Comparing ‘scikit-learn’ to ‘statsmodels’ for Logistic Regression

Group Name: Live Lite
Group Members: Manasa Shivappa Ronur
Parvati Jayakumar
Saikripa Mohan
Ted Liu

Background

Obesity is becoming a major health issue in the US, and its numbers have been on the rise for the past few decades.

Understanding how lifestyle factors, particularly physical activity levels and dietary choices varies is crucial for finding ways to prevent and treat it effectively.

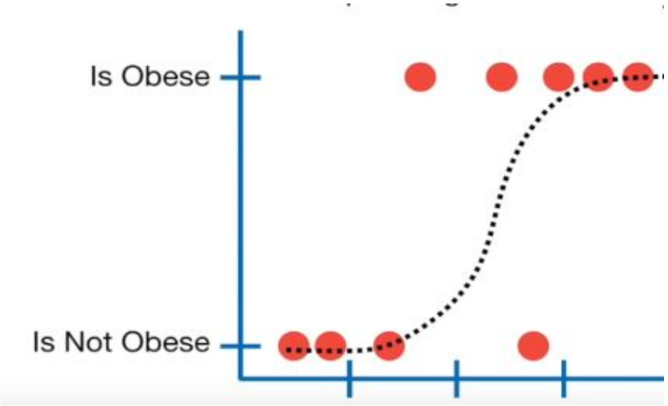
Our **Aim** is to identify the most suitable python package for designing a machine learning model to predict obesity risk.

Use Case

User wants to know whether or not they are on the risk of developing the obesity.

- **Input:** user enters data about self (anthropometric & lifestyle factors).
- **Output:** If user is at risk, a number predicting the percentage of obesity risk with current lifestyle.

Technology required: A library that supports the logistic regression model to enable prediction



Library Choices

Scikit-Learn

- **Original Author:** David Cournapeau
- **Released:** In 2007, Google Summer of Code

- A widely-used ML library in Python known for its versatility and user-friendly interface.
- Offers a comprehensive suite of tools for data preprocessing, modelling and deployment.
- Supported by extensive documentation and an active community, it easy to get started and find help when needed.

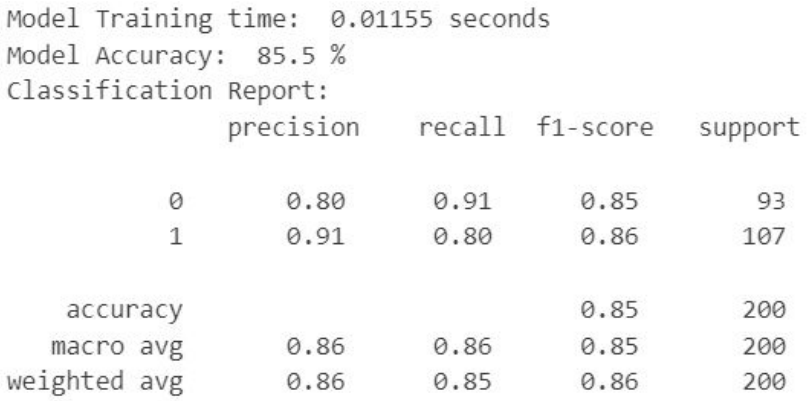
Statsmodels

- **Original Author:** Jonathan Taylor
- **Released:** In 2009, Google Summer of Code

- Python library focused on statistical modeling and analysis.
- Specializes in estimating and interpreting statistical models with precision.
- Supported by extensive documentation and an active community to facilitate statistical modeling and analysis tasks.

Comparing the Libraries

Scikit-Learn Logistic Regression



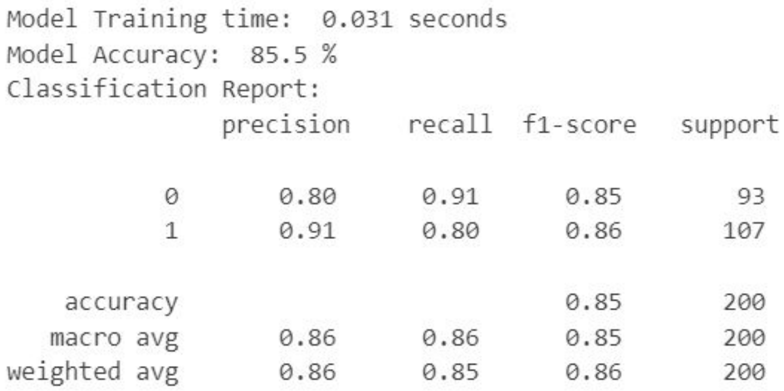
```
Model Training time:  0.01155 seconds
Model Accuracy:  85.5 %
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.91      0.85        93
           1       0.91      0.80      0.86       107

    accuracy                           0.85       200
   macro avg       0.86      0.86      0.85       200
weighted avg       0.86      0.85      0.86       200
```

- Easy for beginners (User-friendly interface and simple syntax) to get started with machine learning.
- Offers a wide range of ML solutions, prioritizing performance and scalability.

Statsmodels Logistic Regression



```
Model Training time:  0.031 seconds
Model Accuracy:  85.5 %
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.91      0.85        93
           1       0.91      0.80      0.86       107

    accuracy                           0.85       200
   macro avg       0.86      0.86      0.85       200
weighted avg       0.86      0.85      0.86       200
```

- Provides Advanced tools for estimating and interpreting statistical models.
- Offers a variety of summary statistics to assess the quality and reliability of the models, however it may not scale very well compared to scikit learn.

DEMO

Our choice: Scikit-learn Logistic Regression

If our priority is simplicity, scalability, and using a broad range of machine learning functionalities, scikit-learn might be a better choice.

If we require detailed statistical output, hypothesis testing, or are more concerned with the statistical properties of the model, statsmodels might be more suitable.

In our case, considering that we are primarily interested in predicting obesity risk and our usage involves straightforward prediction without an immediate need for extensive statistical diagnostics, scikit-learn might be the better choice. It's simpler to use, performs well, and offers a wide range of tools for model evaluation and deployment. However, if we later need to delve into the statistical properties of the model or require more advanced statistical tests, we could revisit statsmodels.

So, in conclusion, the scikit-learn is more appropriate for our use case due to its simplicity, performance, and suitability for predictive modeling tasks.

Drawback and Remaining concerns

Statistical summaries, confidence intervals, p-values, odds-ratios are not built in with scikit learn.

Remaining concerns:

If the dataset becomes complex when scaling, we will have to reevaluate the packages.

We need to compare the performance of Scikit with other packages such as Keras and tensorflow if we expand our scope of re- training model with user data