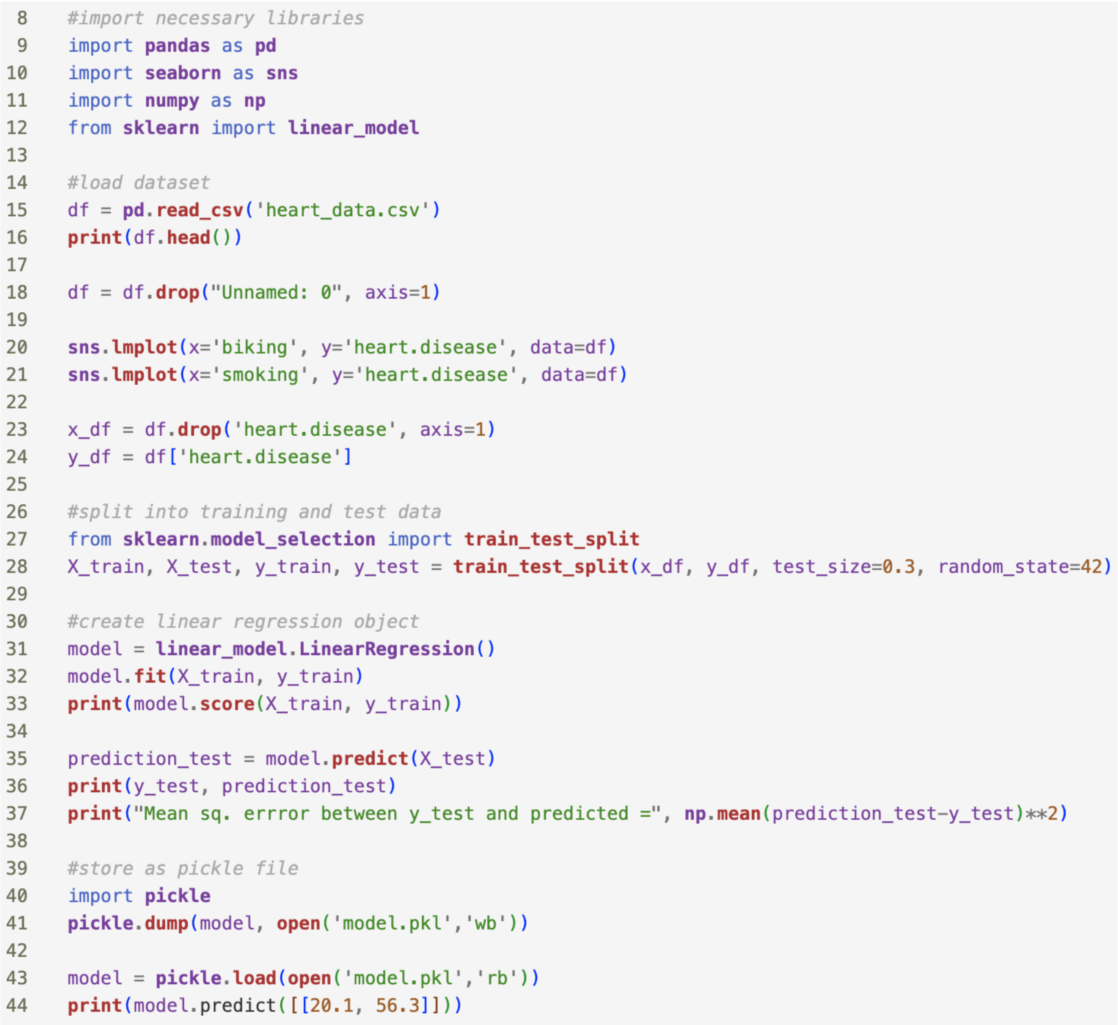
Project Report

Name: Model Deployment on Flask
Report date: September 27th 2022
Internship Batch: LISUM13:30
Version: 1.0
Report by: Devika Chandnani
Data storage location: https://github.com/devikachandnani/linear-regression-deployment-flask

Step 1, Creating the Model: "week4.py"

For this project, I will be using a dataset of an imaginary sample of 500 towns to explore the effect the independent variables, 'biking' and 'smoking' have on the variable 'heart disease' using a linear regression model.

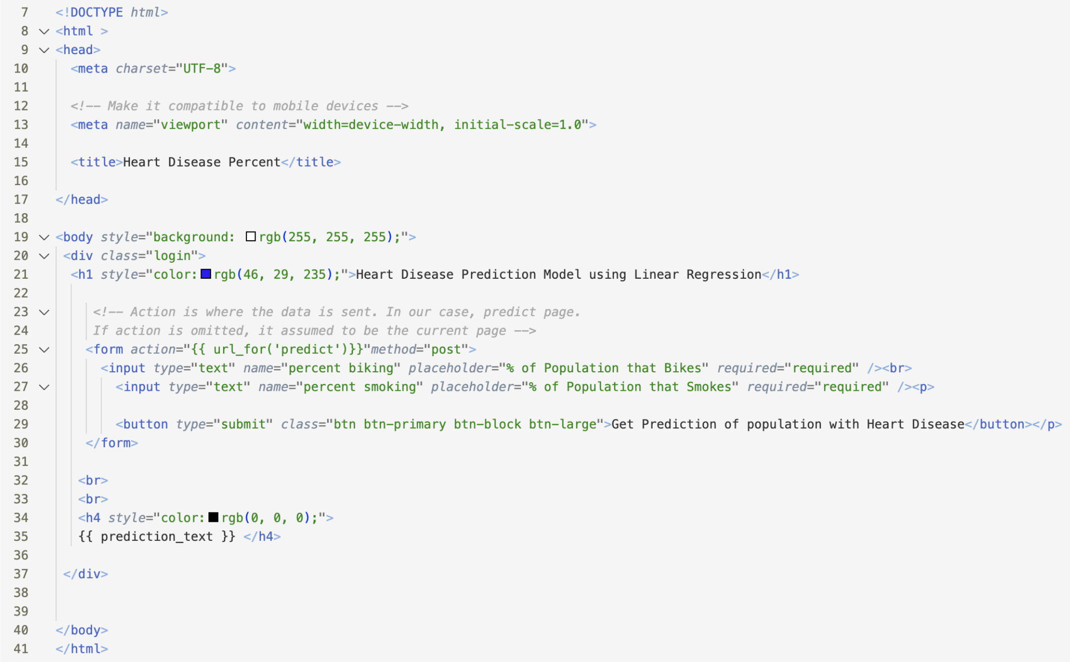
The Model is Fairly simple, at the end of the python file, the model is converted into a Pickle file.



```python
#import necessary libraries
import pandas as pd
import seaborn as sns
import numpy as np
from sklearn import linear_model

#load dataset
df = pd.read_csv('heart_data.csv')
print(df.head())

df = df.drop("Unnamed: 0", axis=1)

sns.lmplot(x='biking', y='heart.disease', data=df)
sns.lmplot(x='smoking', y='heart.disease', data=df)

x_df = df.drop('heart.disease', axis=1)
y_df = df['heart.disease']

#split into training and test data
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x_df, y_df, test_size=0.3, random_state=42)

#create linear regression object
model = linear_model.LinearRegression()
model.fit(X_train, y_train)
print(model.score(X_train, y_train))

prediction_test = model.predict(X_test)
print(y_test, prediction_test)
print("Mean sq. errror between y_test and predicted =", np.mean(prediction_test-y_test)**2)

#store as pickle file
import pickle
pickle.dump(model, open('model.pkl','wb'))

model = pickle.load(open('model.pkl','rb'))
print(model.predict([[20.1, 56.3]]))
```

Step 2, HTML Formatting: “index.html”

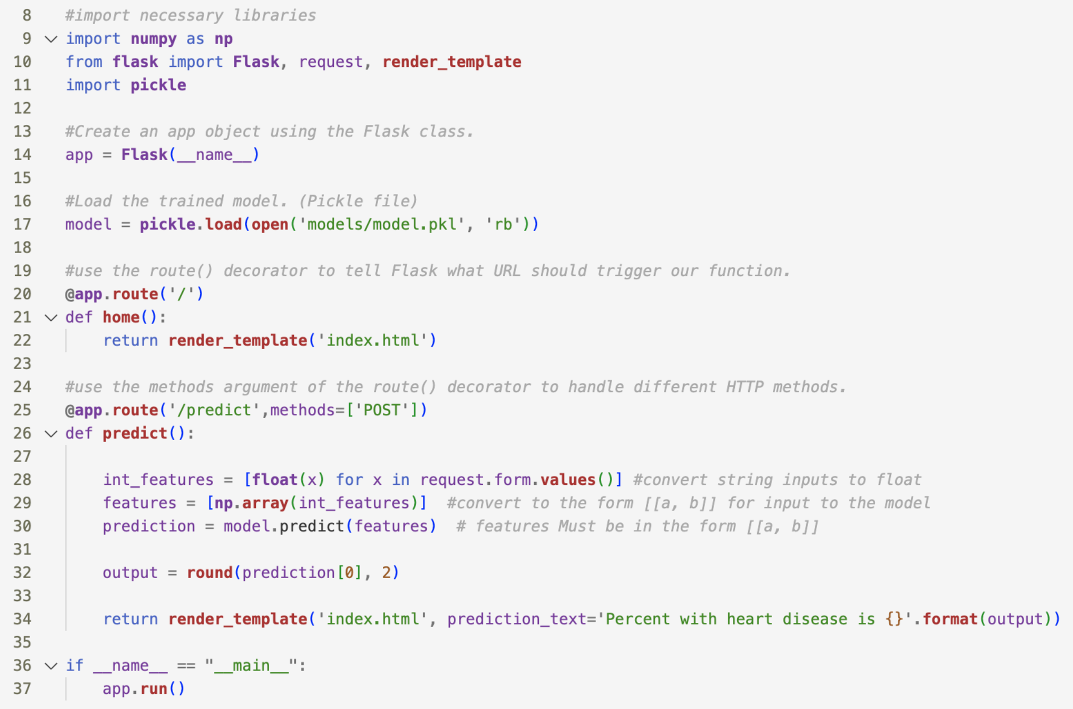
The next step is using a bootstrap model to format the webpage, only basic formatting was done, such as input placeholder texts, setting required fields, title, background color, title color, output formatting and color.



```html
<!DOCTYPE html>
<html >
<head>
  <meta charset="UTF-8">

  <!-- Make it compatible to mobile devices -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Heart Disease Percent</title>

</head>

<body style="background: □rgb(255, 255, 255);">
 <div class="login">
  <h1 style="color:■rgb(46, 29, 235);">Heart Disease Prediction Model using Linear Regression</h1>

     <!-- Action is where the data is sent. In our case, predict page.
     If action is omitted, it assumed to be the current page -->
    <form action="{{ url_for('predict')}}"method="post">
      <input type="text" name="percent biking" placeholder="% of Population that Bikes" required="required" /><br>
        <input type="text" name="percent smoking" placeholder="% of Population that Smokes" required="required" /><p>

        <button type="submit" class="btn btn-primary btn-block btn-large">Get Prediction of population with Heart Disease</button></p>
    </form>

   <br>
   <br>
   <h4 style="color:■rgb(0, 0, 0);">
   {{ prediction_text }} </h4>

 </div>


</body>
</html>
```

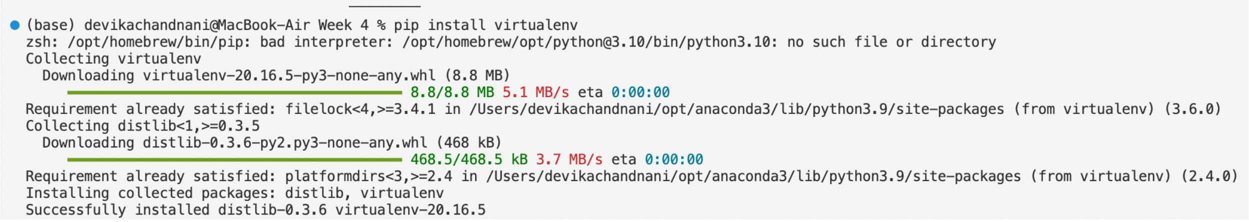
Step 3, Flask App Python File: “app.py”

Now we create the flask app.



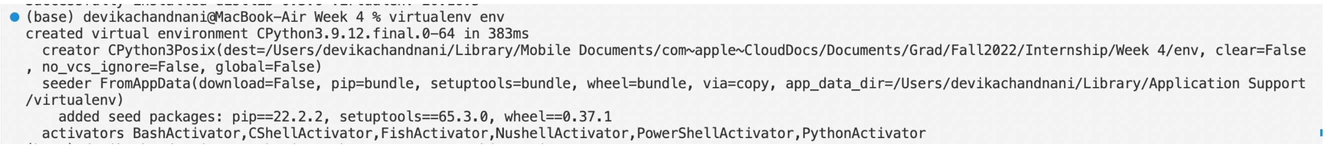
```python
#import necessary libraries
import numpy as np
from flask import Flask, request, render_template
import pickle

#Create an app object using the Flask class.
app = Flask(__name__)

#Load the trained model. (Pickle file)
model = pickle.load(open('models/model.pkl', 'rb'))

#use the route() decorator to tell Flask what URL should trigger our function.
@app.route('/')
def home():
    return render_template('index.html')

#use the methods argument of the route() decorator to handle different HTTP methods.
@app.route('/predict',methods=['POST'])
def predict():

    int_features = [float(x) for x in request.form.values()] #convert string inputs to float
    features = [np.array(int_features)]  #convert to the form [[a, b]] for input to the model
    prediction = model.predict(features)  # features Must be in the form [[a, b]]

    output = round(prediction[0], 2)

    return render_template('index.html', prediction_text='Percent with heart disease is {}'.format(output))

if __name__ == "__main__":
    app.run()
```

Step 4, Installing the Virtual Environment in Visual Studio Code:



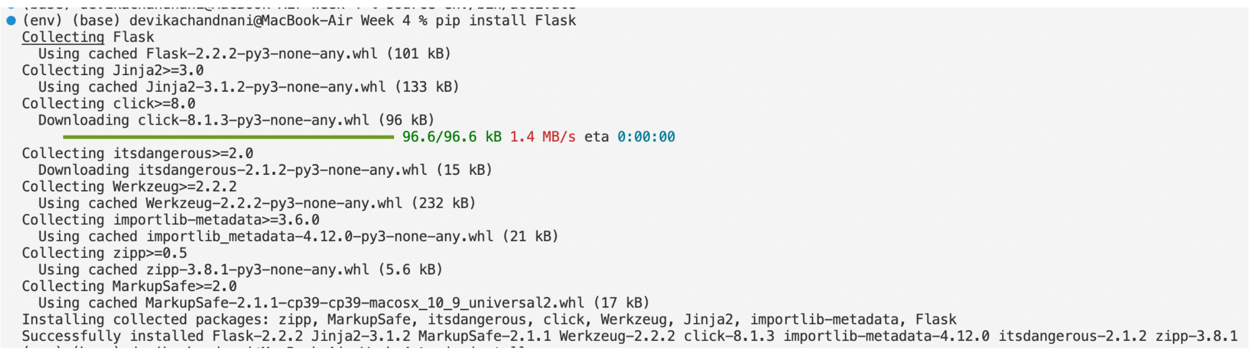
```
● (base) devikachandnani@MacBook-Air Week 4 % pip install virtualenv
  zsh: /opt/homebrew/bin/pip: bad interpreter: /opt/homebrew/opt/python@3.10/bin/python3.10: no such file or directory
  Collecting virtualenv
    Downloading virtualenv-20.16.5-py3-none-any.whl (8.8 MB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.8/8.8 MB 5.1 MB/s eta 0:00:00
  Requirement already satisfied: filelock<4,>=3.4.1 in /Users/devikachandnani/opt/anaconda3/lib/python3.9/site-packages (from virtualenv) (3.6.0)
  Collecting distlib<1,>=0.3.5
    Downloading distlib-0.3.6-py2.py3-none-any.whl (468 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 468.5/468.5 kB 3.7 MB/s eta 0:00:00
  Requirement already satisfied: platformdirs<3,>=2.4 in /Users/devikachandnani/opt/anaconda3/lib/python3.9/site-packages (from virtualenv) (2.4.0)
  Installing collected packages: distlib, virtualenv
  Successfully installed distlib-0.3.6 virtualenv-20.16.5
```

Step 5, Creating the Virtual Environment in Visual Studio Code:



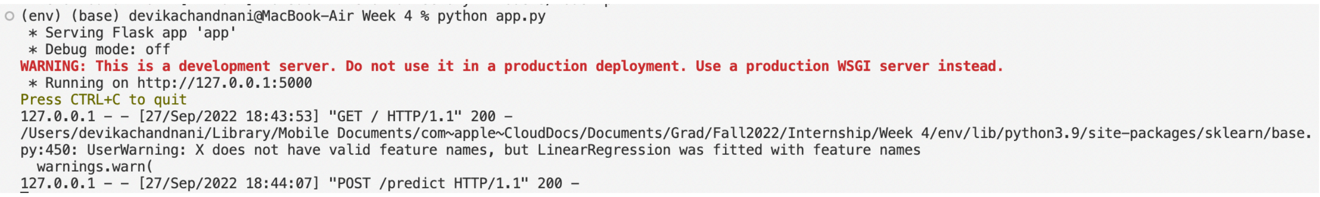
```
● (base) devikachandnani@MacBook-Air Week 4 % virtualenv env
  created virtual environment CPython3.9.12.final.0-64 in 383ms
    creator CPython3Posix(dest=/Users/devikachandnani/Library/Mobile Documents/com~apple~CloudDocs/Documents/Grad/Fall2022/Internship/Week 4/env, clear=False
  , no_vcs_ignore=False, global=False)
    seeder FromAppData(download=False, pip=bundle, setuptools=bundle, wheel=bundle, via=copy, app_data_dir=/Users/devikachandnani/Library/Application Support
  /virtualenv)
      added seed packages: pip==22.2.2, setuptools==65.3.0, wheel==0.37.1
    activators BashActivator,CShellActivator,FishActivator,NushellActivator,PowerShellActivator,PythonActivator
```

Step 6, Installing Flask in the Virtual Environment:



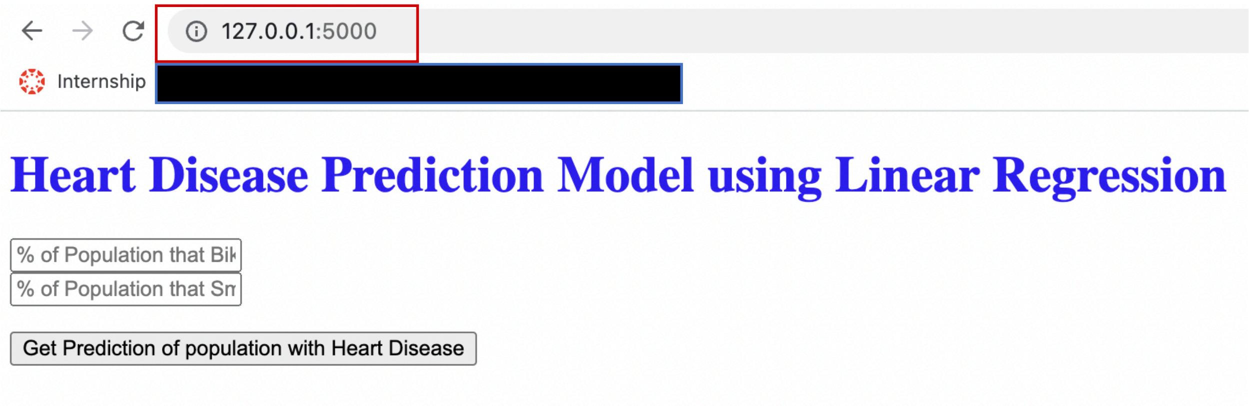
```
● (env) (base) devikachandnani@MacBook-Air Week 4 % pip install Flask
  Collecting Flask
    Using cached Flask-2.2.2-py3-none-any.whl (101 kB)
  Collecting Jinja2>=3.0
    Using cached Jinja2-3.1.2-py3-none-any.whl (133 kB)
  Collecting click>=8.0
    Downloading click-8.1.3-py3-none-any.whl (96 kB)
       ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 96.6/96.6 kB 1.4 MB/s eta 0:00:00
  Collecting itsdangerous>=2.0
    Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
  Collecting Werkzeug>=2.2.2
    Using cached Werkzeug-2.2.2-py3-none-any.whl (232 kB)
  Collecting importlib-metadata>=3.6.0
    Using cached importlib_metadata-4.12.0-py3-none-any.whl (21 kB)
  Collecting zipp>=0.5
    Using cached zipp-3.8.1-py3-none-any.whl (5.6 kB)
  Collecting MarkupSafe>=2.0
    Using cached MarkupSafe-2.1.1-cp39-cp39-macosx_10_9_universal2.whl (17 kB)
  Installing collected packages: zipp, MarkupSafe, itsdangerous, click, Werkzeug, Jinja2, importlib-metadata, Flask
  Successfully installed Flask-2.2.2 Jinja2-3.1.2 MarkupSafe-2.1.1 Werkzeug-2.2.2 click-8.1.3 importlib-metadata-4.12.0 itsdangerous-2.1.2 zipp-3.8.1
```

Step 7, Running the app.py file:



```
○ (env) (base) devikachandnani@MacBook-Air Week 4 % python app.py
   * Serving Flask app 'app'
   * Debug mode: off
  WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
   * Running on http://127.0.0.1:5000
  Press CTRL+C to quit
  127.0.0.1 - - [27/Sep/2022 18:43:53] "GET / HTTP/1.1" 200 -
  /Users/devikachandnani/Library/Mobile Documents/com~apple~CloudDocs/Documents/Grad/Fall2022/Internship/Week 4/env/lib/python3.9/site-packages/sklearn/base.
  py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
    warnings.warn(
  127.0.0.1 - - [27/Sep/2022 18:44:07] "POST /predict HTTP/1.1" 200 -
```

Step 8, opening the webpage and testing:

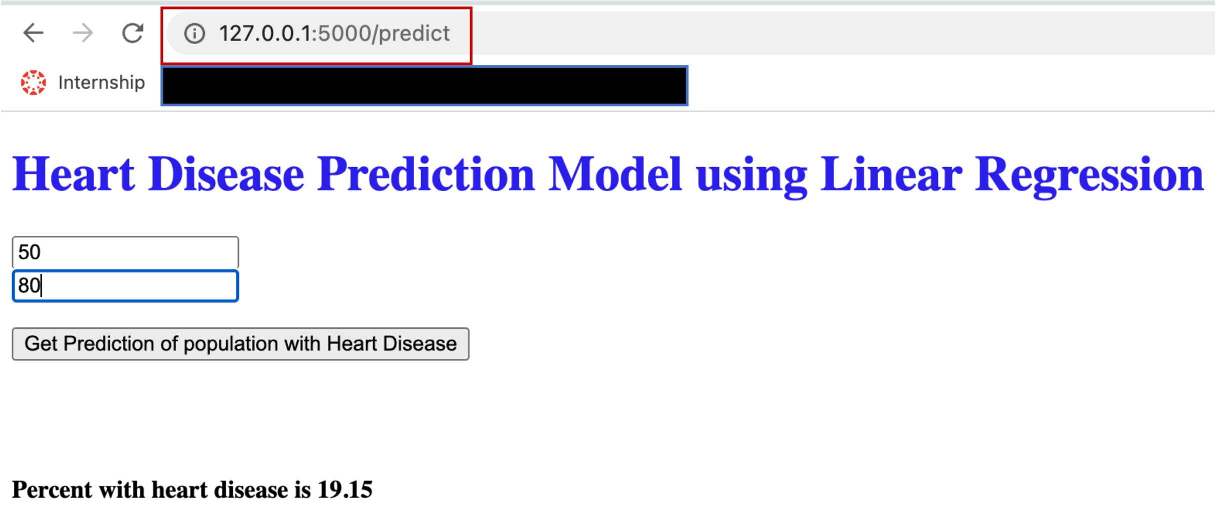


[redacted]

Heart Disease Prediction Model using Linear Regression

% of Population that Bik

% of Population that Sm



[redacted]

Heart Disease Prediction Model using Linear Regression

50

80

Percent with heart disease is 19.15