ASSIGNMENT 1

Prepared by :
1. Amogh Johri(IMT2017003)
2. Arjun Verma(IMT2017008)

Institute: IIIT Bangalore

Course: Machine Learning

Course Instructors:
1. Prof. G.Srinivasaraghavan
2. Prof. Neelam Sinha

Date of Submission : 16 Sept, 2019

Contents

1 Exploratory Data Analysis

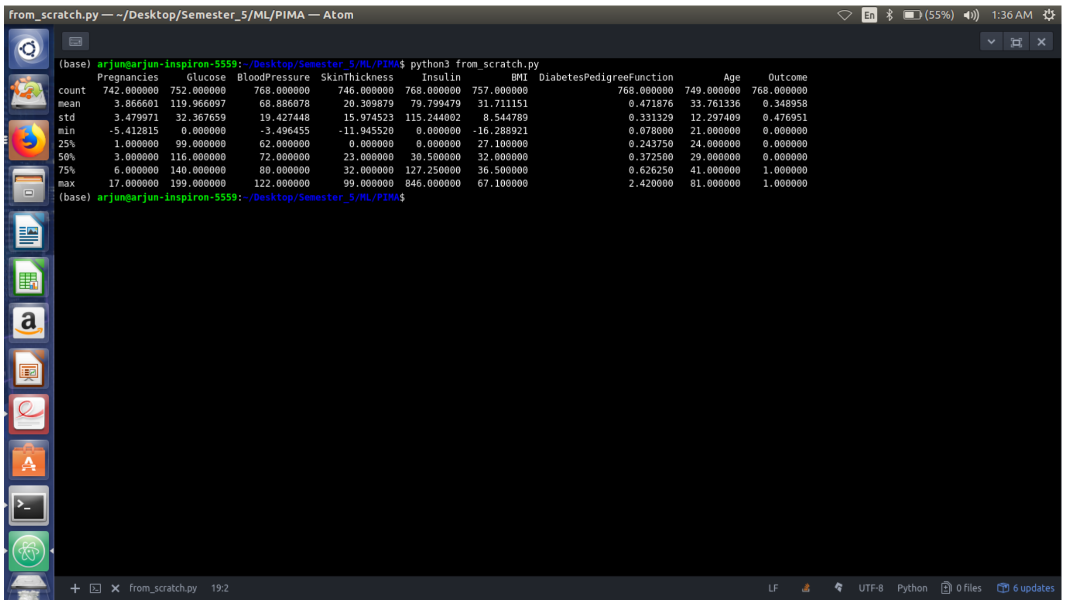
Exploratory Data Analysis refers to the critical process of performing initial investigations on data so as to discover patterns,to spot anomalies,to test hypothesis and to check assumptions with the help of summary statistics and graphical representations.

We performed EDA through the following methods:

- Dataset.describe()
- Correlaton Matrix
- Distribution Plots

1.1 Dataset.describe()

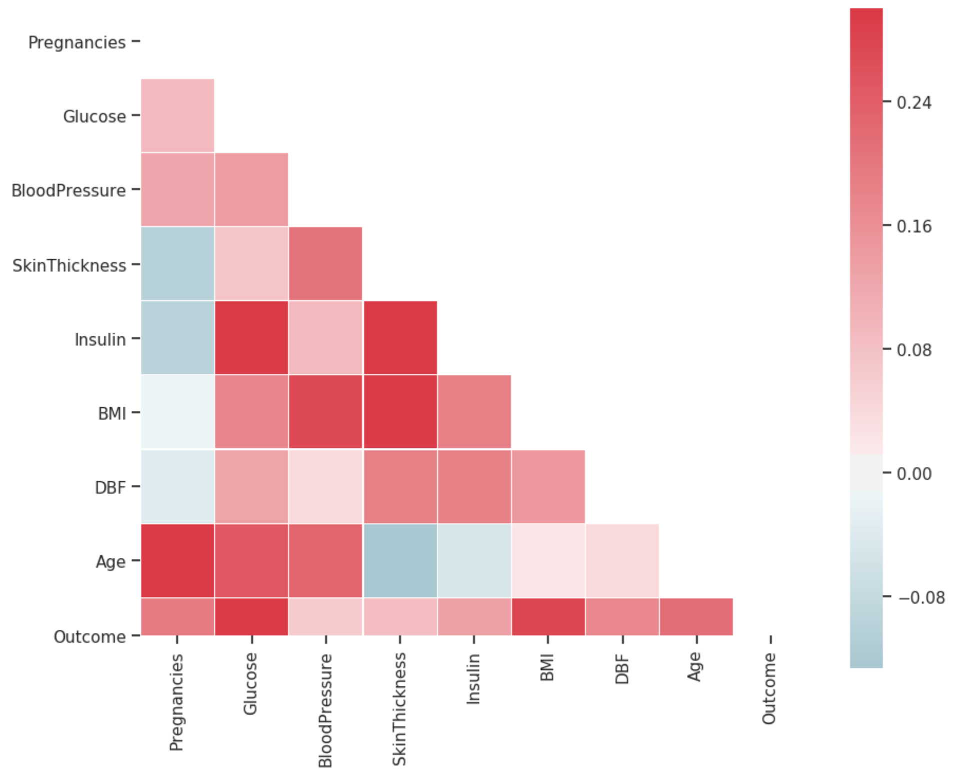
The describe() function in pandas returns the count, mean, standard deviation, minimum and maximum values and the quantiles of the data. By looking at the mean and median values for each attribute, certain observations can be made. Skewness in data can also be predicted by correctly inferring from these values. Predicting the skewness for the distribution of each attribute helped us in detecting outliers.



```
(base) arjun@arjun-inspiron-5559:~/Desktop/Semester_5/ML/PIMA$ python3 from_scratch.py
       Pregnancies     Glucose  BloodPressure  SkinThickness     Insulin         BMI  DiabetesPedigreeFunction         Age     Outcome
count   742.000000  752.000000     768.000000     746.000000  768.000000  757.000000                768.000000  749.000000  768.000000
mean      3.866601  119.966097      68.886078      20.309879   79.799479   31.711151                  0.471876   33.761336    0.348958
std       3.479971   32.367659      19.427448      15.974523  115.244002    8.544789                  0.331329   12.297409    0.476951
min      -5.412815    0.000000      -3.496455     -11.945520    0.000000  -16.288921                  0.078000   21.000000    0.000000
25%       1.000000   99.000000      62.000000       0.000000    0.000000   27.100000                  0.243750   24.000000    0.000000
50%       3.000000  116.000000      72.000000      23.000000   30.500000   32.000000                  0.372500   29.000000    0.000000
75%       6.000000  140.000000      80.000000      32.000000  127.250000   36.500000                  0.626250   41.000000    1.000000
max      17.000000  199.000000     122.000000      99.000000  846.000000   67.100000                  2.420000   81.000000    1.000000
(base) arjun@arjun-inspiron-5559:~/Desktop/Semester_5/ML/PIMA$
```

1.2 Correlation Matrix

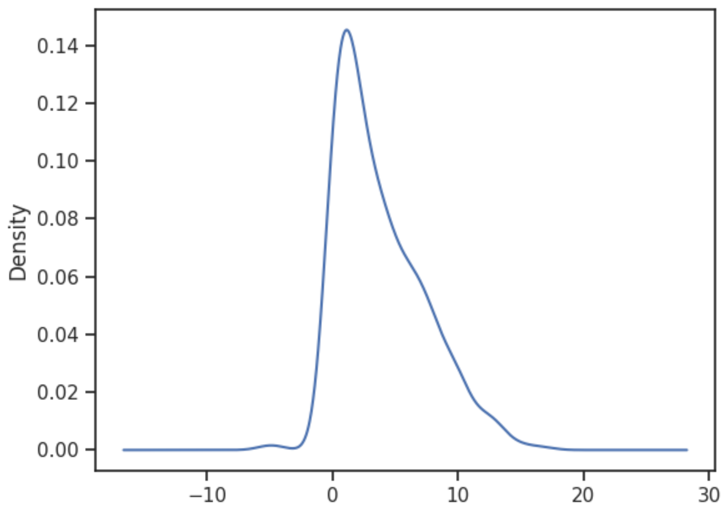
We have used linear regression as a method to impute missing values in some of our columns. To use linear regression, its necessary to remove correlated variables to improve your model. We used ".corr()" function and visualized the correlation matrix using a heatmap in seaborn. In short, we just wanted to check the dependencies of the attributes on each other.



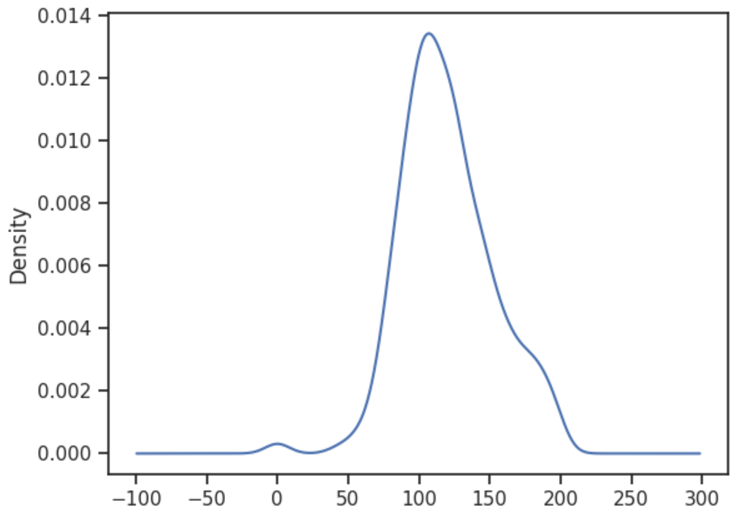
1.3 Distribution Plots

We used several in-built seaborn functions to visualize the attribute data points in various ways (swarmplot(),catplot(),distplot() are a few of those). We found that it is also a good practice to plot distribution graphs and look for skewness of features. Kernel density estimate (kde) is the tool that we used for plotting the shape of the distributions. The various plots obtained using kde are shown below:-

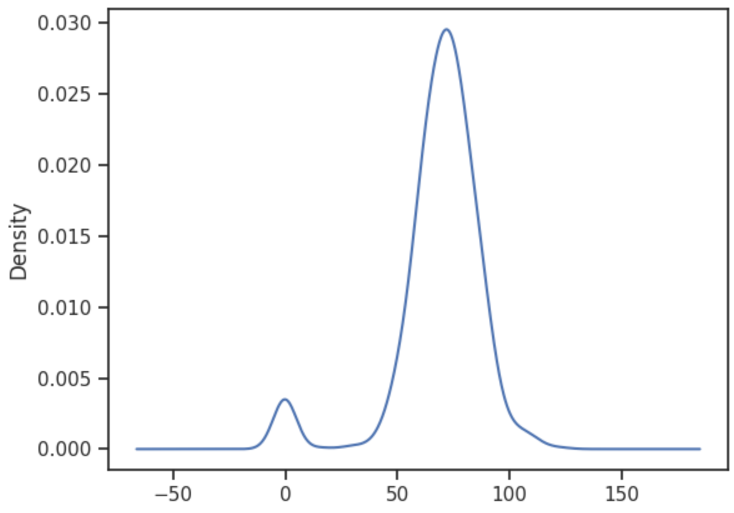
- **Distribution of Pregnancies**



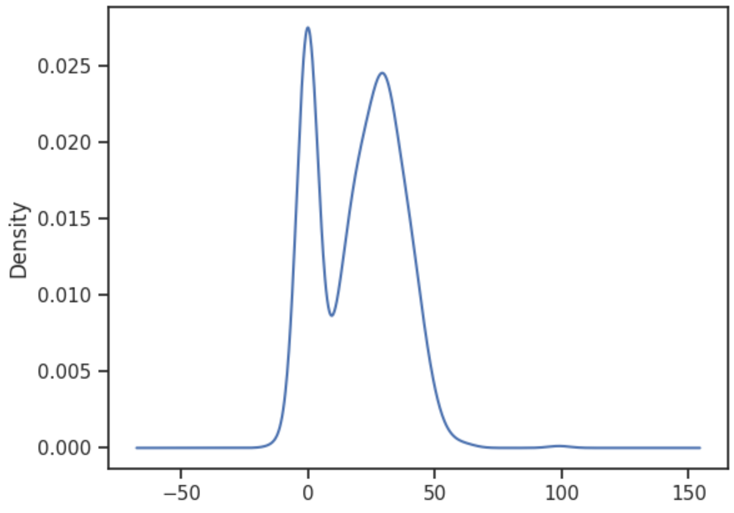
- **Distribution of Glucose**



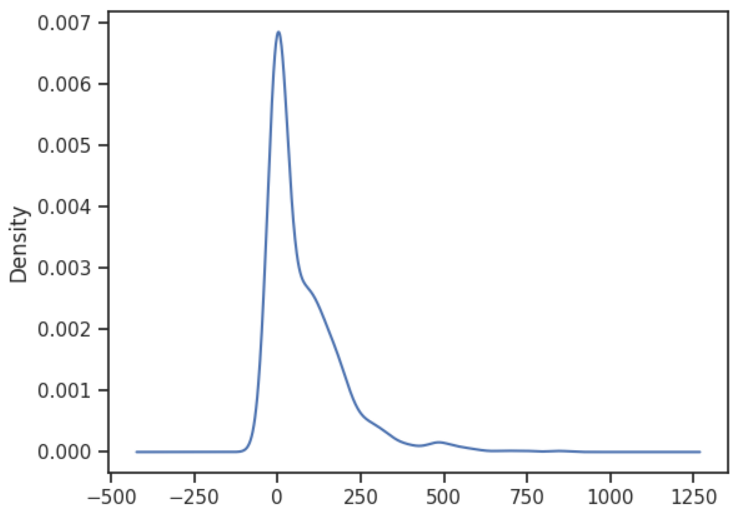
- **Distribution of BloodPressure**



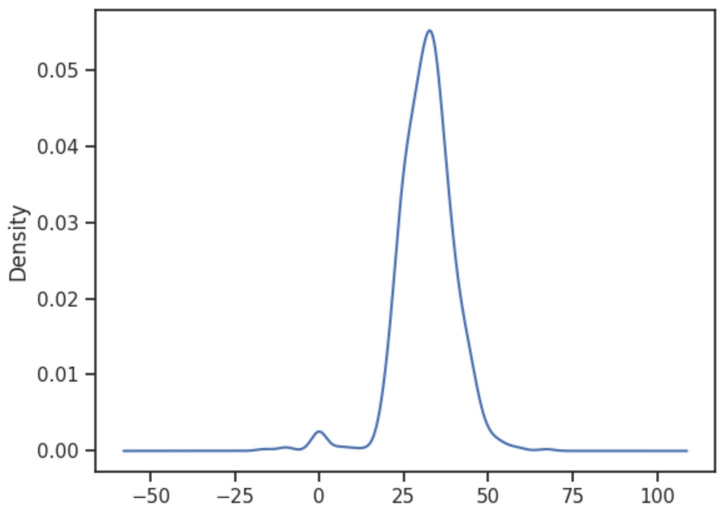
- **Distribution of SkinThickness**



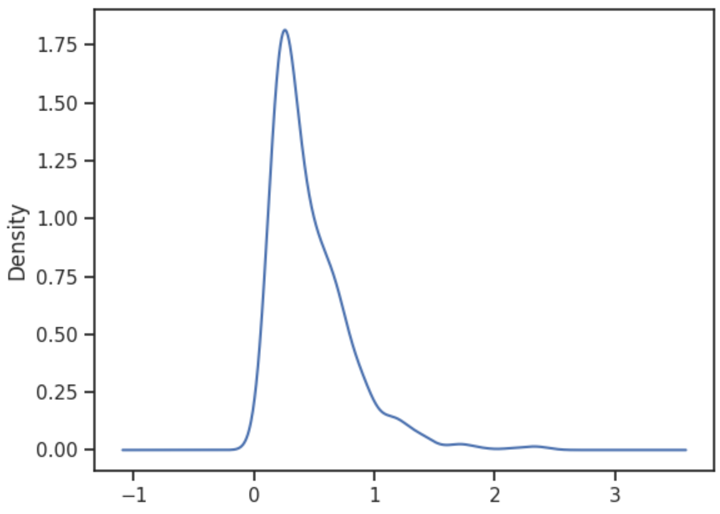
- **Distribution of Insulin**



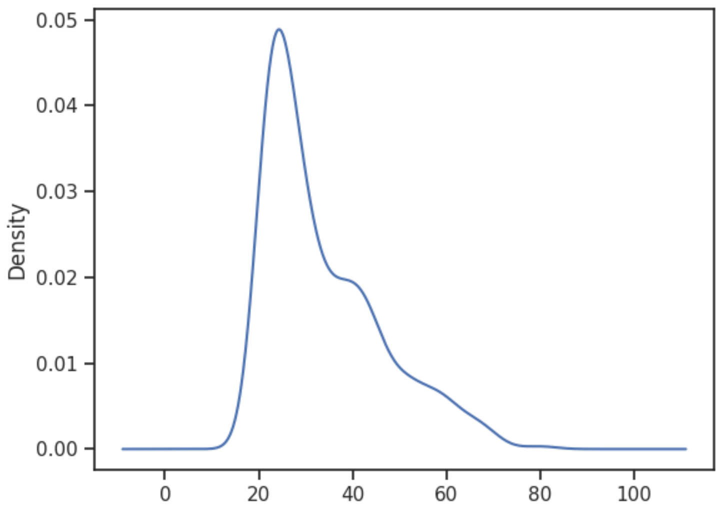
- **Distribution of BMI**



- **Distribution of DiabetesPedigreeFunction**



- **Distribution of Age**



2 Missing Data Handling and Data Preprocessing

We describe column-wise, the missing data handling and data preprocessing that we have applied on each column:-

- **Pregnancies:** The number of pregnancies is an important feature in order to predict diabetes as women who suffer from gestational diabetes are more likely to develop type-2 diabetes in the future (https://www.cdc.gov/reproductivehealth/maternalinfanthealth/diabetes-during-pregnancy.htm). Also, it shows a significant correlation value in the correlation matrix for the data. Hence, the column cannot be ignored. The column should contain only positive integral values. Hence, the following has been carried out:

  1. Negative values has been converted to their positive counterparts considering that a data collection error can result in a negative sign appearing.

  2. The fractional values has been floored to the nearest integer.

  3. In order to deal with outlier (very high values in this case), number of pregnancies has been upper-capped at 6 considering the average number of pregnancies for Indian women lies around 2.3 (http://worldpopulationreview.com/countries/total-fertility-rate/) so 6 shall prove to be a safe uppercapping value.

  4. For the NaN values in the field of pregnancies, a linear-regression model which takes age as the parameter has been utilized to predict the number of pregnancies. As due to socio-cultural reason, the number of pregnancies can become dependent on age and also due to the strong correlation shown by both the features in the correlation matrix.

- **Glucose:** The NaN values in the Glucose column has been taken care of by using the mean of the available data.

- **BloodPressure:** The blood-pressure is another important column for the prediction of diabetes. The values for this column should contain positive values. Also, normal blood-pressure lies aroun 80 and anything below 60 is considered abnormally low (https://www.everydayhealth.com/hypertension/when-is-low-blood-pressure-too-low/) hence, considering this a safe lower cap of 40 has been provided. In order to fill for the NaN values, a mean fitting has been carried out.

- **SkinThickness:** The values for SkinThickness has to be positive values. The following has been done for it:

  1. For the missing NaN values a median of the available values has been utilized as maximum value (60) is almost twice as much as the 75per-cent (31) which depicts that the data might get skewed to the right due to a few large values.

  2. Another method of using linear-regression with BloodPressure, Insulin, DiabetesPedigreeFunction and BMI as parameters was almost used to predict the SkinThickness but it affected the accuracy negatively. The attempt was made due to the large correlation values of SkinThickness with these features.

  3. Considering the average SkinThickness for Indian women (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5083983/), which lies around 35, a safe upper cap of 60 has been utilized.

- **Insulin:** Insulin values correspond to values of insulin 2 hours after glucose administration, which should usually be around 16-166 mIU/L (https://emedicine.medscape.com/article/2089224-overview). Considering this, a safe upper cap of 300 has been provided. For the NaN values, median of the available data has been taken considering that the 75per-cent value is 130 where as the max is going all the way to 600.

- **BMI:** The BMI needs to be a positive value lying in the range of 15-50. The following has been done for it:

  1. The data has been lower capped to 13 and upper capped to 55 (https://en.wikipedia.org/wiki/Body_mass_index#Scalability).

  2. For the missing NaN values, median has been taken as 75per-cent value is 36.5 which is almost half of the max, being 67.

- **DiabetesPedigreeFunction:** The DiabetesPedigreeFunction has been upper capped at 1.5 and the NaN values has been fixed using the median as the 75per-cent value is 0.59 which is much lower compared to the max being 2.3.

- **Age:** Age shows a strong-correlation with BloodPressure and Glucose and hence, a linear regression model with BloodPressure and Glucose as features has been utilized to predict the age for the NaN values.

3 Features Extraction

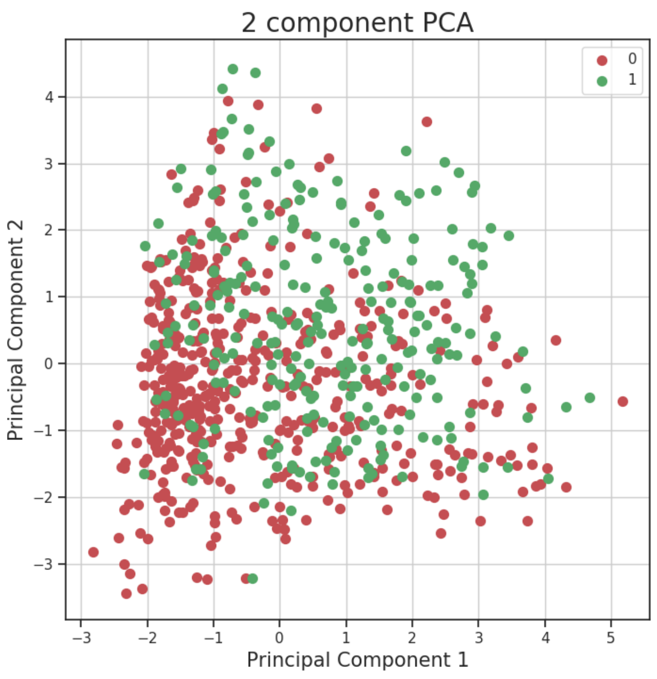
3.1 Principal Component Analysis

PCA is a technique for reducing the number of dimensions in a dataset whilst retaining most information. It is using the correlation between some dimensions and tries to provide a minimum number of variables that keeps the maximum amount of variation or information about how the original data is distributed. Our dataset needs to be standardized before we can apply PCA on it. We use sklearn's StandardScaler to help us standardize the dataset's features onto unit scale (mean = 0 and variance = 1) which is a requirement for the optimal performance.

Notice the code submitted has .95 for the number of components parameter. It means that we are choosing the minimum number of principal components such that 95% of the variance is retained.

We fit PCA on the training set only. After fitting on the training set, we apply the mapping (transform) to both the training set and the test set. The various ML algorithms are then applied to the transformed Data

The visualization of data in 2D using the first two principal components obtained after applying PCA is shown below:-



3.2 Feature engineering

We have engineered only one new feature. Using the 'Age' feature, we constructed a new-feature - 'AgeGroup', which creates groups of 10. The reason for doing this was so that we could group certain other attributes such as BloodPressure, Glucose, Pregnancies into characteristic age groups and then determine an average value of these characteristics for each group and then use those values to substitute the missing values.

4 Model Building

The final model is the **Ensemble Model**, a combination of Logistic Regression and SVM (with a Linear Kernel).

The proposed model gives an accuracy of 0.773 with a standard deviation of 0.029 averaged over 10,000 random splittings of the data-set where 80 per-cent of the data was utilized as the training set while the rest 20 per-cent was given as the test-set.

The accuracies of applying only Logistic Regression, only SVM and the Ensemble Model of Logistic regression and SVM are given below for comparison:-

1. Logistic Regression

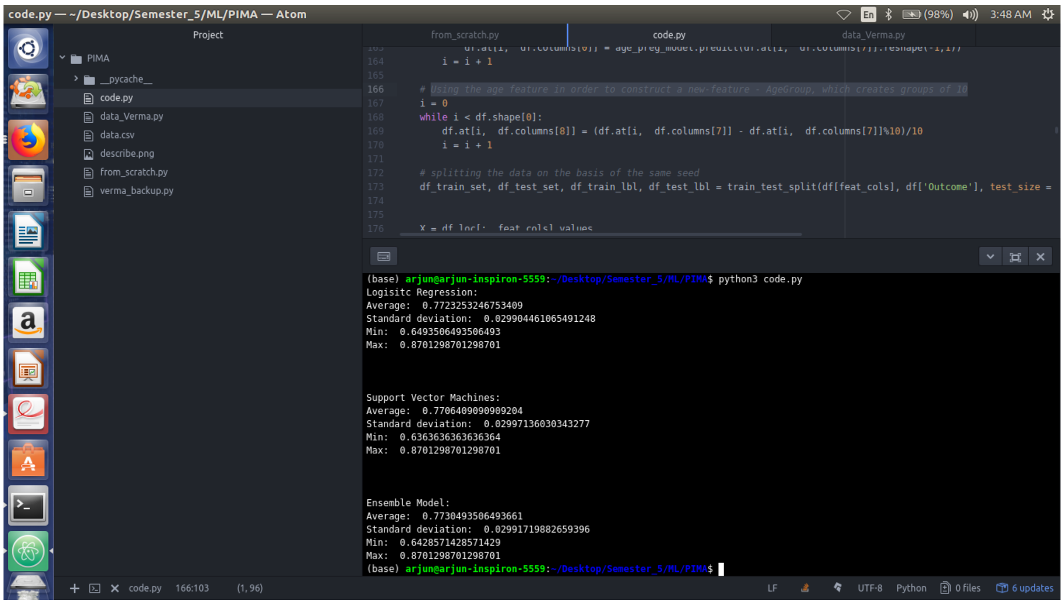
Model	**Logistic Regression**
Average Accuracy	0.7723253246753409
Standard Deviation	0.029904461065491248
Maximum Accuracy	0.8701298701298701

2. SVM

Model	**SVM**
Average Accuracy	0.7706409090909204
Standard Deviation	0.02997136030343277
Maximum Accuracy	0.8701298701298701

3. Ensemble Model (Logistic Regression + SVM)

Model	Ensemble Model
Average Accuracy	0.7730493506493661
Standard Deviation	0.02991719882659396
Maximum Accuracy	0.8701298701298701



```python
            df.at[i,  df.columns[0]] = age_preg_model.predict(df.at[i,  df.columns[7]].reshape(-1,1))
        i = i + 1

    # Using the age feature in order to construct a new-feature - AgeGroup, which creates groups of 10
    i = 0
    while i < df.shape[0]:
        df.at[i,  df.columns[8]] = (df.at[i,  df.columns[7]] - df.at[i,  df.columns[7]]%10)/10
        i = i + 1

    # splitting the data on the basis of the same seed
    df_train_set, df_test_set, df_train_lbl, df_test_lbl = train_test_split(df[feat_cols], df['Outcome'], test_size =


    X = df.loc[:,  feat_cols].values
```

```
(base) arjun@arjun-inspiron-5559:~/Desktop/Semester_5/ML/PIMA$ python3 code.py
Logisitc Regression:
Average:  0.7723253246753409
Standard deviation:  0.029904461065491248
Min:  0.6493506493506493
Max:  0.8701298701298701



Support Vector Machines:
Average:  0.7706409090909204
Standard deviation:  0.02997136030343277
Min:  0.6363636363636364
Max:  0.8701298701298701



Ensemble Model:
Average:  0.7730493506493661
Standard deviation:  0.02991719882659396
Min:  0.6428571428571429
Max:  0.8701298701298701
(base) arjun@arjun-inspiron-5559:~/Desktop/Semester_5/ML/PIMA$
```

The predictions from Logistic Regression and SVM have been combined in a fashion that, the model outputs positive for Diabetes only when both Logistic Regression and SVM output positive for Diabetes and in all other cases the model outputs negative. This has been done as 70 percent of the data that is being dealt with corresponds to a negative test for Diabetes. The accuracy of the Ensemble Model is higher than both the individual models in most cases, and marginally lower in a very few cases. Thus, by looking at the average accuracies provided above, the model proposed by us is the one with the highest average accuracy- The Ensemble Model.