# DATA SCIENCE FOR ALL – BATCH 3 CAPSTONE PROJECT

**BNPL**

# A Report on Classification of High Converting Customers using ML Algorithm

**Icon

Description automatically generated with medium confidence**

**Medha Wagh**

**Problem Statement**: The retail industry has reported the emergence of “Buy now pay later” (BNPL) within the mobile app platform business. The idea is for the retail company to tie up with FinTech, who would, in turn, provide a free credit line for the mobile users of the retail company. There are two kinds of Buy Now Pay Later options. One is the Interest-free option available to the consumer, which is sub-vented by the brand. In this model, brands bear the cost because Fintech has enabled a sale for them, which otherwise would have been impossible if the consumer did not have an incentive to pay it in installments at no extra cost. The second kind of BNPL is where a customer must pay a nominal amount as interest. Again, the interest is minor, and the total amount is split into smaller installments, making the purchase more affordable for the consumer. There is a need to use the churn analysis to predict if a customer will ignore or enroll in our BNPL feature, target only high converting customers, and leave out the rest.

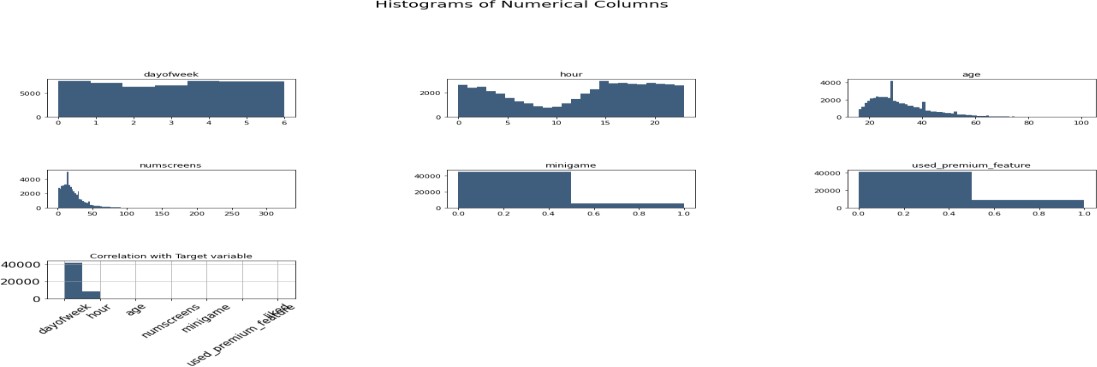
**Dataset 1 - mobileusebasedata.csv**: This dataset contains all the data of existing usage data of mobile app users. Important features are:

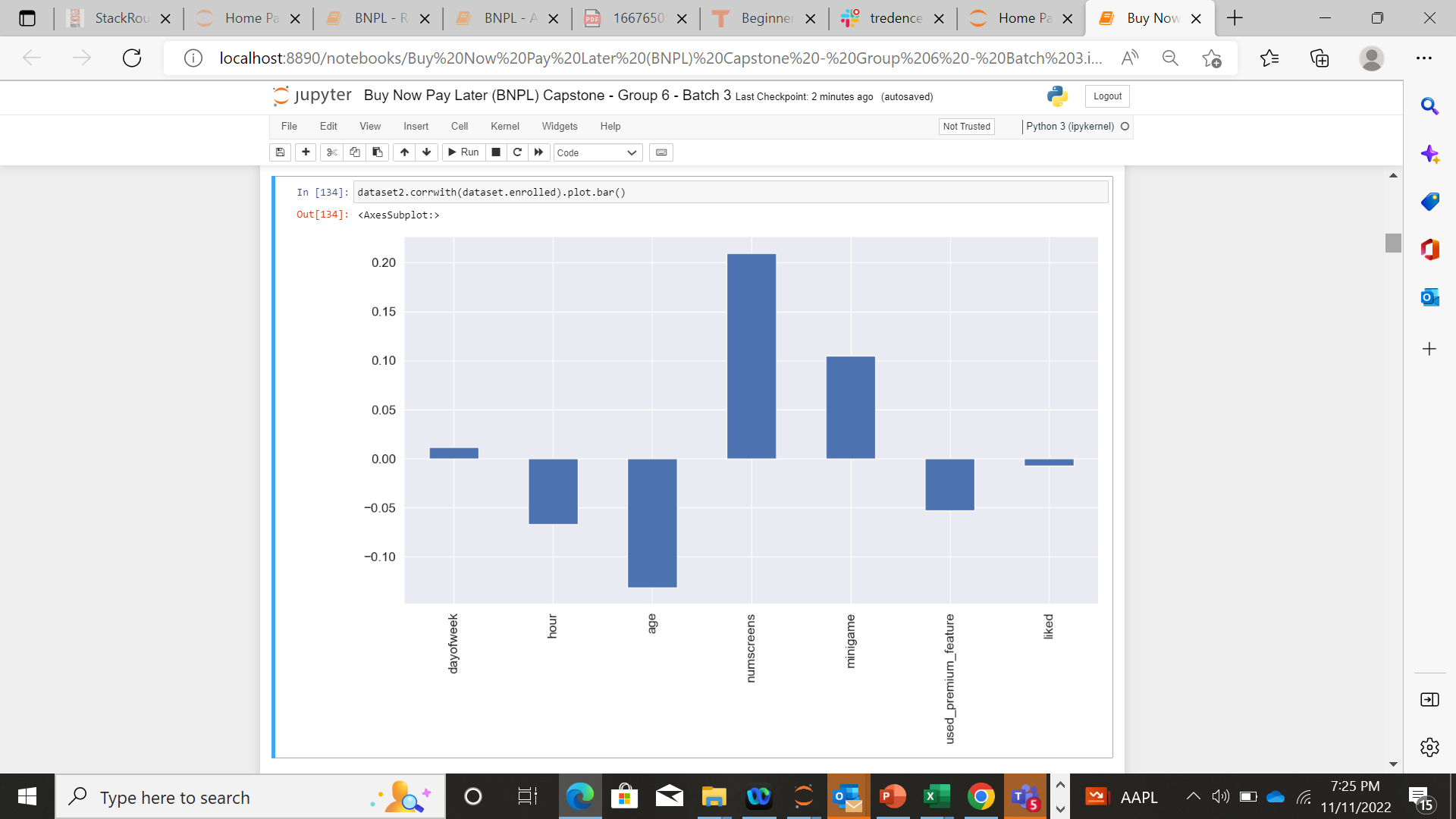
* User: user number as per installation order
* Dayofweek: day of the week app installed like Sunday is 0
* Numsscreen: number of screens browsed
* Minigame: a small game made on BNPL feature
* Used\_premium\_feature: Bought premium feature earlier
* Enrolled: If the user finally enrolled in BNPL (target label)
* First\_open: First time the app was opened by that user
* Enrolled\_date: Data enrolled in BNPL feature

**Dataset 2 - most\_used\_screens.csv**: This is a list made by the Business Analyst to list the screens used in a serial order. The list can be used to aggregate the number of times certain screen were opened. Example: (Savings 1 + Savings 2 + ... + Savings 10) is number of times the Savings screen was opened. This dataset was used to perform some feature engineering to reduce the number of columns.

# EDA:

**Histogram of all the numerical columns:**





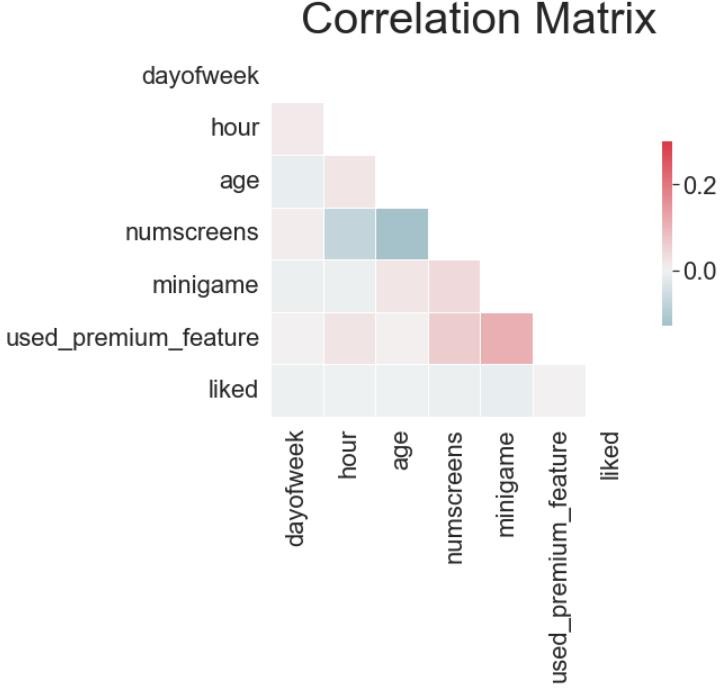
# Inference:

* Least number of users visited on Tuesday followed by Wednesday and the number of users were approximately same on rest.
* The ages of the customers seem to be skewed to the left of the mean age. The 20-40 age group tend to visit app quite a lot of time as compared to 60-80 age group.
* Peoples are highly engaged during 3pm to 3am and less during 3am to 3pm.
* Most of the people don't play mini games and haven't used premium features.
* Hours, dayofweek, minigame, premium features cannot be considered as a numerical data.
* The other two follow a normal distribution but are skewed towards their right side.
* As the correlation with histogram is not very clear, we have plotted a bar graph showing the correlation between different features and the target variable. The 'numscreens' and 'age' features show a higher positive and negative correlation respectively.

**Normal Distribution:**

Hours, dayofweek, minigame, premium features cannot be considered as a numerical data. The other two follow a normal distribution but are skewed towards their right side

# Correlation Matrix of all Numerical Values:



**Inference:** The predictors are not highly correlated to each other which shows that there is no Multi collinearity between the predictors. When there is multi collinearity, It becomes difficult for the model to estimate the relationship between each independent variable and the dependent variable independently because the independent variables tend to change in unison. Multicollinearity reduces the precision of the estimated coefficients, which weakens the statistical power of the model.

While the features don't really show a very high level of correlation amongst themselves, it seems like the features ['minigame','used\_premium\_feature'] shows a slightly higher positive correlation whereas the features ['numscreens','age'] show a negative correlation. This shows that younger people have a very short window of usage. Older people seem to spend a longer time exploring the app. Similarly, users who are already loyal as they have purchased a premium feature previously, tend to try out new features like minigames.

# Data Pre-Processing:

We are focusing only on the high converting customer, who are effectively taking less than 48 hours in Enrolling from the date of opening the app for the first ever time. The user is not considered as Enrolled if they have taken more than 48 hours to Enroll for BNPL.

The Top Screens are grouped based on their application and are effectively added as new features in the dataset which eventually explains the number of visits of the respective user to the respective screen.

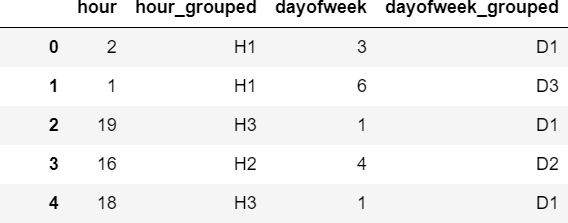
# Handling Categorical Variable:

The hour feature is grouped in the following format:

* 0 – 8 hours -> H1
* 8-16 hours -> H2
* 16-24 hours -> H3

The dayofweek feature is grouped in the following format:

* Monday, Tuesday, Wednesday -> D1
* Thursday, Friday -> D2
* Saturday, Sunday -> D3



Dummy Features are created for the Categorized Feature with 1s and 0s as entries.

# Outliers Detection:

Out of the 50 columns, most of the columns had outliers but all these columns had most entries with 0’s and the

remaining entries with numerical values. Thus, in these columns, removing outliers that are lying outside the

(80th Percentile + 1.5 IQR) meant eliminating all the values which are having non-zero value. Thus, we can conclude that the imbalance is causing more data to be lying outside of the expected range and removing them could lead in information loss.

# Normality Check with Shapiro Test:

According to Shapiro test all the features are normal in nature.

# Skewness:

Few of the columns had skewness beyond acceptable range (-3 to +3), and those columns were later removed to check how the model is getting improved.

# Scaling:

Standard Scaler and Min-Max Scaler were implemented individually to try out different combinations and arrive at the best model. Scaling of the data makes it easy for a model to learn and understand the problem so that the weight of all the data is equal. Machine Learning algorithms require all features to be in the same range to function properly, or they'll tend to pay more attention to some features rather than the other.

# Modelling and Evaluation:

To derive the best model to predict the Enrollment of users, we have used multiple methods upon which the modelling has been performed and the best precision score has been taken.

**Why Precision Score:** The emphasis is on picking the user who would enroll rather than checking on users who

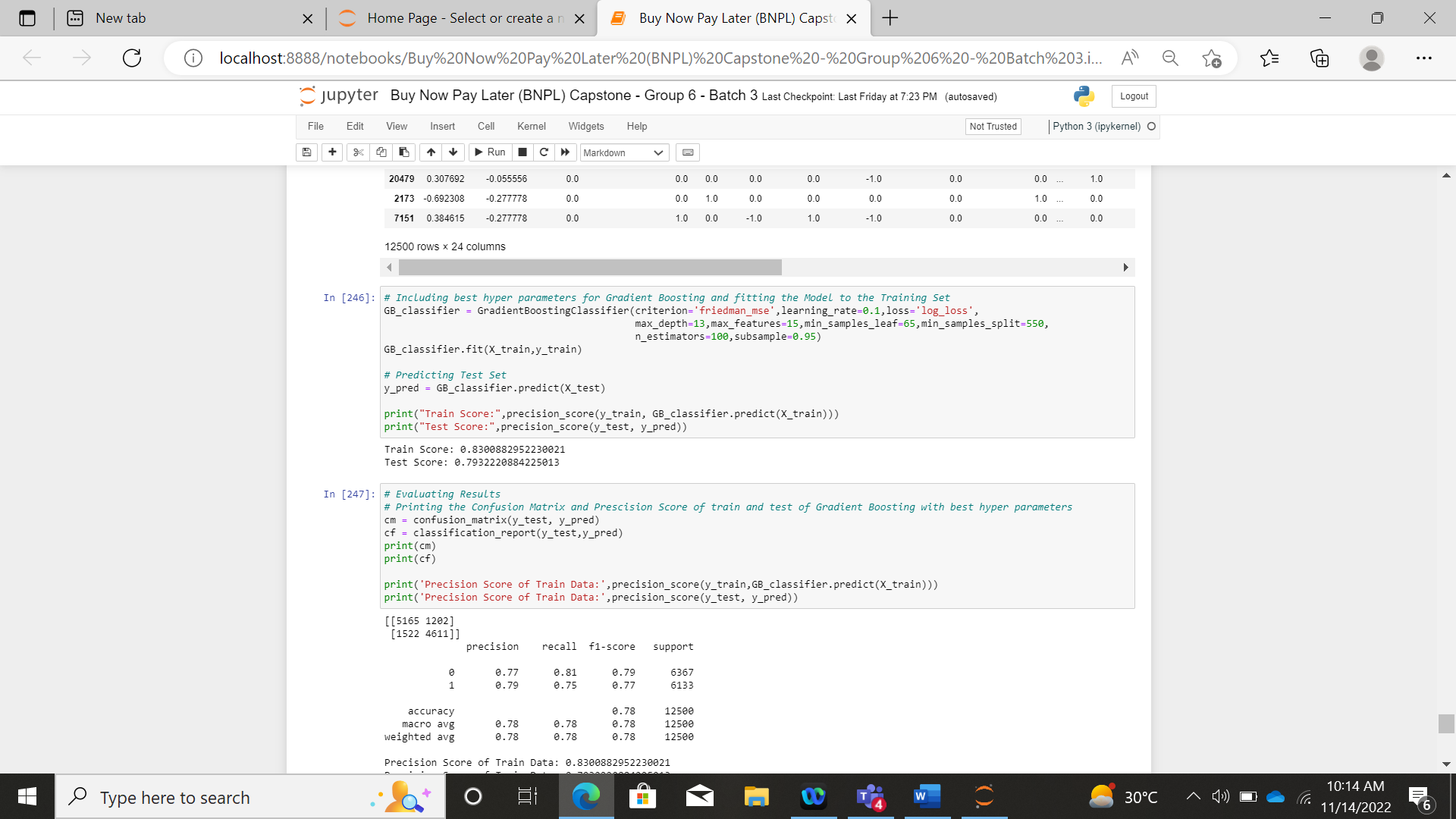
wouldn’t, thus the emphasis is on predicting the 1s correctly rather than 0s.

The output of various algorithms across different cases are:

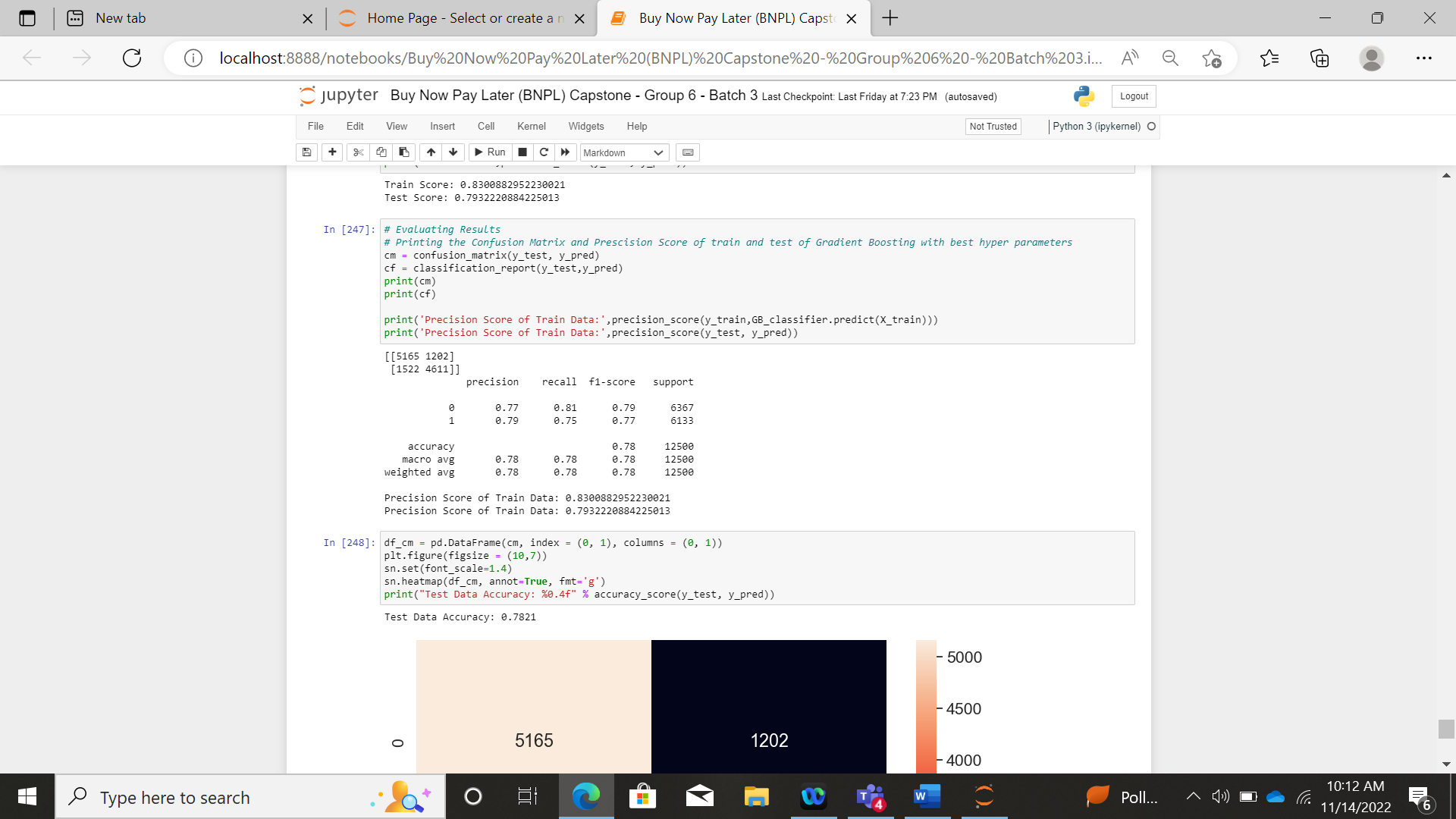
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Precision %** | | |
| **Models** | **Split** | **All Columns- 24** | **All Columns-24 with GSCV** | **GSCV with PCA - 15** |
| Gradient Boosting | Train | 79.41 | 83.04 | 83.96 |
| Test | 79.50 | 79.32 | 77.31 |
| AdaBoost | Train | 76.12 | 75.80 | 75.49 |
| Test | 76.02 | 76.06 | 75.20 |
| Logistic Regression | Train | 75.88 | 75.79 | 74.40 |
| Test | 75.55 | 75.56 | 74.10 |

Out of all the Algorithm, Gradient Boosting (the model with 24 columns with GSCV) is giving the best Precision Score – Train score of 83.04 and test score of 79.32. The minimal difference between the numbers shows that there is no underfitting or overfitting and the model can predict the 1s in unseen data by 79.32%.

The parameters used in this model is



The Confusion Matrix and Classification Report are as follows:



**Conclusion:**

On comparing the precision scores of the above build models, we find that the precision scores for the Gradient Boosting model with GSCV and without PCA gives us best precision percentage for the Test data. Hence we are choosing that model as the best model.