Voting Classifier – Diabetes

The main objective is to predict whether the people in the dataset has diabetes. Because in this example we focused on improving the accuracy of our prediction, we used an ensemble method – Voting Classifier – to combine the results of the base models. Two of the base models we have used are also ensemble models (Random Forest and Adaptive Boosting). We have used two versions of Voting Classifiers – with and without weights.

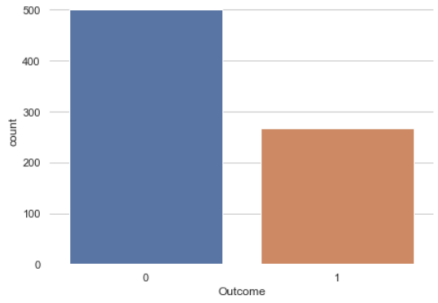
Data Source: https://www.kaggle.com/uciml/pima-indians-diabetes-database

Analysis

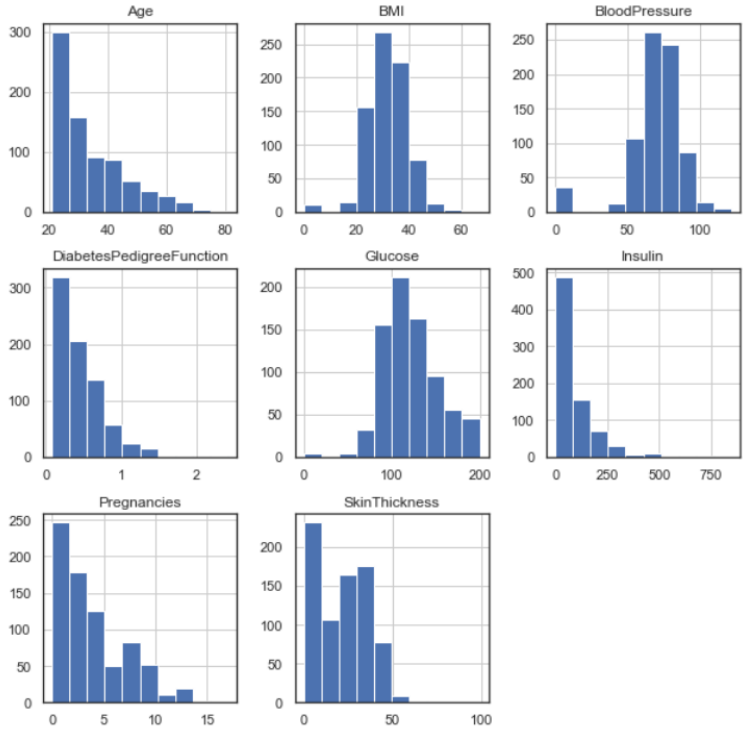
Exploratory Data Analysis

Before performing EDA, we checked the dimension of the dataset and data types of the variables and examined the dataset for null values.

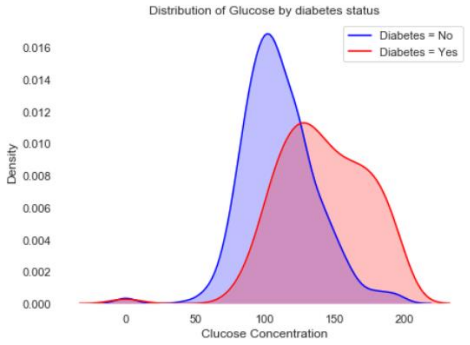
1. About one third of the people in the dataset has diabetes, and this split-up of the 'Outcome' will enable our models to predict more accurately for both the classes (1 and 0) – as compared with our earlier datasets in which one of the classes formed only about 10% of the entire data.

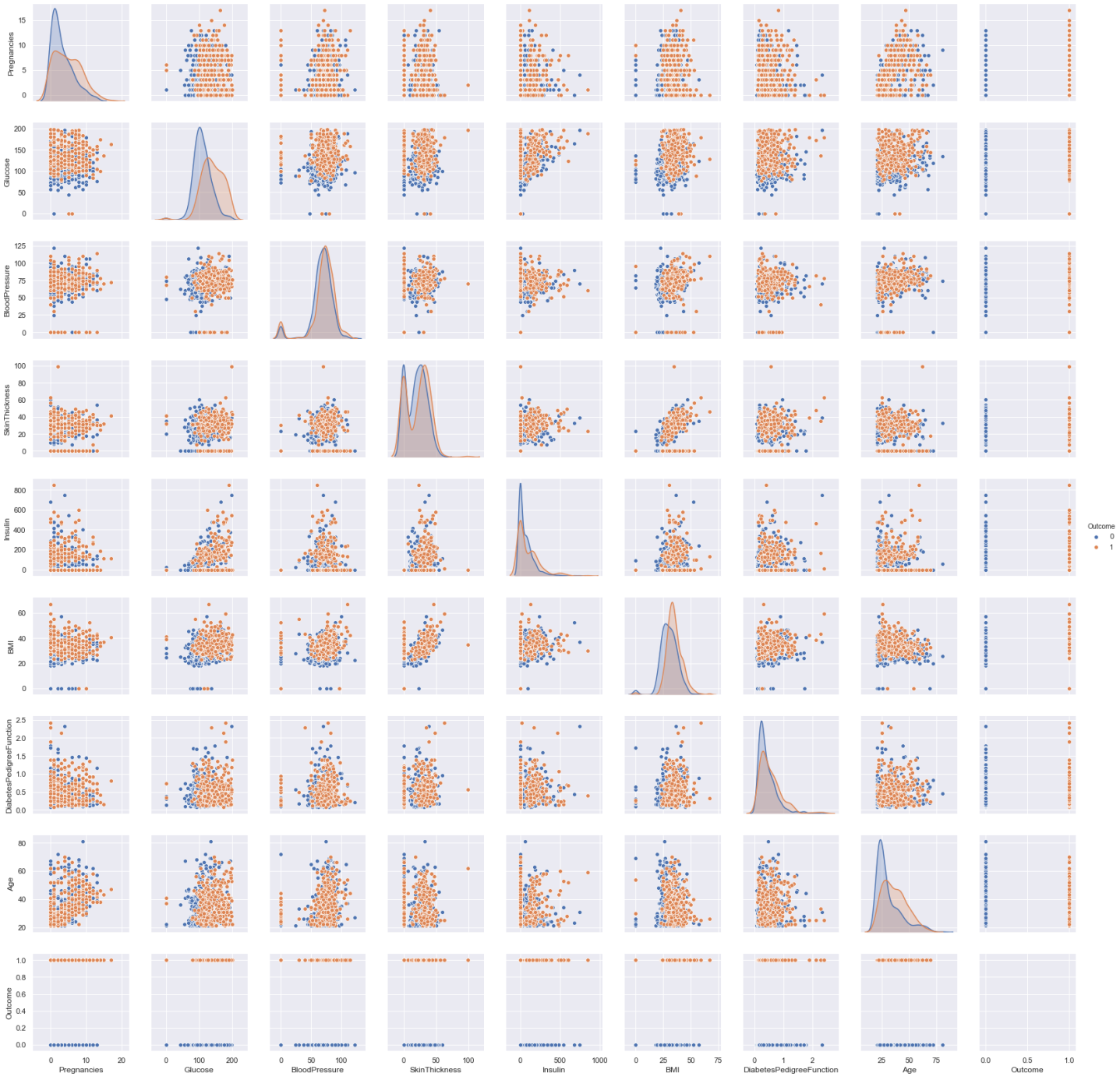


2. We plotted the distribution of the features that we will try to incorporate in our models

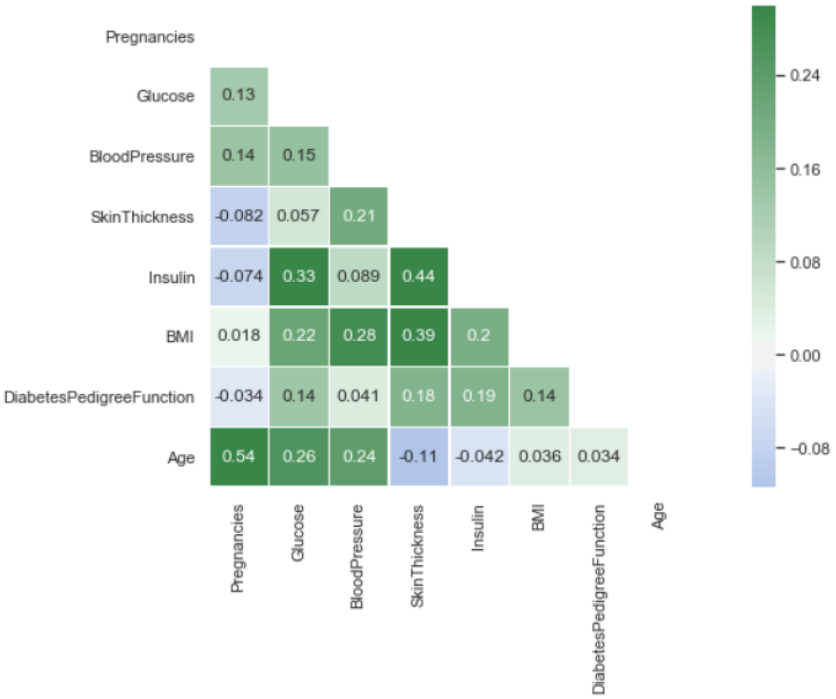


3. We displayed one KDE plot to show the distribution of Glucose Concentration separately for people with and without Diabetes, and then we used Pair Plot to plot the distribution (for with and without diabetes) of other features as well. Pair Plots also show scatter plots of the variables.





4. Then we plotted a correlation matrix to show the correlation between different variables. This plot allows us to know which features have high correlation with others and thus to exclude the highly correlated ones from the model; however, none of the variables had a strong correlation with any of the others.



Models and Performance

1. To build our model, we made 30% of the dataset test data and the remainder training. Moreover, we also scaled the features using standardization because the features varied in magnitude and range. Scaling enabled us to have same magnitude levels for all the features.

Out[95]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Out[108]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.639947	0.848324	0.149641	0.907270	-0.692891	0.204013	0.468492	1.425995
1	-0.844885	-1.123396	-0.160546	0.530902	-0.692891	-0.684422	-0.365061	-0.190672
2	1.233880	1.943724	-0.263941	-1.288212	-0.692891	-1.103255	0.604397	-0.105584
3	-0.844885	-0.998208	-0.160546	0.154533	0.123302	-0.494043	-0.920763	-1.041549
4	-1.141852	0.504055	-1.504687	0.907270	0.765836	1.409746	5.484909	-0.020496

2. Finally, we developed predict models using the following classifiers
    a. Logistic Regression
    b. Decision Tree
    c. KNN
    d. Random Forest
    e. Adaptive Boosting

We have already used two ensemble methods - Random Forests (Averaging) and Adaptive Boosting (Boosting) – as our base models. To improve accuracy, we combined different classifiers using a Voting Classifier (with and without weights), which is also an ensemble method.

Following are the accuracies of the base models and the Voting Classifier.

```
Model Accuracy


Logistic Regression: 77.92%
Decision Tree: 74.46%
KNN: 77.92%


Averaging Method
Random Forest: 77.92%


Boosting Method
AdaBoost: 72.73%


Voting Classifiers
Voting Classifier without Weights: 80.52%
Voting Classifier with Weights: 81.39%
```

- Among the base models, Logistic Regression, KNN, and Random Forest performed the best.
- Voting Classifier without weights improved the accuracy to 80.52%
- Voting Classifier with weights slightly improved the accuracy to 81.39%. Following weights were given to the different base models. Heavier weights were assigned to the better performing models.

Model	**Weight**
Logistic Regression	2
Random Forest	2
KNN	2
Decision Tree	1
Adaptive Boosting	1