Bank Marketing Analysis Milestone Report

Financial organizations often struggle with differentiating themselves from competitors, who often use similar tactics, market to the same demographics, and offer similar or even the same rates and services. Using different marketing strategies for bank can help differentiate yourself from competitors and move past them by offering new technologies, offering the same services in different ways, or otherwise targeting new people in new ways.

Sales and marketing area of any business is responsible for finding clients, making sales and generating revenue. In this project we will analyze on Marketing strategy with mentioned attributes from the available bank data for sales and marketing team with the target outcome of which features helped them in opening a new term deposit as a result of this campaign.

Data from a marketing campaign run by Bancode Portugal is examined. **The campaign's aim was to increase customers' subscription rates to fixed-term deposit products, such as CDs**. Using data science tools and techniques, applying number of machine learning algorithms to answer the question**: How can banks successfully market these products in the most efficient way possible and with the highest possible rate of success?**

Programming in Python:
Python provides a number of packages and libraries for the convenience of the programmer. The whole project is coded using Python 3. Packages/libraries used are numpy for array manipulation, pandas for dataframe operations, and matplotlib and seaborn for visualization.

Data Cleaning and Exploratory Analysis:
Exploratory data analysis of data set and how to handle missing or unknown values are explained in data wrangling section. Subplot to draw visualization of the data representation and methods to obtain clean data for further analysis are mentioned in this section.

Inferential Statistical Analysis:

This section we define the feature and target variables and check the correlation of the target variable with other attributes by plotting a heatmap and do some null hypothesis testing.

Dataset taken from Kaggle.com (Bank Marketing Dataset)

DATA:

1 - **age**: (numeric)
2 - **job**: type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown')
3 - **marital**: marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed)
4 - **education**:
(categorical:"basic.4y","basic.6y","basic.9y","high.school","illiterate","professional.course","university.degree","unknown")
5 - **default**: has credit in default? (categorical: 'no','yes','unknown')
6 - **housing**: has housing loan? (categorical: 'no','yes','unknown')
7 - **loan**: has personal loan? (categorical: 'no','yes','unknown')

Related with the last contact of the current campaign
9 - **contact**: contact communication type (categorical: 'cellular','telephone')
10 - **month**: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
11 - **day_of_week**: last contact day of the week (categorical: 'mon','tue','wed','thu','fri')
12 - **duration**: last contact duration, in seconds (numeric).

Other attributes
13 - **campaign**: number of contacts performed during this campaign and for this client (numeric, includes last contact)
14 - **pdays**: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
15 - **previous**: number of contacts performed before this campaign and for this client (numeric)
16 - **poutcome**: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
17 - **emp.var.rate:** employment variation rate - quarterly indicator (numeric)
18 - **cons.price.idx:** consumer price index - monthly indicator (numeric)
19 - **cons.conf.idx:** consumer confidence index - monthly indicator (numeric)
20 - **euribor3m:** euribor 3 month rate - daily indicator (numeric)
21 - **nr.employed:** number of employees - quarterly indicator (numeric)
22 - **y** - has the client subscribed a term deposit? (binary: 'yes','no')

From the historical marketing campaign data set, we will identify the patterns that will help us find conclusions in order to develop future strategies and create intelligent targeting system :

- Find the best strategies to improve on next marketing campaign.
  (Predict **Y** based on the historical data, any correlation of **Y(term deposit)** with attributes like job/marital status/education/credit(housing,loan))
- Draw some insights on how financial institution have a greater effectiveness for future campaigns.
  (correlation of **poutcome** (success rate) on other attributes for historical data)
- Determine customer among the sample population that will most likely open term deposit accounts.
  (how **poutcome** and **y** relation if any based on what type of customers be targeted).
- The analysis on these data should suggests how the best strategies to improve next marketing campaign.

Data Wrangling Techniques:

The first step is to load the dataset into a dataframe for easy manipulation and exploration using the pandas package. The 'duration' feature was dropped due to the risk of data leakage. This feature measures the length of the phone call between the bank's marketing representative and the customer.

The next step was to explore and clean the categorical variables, rename the columns and check the categorical variables (Job, Marital, Education, HasCredit, HouseLoan, PersonalLoan, ContactType, LastContMonth, LastContDay, PrevOut, TermDeposit) for missing values.

Plots for each were produced that looked at their relative frequency as well as normalized relative frequency. In Python, these graphs are created using the subplot (fig 1).

```
categorical_variables = ['Age', 'Marital', 'Job', 'Education', 'HasCredit', 'HouseLoan', 'PersonalLoan', 'ContactType', 'LastCont
nrows=4
ncols=int(len(categorical_variables)/nrows)
fix,ax2d=plt.subplots(nrows,ncols, figsize=(30,30))

fig.subplots_adjust(wspace=0.6, hspace=6.0)
ax=np.ravel(ax2d)

for count,col in enumerate(categorical_variables):
    ax[count].bar(df[col].value_counts().index,df[col].value_counts().values)
    ax[count].legend(loc=1)
    ax[count].set_title(col, fontsize= 20)
    ax[count].set_xticklabels(df[col].value_counts().index, rotation=40, fontsize= 12)
plt.show()
```

There are unknown values for many variables in the Data set. One way to handle is to discard the row but that would lead to reduction of data set which wouldn't serve the purpose of building accurate and realistic prediction model. Another way is to infer the value from other variables, however it doesn't guarantee that all the missing values will be address but majority of them will be cleaned up for analysis. We will start cleaning data by updating the unknown values to Nan.

```
categorical_variables = ['Marital', 'Job', 'Education', 'HasCredit', 'HouseLoan', 'PersonalLoan', 'ContactType', 'LastContMonth',
for col in categorical_variables:
    df.ix[df[col]=='unknown',col] = np.nan
```

Variables with Nan values are : Education, Job, HasCredit, HouseLoan, PersonalLoan and Marital. However the Marital status has few unknown values, significant ones are Education, Job, HouseLoan and PersonalLoan. We will try to see the pattern for these missing values.

We will write a function which will return the Variable 1 groupby Variable 2 unique value counts as DataFrame.
We will use this function to check the missing/unknown values and see if can draw ay intuitions in filling the Nan values.

```
def var_test(df,f1,f2):
    var1 = list(df[f1].unique())
    var2 = list(df[f2].unique())
    dataframes = []
    for e in var2:
        dfv2 = df[df[f2]==e]
        dfv1 = dfv2.groupby(f1).count()[f2]
        dataframes.append(dfv1)
    xx=pd.concat(dataframes, axis=1)
    xx.columns=var2
    xx=xx.fillna(0)
    return dfv2
```

```
var_test(df,'Job','Education')
```

Inferring Education from Jobs: From the above table it can be seen that people with management will usually have a university degree, so we can replace the 'unknown' with 'university degree'. Similarly job with 'services' education as 'high.school', job with 'housemaid' education as 'basic.4y'.
Similarly we can also infer the jobs from education where 'Education' = 'basic.4y' or 'basic.6y' or 'basic.9y' with job as 'blue-collar', if Education is 'professional.course' the job = 'technician'.
It would also make sense to replace the unknown values for job where age > 60 as 'retired'.

Missing Values: From the available dataset description, missing values or NaNs are encoded as '999'. From the above screen it is clear that only PreviousDay has majority of missing values.
To deal with this variable, we will remove numerical variable PreviousDay and replace with additional categorical variables as following categories: pdays_missing (0 for contacted before and 1 for not previously contacted) , pdays_less_5, pdays_betw_5_15 and pdays_greater_15.

```
df.loc[(df.Age>60) & (df.Job.isnull()) ,'Job'] = 'retired'
df.loc[(df.Education.isnull()) & (df.Job=='management'), 'Education'] = 'university.degree'
df.loc[(df.Education.isnull()) & (df.Job=='services'), 'Education'] = 'high.school'
df.loc[(df.Education.isnull()) & (df.Job=='housemaid'), 'Education'] = 'basic.4y'
df.loc[(df.Job.isnull()) & (df.Education=='basic.4y'), 'Job'] = 'blue-collar'
df.loc[(df.Job.isnull()) & (df.Education=='basic.6y'), 'Job'] = 'blue-collar'
df.loc[(df.Job.isnull()) & (df.Education=='basic.9y'), 'Job'] = 'blue-collar'
df.loc[(df.Job.isnull()) & (df.Education=='professional.course'), 'Job'] = 'technician'
```

```
df['pdays_missing'] = 0
df['pdays_less_5'] = 0
df['pdays_betw_5_15'] = 0
df['pdays_greater_15'] = 0
df['pdays_missing'][df['PreviousDay']==999] = 1
df['pdays_less_5'][df['PreviousDay']<5] = 1
df['pdays_betw_5_15'][(df['PreviousDay']>=5) & (df['PreviousDay']<=15)] = 1
df['pdays_greater_15'][(df['PreviousDay']>15) & (df['PreviousDay'] < 999)] = 1
df_dropped_pdays = df.drop('PreviousDay', axis=1)
```

Based on the available clean data after applying data wrangling techniques, we will plot the bar graph for the features Marital Status, Job, Education, Has any House Loan, Any Personal Loan, Previous Contact Type (Cellular / Telephone), Last Contact Period Month, Last Contact Period Day of the week, Previous Outcome from conversation and see how many clients open the Term Deposit(Which Bank offers). We need to see what category of these features has more Yes and how it can be improved in a combination of these features to have more clients open Term Deposit.

```
categorical_variables = ['Marital', 'Job', 'Education', 'HasCredit', 'HouseLoan', 'PersonalLoan', 'ContactType', 'LastContMor
nrows=5
ncols=int(len(categorical_variables)/nrows)
fig,ax2d=plt.subplots(nrows,ncols, figsize=(20,30))

fig.subplots_adjust(wspace=0.5, hspace=0.3)
ax=np.ravel(ax2d)

for count,col in enumerate(categorical_variables):
    df3 = pd.crosstab(df[col], df['TermDeposit'])
    df3.plot(ax=ax[count], kind='barh', fontsize=12)

plt.show()
```

(Refer to fig 2 for subplots)

Subplots clearly tells us high percentage of clients from the previous campaign did not open the Term Deposit, so bank has to come up with better marketing campaign to target more clients for opening Term Deposit. Some insights from above subplot:

- Target more people with better strategy for clients whose marital status is married with better education (University Degree) and has more qualified job (Admin, Blue Collar).
- Clients having personal loan rather than house loan should be considered. It looks like not enough people with personal loan have been contacted.
- More existing clients who already have Term Deposit should be considered for future marketing and these clients may more likely to open another one .

```
f, ax = plt.subplots(1, 1, figsize = (40, 20))
pd.crosstab(df['Age'],df['TermDeposit']).plot(kind='bar', ax=ax)
plt.xticks(fontsize=24)
plt.yticks(fontsize=24)
plt.show()
```

Age group (26-40) has good target for client as per previous campaign but also tells more big percentage didn't opt for Term Deposit. It clearly indicates that older people(above 55) have been neglected, but the good amount of people who have been contacted did open the account. So bank should definitely consider contact more older people for future campaign.

Inferential Statistical Analysis:

After doing the initial exploratory data analysis , applying data wrangling techniques, we will apply some statistics concepts to further analyze data. To apply machine learning algorithms, we have to define the features and target variable. Before we apply these techniques we will check the correlation of our target variable which is whether the customer is willing to open a Term deposit or not and do some null hypothesis testing on variables as part of statistical analysis.

We will create dummy variables to have a good visual correlation graph.

df_with_dummies=pd.get_dummies(df_dropped_pdays)

We will drop one of the dummy variables are for n categories we will have only n-1 variables.

```
def dropfeatures(df,f):
    '''Drop one of the dummy variables'''
    df=df.drop(f, axis=1)
    return df
```

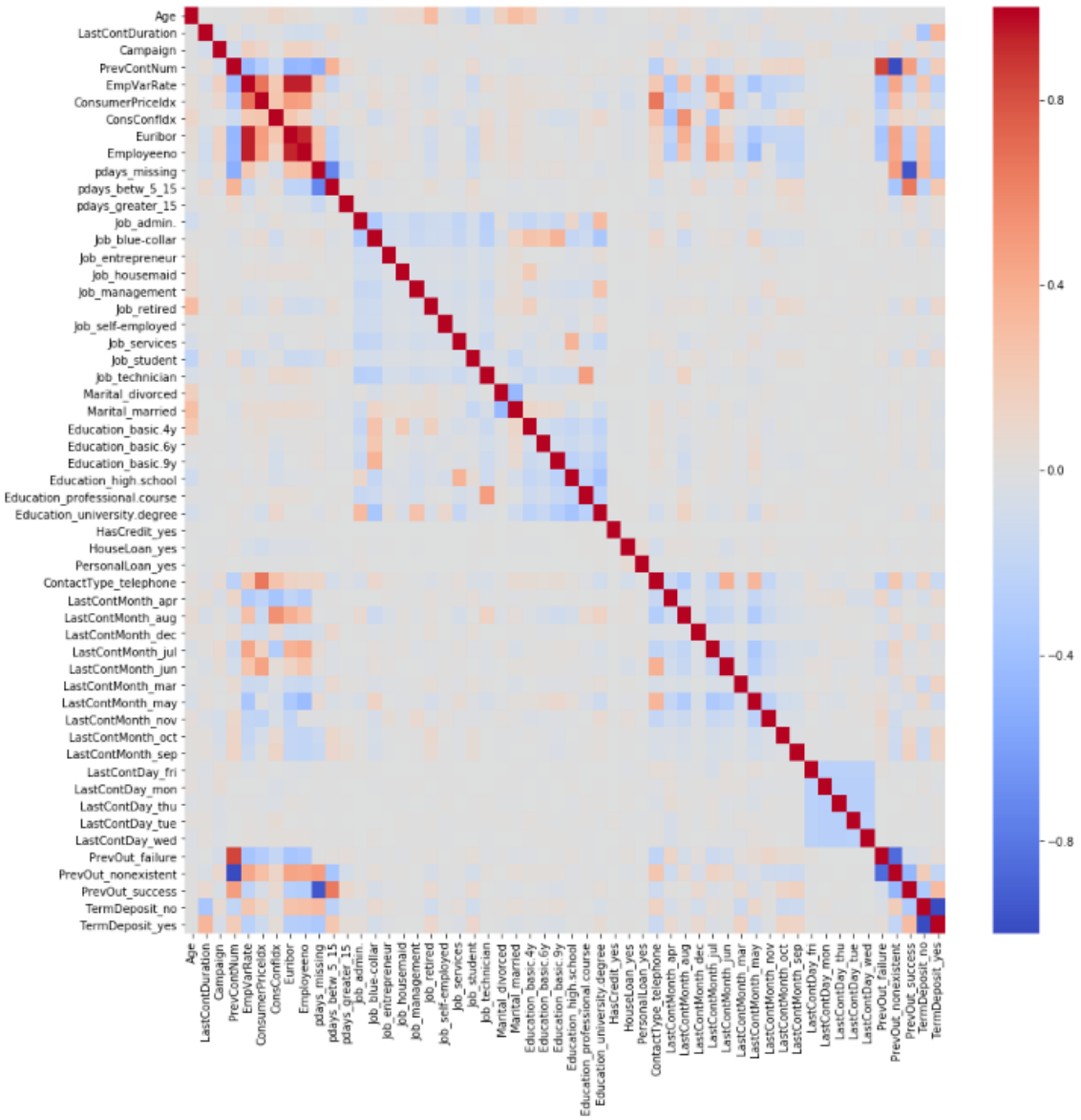
```
features_dropped = ['HasCredit_no','PersonalLoan_no','HouseLoan_no','Marital_single','ContactType_cellular','Education_illite
                    'Job_unemployed', 'pdays_less_5']
df_clean = dropfeatures(df_with_dummies, features_dropped)

f, ax = plt.subplots(figsize=(15, 15))
sns.heatmap(df_clean.corr(method='spearman'), annot=False, cmap='coolwarm')
plt.show()
```

Finally, a heatmap was created to show us whether there is strong correlation between the target variable and any independent variables. The heatmap is created using Spearman correlation, which measures the degree to which the rankings of each variable (as opposed to the actual values) align, thus minimizing the effect of outliers[2]. Once this is measured, those variables are expected to be significant during the modeling stage.

This graphic was created using Python's seaborn package and the specially written function drawheatmap, which takes a dataframe as an input.

Inferences from below heatmap: From the above heat map we can see that 'TermDeposit_yes' (our target variable) has good correlation with **'LastContDuration', 'EmpVarRate', 'Euribor', 'Employeeno', 'pdays_missing', 'pdays_betw_5_15', 'Prevout_nonexistent' and 'Prevout_success'**. We expect to see these independent variables as significant while building the models.

We conclude by doing hypothesis testing before applying any machine learning algorithms.



Null Hypothesis: Customer who opened the previous term deposit are likely have another.
Alternate Hypothesis: There is no correlation between previous outcome success and current term deposit.

```
df3=df
df3['TermDeposit'] = df['TermDeposit'].map({'yes': 1, 'no':0})
Term_Call = df3[df3.TermDeposit==1].TermDeposit
Prev_Call = df3[df3.PrevOut=='success'].TermDeposit
stats.ttest_ind(Term_Call, Prev_Call)
```

```
Ttest_indResult(statistic=49.852363111551121, pvalue=0.0)
```

Since the pvalue is 0.0 we can reject the Null hypothesis, there might not be the complete correlation on previous customers who already have term deposit likely to open another, there can be other factors to be considered for this target variable.

Fig 1:

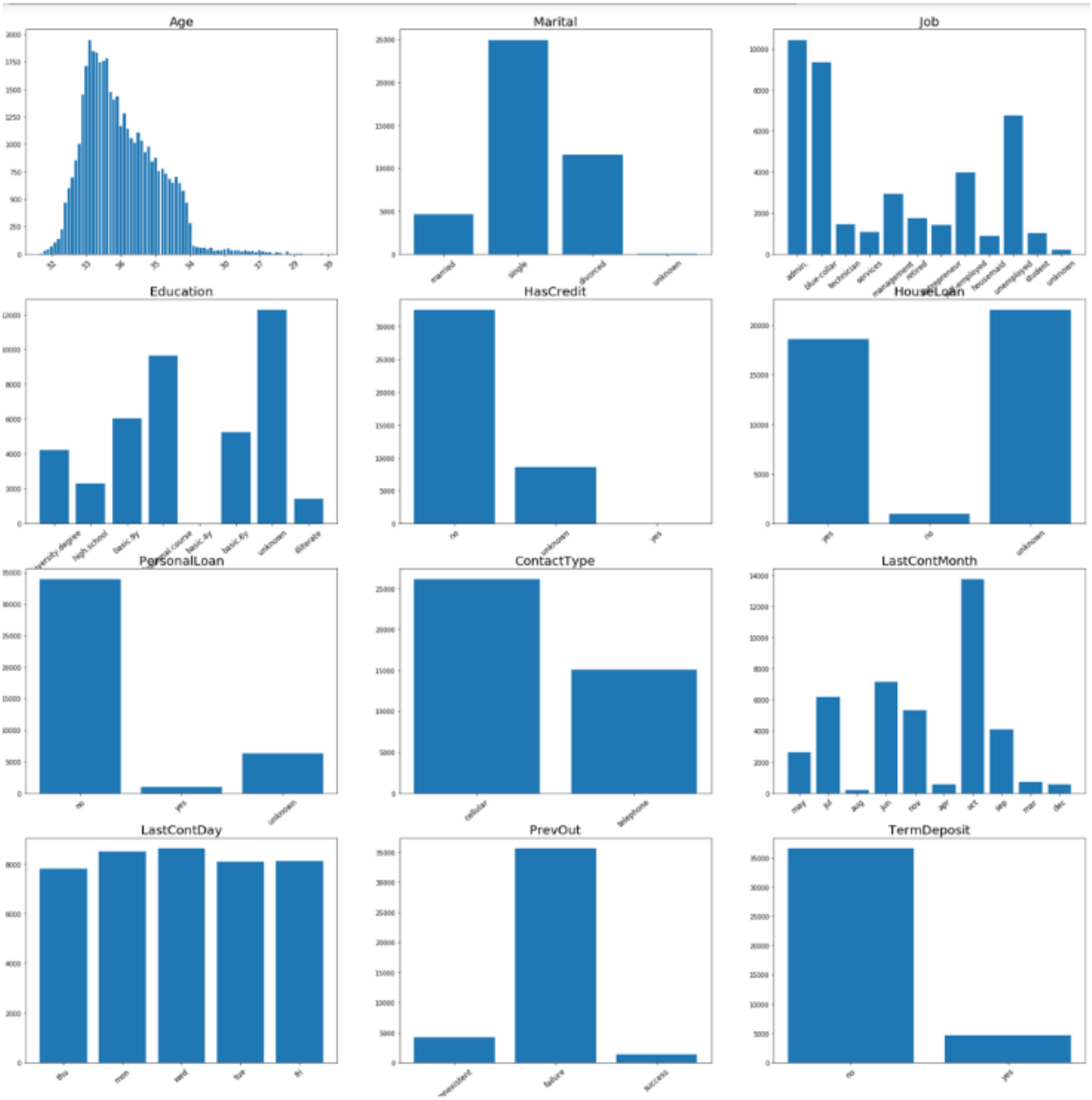
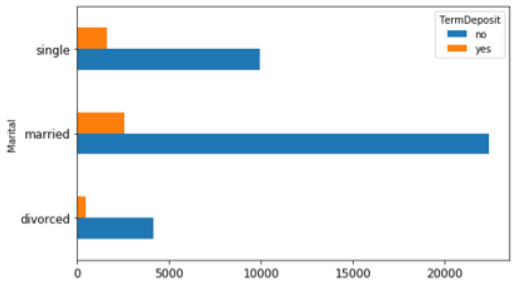
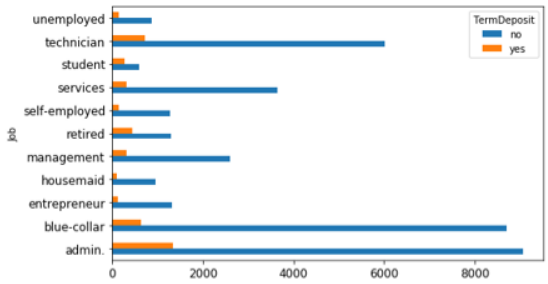
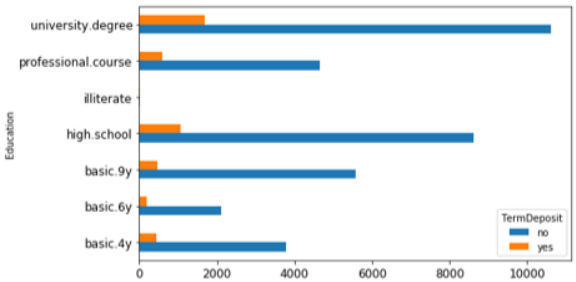
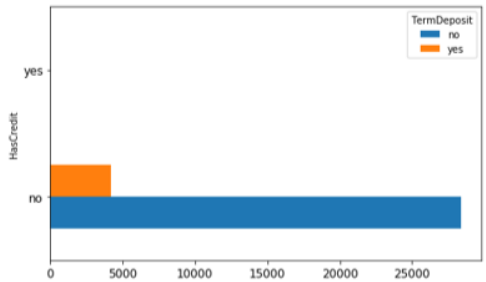

Age
Marital
Job
Education
HasCredit
HouseLoan
PersonalLoan
ContactType
LastContMonth
LastContDay
PrevOut
TermDeposit

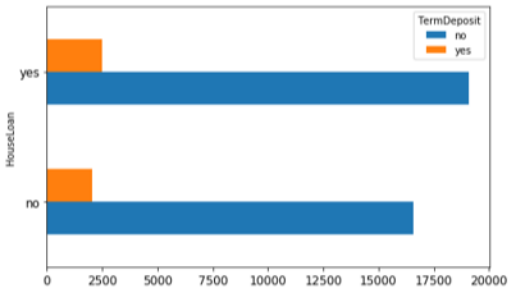
fig 2:

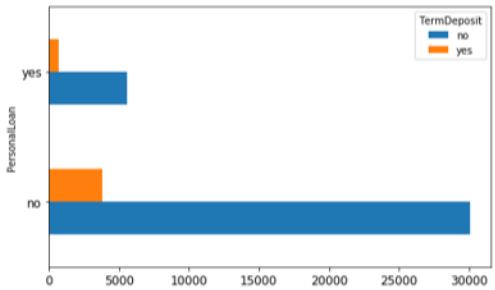

TermDeposit
no
yes
Marital
single
married
divorced
0
5000
10000
15000
20000

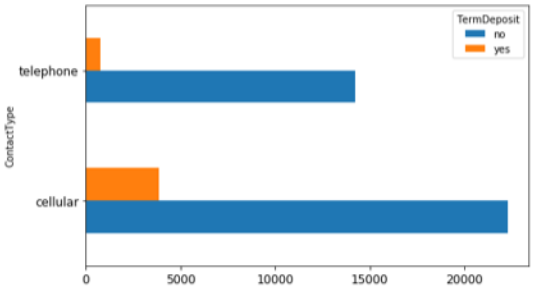

TermDeposit
no
yes
Job
unemployed
technician
student
services
self-employed
retired
management
housemaid
entrepreneur
blue-collar
admin.
0
2000
4000
6000
8000

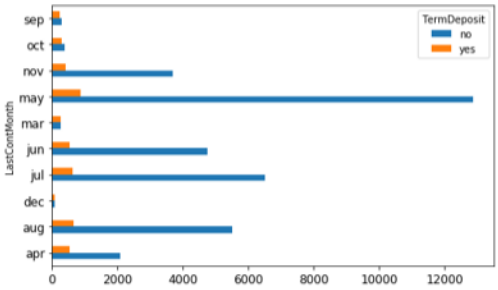

Education
university.degree
professional.course
illiterate
high.school
basic.9y
basic.6y
basic.4y
TermDeposit
no
yes
0
2000
4000
6000
8000
10000

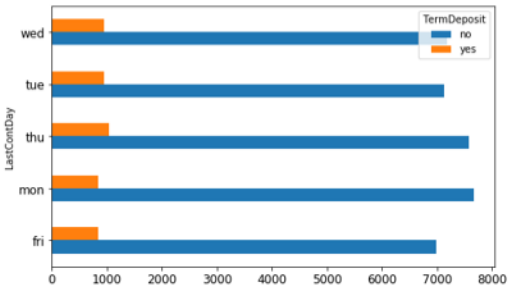

TermDeposit
no
yes
HasCredit
yes
no
0
5000
10000
15000
20000
25000

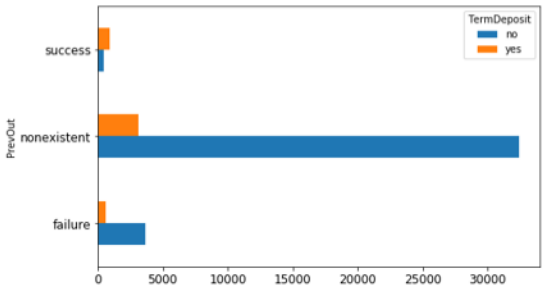

TermDeposit
no
yes
HouseLoan
yes
no
0
2500
5000
7500
10000
12500
15000
17500
20000


TermDeposit
no
yes
PersonalLoan
yes
no
0
5000
10000
15000
20000
25000
30000


TermDeposit
no
yes
ContactType
telephone
cellular
0
5000
10000
15000
20000


TermDeposit
no
yes
LastContMonth
sep
oct
nov
may
mar
jun
jul
dec
aug
apr
0
2000
4000
6000
8000
10000
12000


TermDeposit
no
yes
LastContDay
wed
tue
thu
mon
fri
0
1000
2000
3000
4000
5000
6000
7000
8000


TermDeposit
no
yes
PrevOut
success
nonexistent
failure
0
5000
10000
15000
20000
25000
30000