users")
plt.title("Estimated Salary Barchart")
plt.show()
```



I'm now looking at the top 10 estimated salarys that clicked through this ad. It seems thos who make 71k-80k click on this ad more than others in the top 10. This is interesting as I am eager to see how this links up with the other predictor variables and how it splits based on them as well.

[8]:
```
ad_df['Purchased'].value_counts().plot(kind='bar')

plt.xlabel("Purchased or Not")
plt.xticks(rotation=70)
plt.ylabel("# users")
plt.title("Purchasing")
plt.show()
```



Looking at this, I can see the users that clicked on this are 40% more likely to not purchase anything, but there is still a good chunk of users that did. This tells me that the ad is working, but not to a desired extent and hopefully the data will be able to deterine a way to get the purchase rate up.

Overall, this graphical analysis was insightful and imformative. I now know my data better and can start to prepare my data better for modeling. Gaining an understanding of this data in this part of my analysis is important as it will be one of the only times I get to visualize my data. The first 3 visauls that show my independent variables help me understand how the data is broken up amongst those fields. Then in my last visual, looking at the dependent variable gives me an understanding that this analysis is needed, first to determine who is clicking through and purchasing, and secondly to get that click-thru purchase rate up.

1.2 Milestone 2

Within this Milestone I will be doing some data preparation. I want to make sure that my data is going to be set for building models and analysis in the next milestones.

1.2.1 Cleaning Up Variables

I noticed in the 'Gender' column there is trailing white space and that is not going to be good for what I plan on doing with that variable in the following steps. So, the first transformation/cleaning I will make is to strip this column of its whitespace.

[9]:
```
ad_df.Gender = [Gender.strip() for Gender in ad_df.Gender]
```

[10]:
```
ad_df.head(3)
```

[10]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0

1.2.2 Creating Dummy Variables

For my analysis I am going to want some numeric variables but instead of creating dummy variables, I'm going to keep it in the same 'Gender' variable.

[11]:
```
ad_df['Gender'] = ad_df['Gender'].replace(['Male', 'Female'], [0, 1])
```

[12]:
```
ad_df.head()
```

[12]:

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	0	19	19000	0
1	15810944	0	35	20000	0
2	15668575	1	26	43000	0
3	15603246	1	27	57000	0
4	15804002	0	19	76000	0

Now I will check to see if this got it to an integer

[13]:
```
ad_df.dtypes
```

[13]:
```
User ID            int64
Gender             int64
Age                int64
EstimatedSalary    int64
Purchased          int64
dtype: object
```

Perfect! Now this dataset is thankfully really clean already but I will now check for some other possible issues.

1.2.3 Checking for Missing Values (NaN's)

I want to make sure this data is very clean or at least see if there are missing values to see what could be the issue for that. If there are missing values in any of the 'Gender', 'Age', 'EstimatedSalary' or 'Purchased' columns I will remove those rows as I have few independent variables as it is and a missing value in one of those 3 could impact the model.

[14]:
```
ad_df.isnull().sum()
```

[14]:
```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

Thankfully, I found a nice fairly clean dataset and there are no missing values.

1.2.4 Checking Correlation and Final Preparation

Since my dataset is so small and there are only 3 total independent variables that will go into predicting my dependent variable. I do not want to remove features or much data at all but want to make sure that the data is suitable and ready for my model in the upcoming steps. In this step, I want to just check how the correlation varies amongst all the features related to the target.

[15]:
```
ad_df.corr()['Purchased']
```

[15]:
```
User ID            0.007120
Gender             0.042469
Age                0.622454
EstimatedSalary    0.362083
Purchased          1.000000
Name: Purchased, dtype: float64
```

Looking at output, I can see that 'EstimatedSalary' and 'Age' have the highest correlation with whether or not a user purchased an item because of the ad or not. This is useful to me and will be useful to keep in mind when interpreting the results of my model.

All in all, these short, but important steps in preparing my data for analysis reigned useful and impactufl. I needed to make sure that the data is suitable for Logistic Regression and I made sure to do that by making the dataset numeric, checking for missing values, and by ensuring the features I have are correlated to my target variable. I am excited for the next steps in this project and hope this dataset can provide the answers I want to be able to help out the business case.

1.3 Milestone 3.

For this milestone I am going to start getting into the creation of the model, but beforehand I want to explain what model I plan on training this data to.

When looking at my dependent variable, I know that there are only two posibilities, whether someone purchased based on the ad (1) or didn't (0). Along with this, I'm trying to create a prediction of when someone will purchase something based on the ad so I know I need some sort of prediction model. With these two things in mind and since I've already mentioned it earlier in this analysis, I will be building a Logistic Regression Model.

The Logistic Regression Model isn't really regression in a sense. It is more so a classification algorythm that predicts the probability of a class, in our case purchase or not purchase. This model will be great as it works really well with binary classification.

1.3.1 Step 1.

So, without further or do, I'm going to get into building this model! I'll explain my steps along the way! I first want to get rid of the 'User ID' column, as it wont have any good use in my model. Then I'm going to split my data into train and test sets as well as identifying the features (independent variables) and the target (dependent variable).

[22]:
```
del ad_df['User ID']
train, test = train_test_split(ad_df, test_size=0.2)
features = train.loc[:, train.columns != 'Purchased']
target = train.Purchased
```

[23]:
```
train.shape
```

[23]:
```
(320, 4)
```

[24]:
```
test.shape
```

[24]:
```
(80, 4)
```

[25]:
```
test_features = test.loc[:, test.columns != 'Purchased']
test_target = test.Purchased
```

1.3.2 Step 2.

Next I'm going to standardize my features so the analysis of the results is easier and I dont have to worry about units.

[29]:
```
scaler = StandardScaler()
features_standardized = scaler.fit_transform(features)
test_features_standardized = scaler.fit_transform(test_features)
```

1.3.3 Step 3.

Now it is time to create a logistic regressor and train the data to it to create a model.

[27]:
```
logistic_regression = LogisticRegression(random_state=0)
model = logistic_regression.fit(features_standardized, target)
```

1.3.4 Step 4.

Now I want to create my predictions variable so I can get to calculating some statistics on my model's results.

[30]:
```
predictions = model.predict(test_features_standardized)
```

1.3.5 Step 5.

Wooo, I have my model and am ready to analyze some results. These will be just some simple checks to see how well my model was created. I first want to take a look at the coefficients for the

model.

[41]:
```
pd.DataFrame({'coeff': logistic_regression.coef_[0]},
             index=features.columns)
```

[41]:

	coeff
Gender	-0.105610
Age	2.180772
EstimatedSalary	0.981216

Just looking at the coefficient, it seems that 'Age' has the biggest weight on whether or not a purchase was made based on the ad. 'EstimatedSalary' also has a slight impact on the outcome. This is cool to see and good information, but this will only mean something if the model is a good fit for the data. I'll calculate some stats that will check this. First, the accuracy:

[42]:
```
print('Accuracy of logistic regression classifier on test set: {:.2f}'.
  ↪format(logistic_regression.score(test_features_standardized, test_target)))
```

```
Accuracy of logistic regression classifier on test set: 0.88
```

This is not bad at all, an 88% accurate model is something I was hoping to see, I was wanting to get a model that could predict purchases at least 80% of the time.

[45]:
```
print(classification_report(test_target, predictions))
```

```
              precision    recall  f1-score   support

           0       0.91      0.91      0.91        53
           1       0.81      0.81      0.81        27

    accuracy                           0.88        80
   macro avg       0.86      0.86      0.86        80
weighted avg       0.88      0.88      0.88        80
```

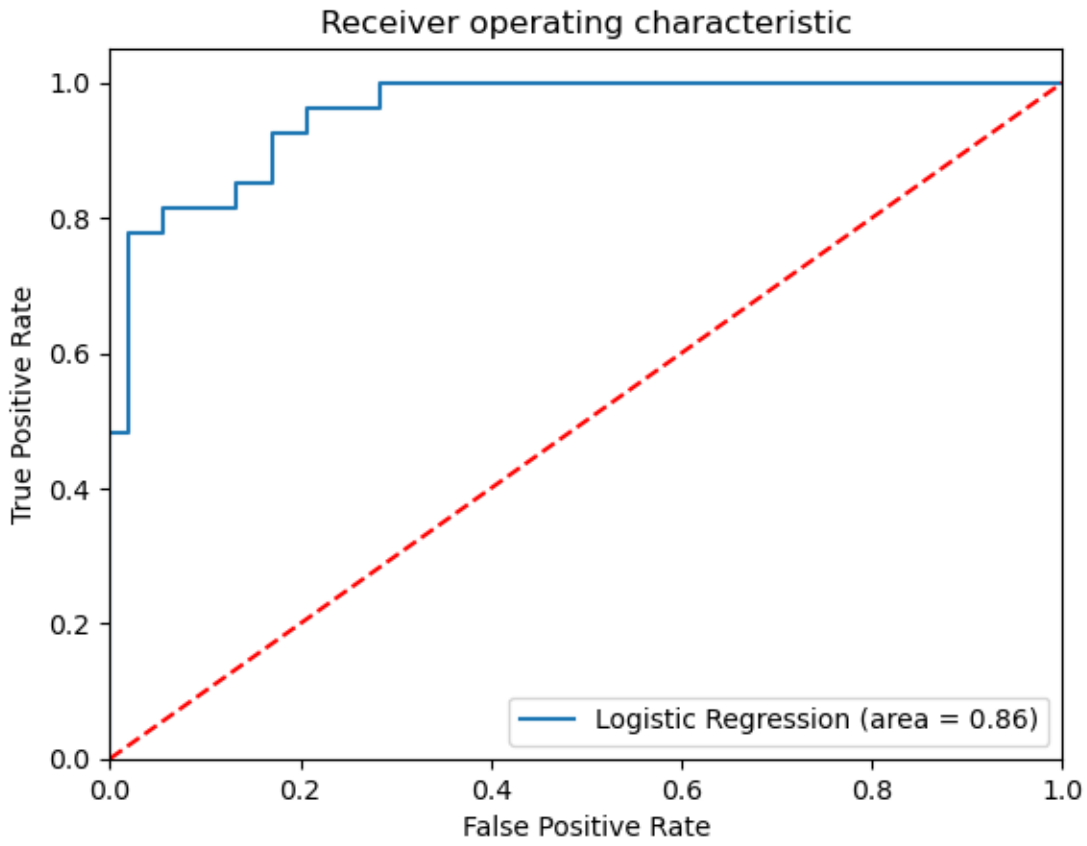
Looking at my recall, I can see that it is also .91 for classifying non-purchases and .81 for purchases. The recall tells me what proportion of my true positives (the correct predictions) were identified correctly. This is a good to me as the average recall is .88 and that means 88% of the data was truly identified in the model.

Precision is somewhat similar, it is telling me what parts of my positively identified data is actually correct. This is a high number as well (average of .88) and that leaves me satisfied to say that this model is good for predicting whether or not the ad will produce a purchase or not based on the independent variables.

Finally for this Milestone, I will create an ROC Curve to visualize my model.

[49]:
```
logit_roc_auc = roc_auc_score(test_target, predictions)
fpr, tpr, thresholds = roc_curve(test_target, logistic_regression.
  ↪predict_proba(test_features_standardized)[:,1])
plt.figure()
```

```
plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit_roc_auc)
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.savefig('Log_ROC')
plt.show()
```



The goal of the ROC curve for a good fit model, is to have curve (blue) be as far away from the dashed red line as possible. Thisis the case here as there is a lot of area under the curve leaving this model, again, to be a good fit.

In the next Milestone I will bring these interpretations into some suggestions, results and ideas on how to leverage this model.

1.4 Conclusion.

When starting off this project, I was concerned with the lack of features in my dataset. There were only 3 features I had that could predict ad click-thru-purchase. This is something I haven't mentioned much throughout but wanted to address this in the conclusion. For a business stance, this model is still helpful with it being almost 90% accurate at predicting if someone will purchase a product from an ad. I want to investigate the feature that is most influencial and that is; age. With the highest correlation and the strongest coefficient of determination, the age variable influences the buying power of an ad. Something worth investigating further for the business is determining what age groups are doing a lot of the purchasing, then target even more people in that age range. This will most likely increase sales from that ad. The business should also take this as something to learn from, what additions/changes can they make to the ad that will get other features to be more involved. I mean the more you get that involved, the higher the accuracy of the model could possibly be.

These are all things to consider for the business and I hope they would take the insights and model into their process for who to target. This is the best thing about logisitc regression, being able to utilize features within the data to make a better prediction and outcome for the desired target. Having more features could make the process and model even better. There could be other features that could impact click-thru-purchase and that could increase the model's accuracy and impact. I would want to possibly see time of day, (depending on the product in the ad) season, country/state, etc. There would be more ideas that come to mind if I knew the product, but these are just a few. This was a fun project and I learned and practiced a lot for preparing, cleaning, and analyzing data through logistic regression.