CEBD 1260 – Big Data Analytics

Black Friday

A study of sales trough consumer behaviours

Team Project

Ricardo Luis da Costa Rocha

Frank So

Black Friday Study

Sections

Problem definition

Dataset description

Approach

Results

Discussion

Black Friday Study

Problem statement

“A retail company “ABC Private Limited” wants to understand the customer purchase behaviour (specifically, purchase amount) against various products of different categories. They have shared purchase summary of various customers for selected high volume products from last month.

The data set also contains customer demographics (age, gender, marital status, city_type, stay_in_current_city), product details (product_id and product category) and Total purchase_amount from last month.

Now, they want to build a model to predict the purchase amount of customer against various products which will help them to create personalized offer for customers against different products.”

https://datahack.analyticsvidhya.com/contest/black-friday/

Black Friday Study

Problem definition

“The dataset here is a sample of the transactions made in a retail store. The store wants to know better the customer purchase behaviour against different products. Specifically, here the problem is a regression problem where we are trying to predict the dependent variable (the amount of purchase) with the help of the information contained in the other variables.

Classification problem can also be settled in this dataset since several variables are categorical, and some other approaches could be "Predicting the age of the consumer" or even "Predict the category of goods bought". This dataset is also particularly convenient for clustering and maybe find different clusters of consumers within it.”

https://www.kaggle.com/mehdidag/black-friday

Black Friday Study

Dataset description

“Dataset of 550 000 observations about the black Friday in a retail store, it contains different kinds of variables either numerical or categorical. It contains missing values”

Activities LibreOffice Calc Sat 04:22 en

BlackFriday_sample.csv - LibreOffice Calc

User_ID	Product_ID	Gender	Age	Occupation	City_Category	Stay_In_Current_City_Years	Marital_Status	Product_Category_1	Product_Category_2	Product_Category_3	Purchase
1000001	P00069042	F	0-17	10	A	2	0	3			8370
1000001	P00248942	F	0-17	10	A	2	0	1	6	14	15200
1000001	P00087842	F	0-17	10	A	2	0	12			1422
1000001	P00085442	F	0-17	10	A	2	0	12	14		1057
1000002	P00285442	M	55+	16	C	4+	0	8			7969
1000003	P00193542	M	26-35	15	A	3	0	1	2		15227
1000004	P00184942	M	46-50	7	B	2	1	1	8	17	19215
1000004	P00346142	M	46-50	7	B	2	1	1	15		15854
1000004	P0097242	M	46-50	7	B	2	1	1	16		15686

Analytics Vidhya
Learn everything about analytics

Blog Learn Engage Compete Get Hired User Rankings All Hackathons

Data

Variable	Definition
User_ID	User ID
Product_ID	Product ID
Gender	Sex of User
Age	Age in bins
Occupation	Occupation (Masked)
City_Category	Category of the City (A,B,C)
Stay_In_Current_City_Years	Number of years stay in current city
Marital_Status	Marital Status
Product_Category_1	Product Category (Masked)
Product_Category_2	Product may belongs to other category also (Masked)
Product_Category_3	Product may belongs to other category also (Masked)
Purchase	Purchase Amount (Target Variable)

Your model performance will be evaluated on the basis of your prediction of the purchase amount for the test data (test.csv), which contains similar data-points as train except for their purchase amount. Your submission needs to be in the format as shown in "SampleSubmission.csv".

We at our end, have the actual purchase amount for the test dataset, against which your predictions will be evaluated. Submissions are scored on the root mean squared error (RMSE). RMSE is very common and is a suitable general-purpose error metric. Compared to the Mean Absolute Error, RMSE punishes large errors:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2},$$

Feedback

Black Friday Study

Approach

Data Wrangling & Cleaning

jupyter data_wrangling_cleaning_final (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Code

Removing not relevant columns

All the columns appears to be relevant. Even User_ID and Product_ID can be used depending on the type
be involving Product_Category_2 & Product_Category_3 in our calculations due to it obscurity. Nonetheless
said categories as complimentary products due to its association with Prodcut_Category_1 and product ID.

Removing duplicates

In [7]:

```python
# There are no duplicates to be removed
duplicateRowsDF = df[df.duplicated()]
print("Duplicate Rows except first occurrence based on all columns are :")
print(duplicateRowsDF)
```

```
Duplicate Rows except first occurrence based on all columns are :
Empty DataFrame
Columns: [User_ID, Product_ID, Gender, Age, Occupation, City_Category, Stay_In_Curre
ct_Category_1, Product_Category_2, Product_Category_3, Purchase]
Index: []
```

Dealing with missing values ¶

In [8]:

```python
# Checking for missing values by columns
print(df.isnull().sum())
```

```
User_ID                           0
Product_ID                        0
Gender                            0
Age                               0
Occupation                        0
City_Category                     0
Stay_In_Current_City_Years        0
Marital_Status                    0
Product_Category_1                0
Product_Category_2           166986
Product_Category_3           373299
Purchase                          0
dtype: int64
```

In [9]:

```python
## Assigning value zero for the NaN cases
df.fillna(value=0,inplace=True)
```

Black Friday Study

Approach

Exploratory Data Analysis

jupyter exploratory_data_analysis_final (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Run Markdown

Data Perspective

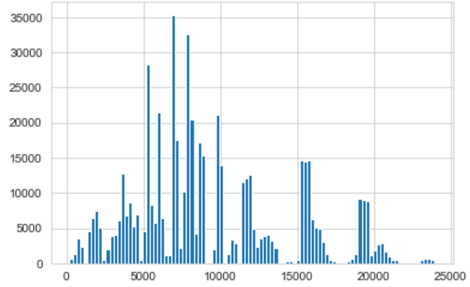
One Variable (numeric)

In [32]:

```
# Analize purchase distribution

# Histogram
print(df['Purchase'].describe().round())
plt.hist(df['Purchase'], bins=100)
plt.show()
# Box plot
plt.boxplot(df['Purchase'])
plt.xticks([1], ['Purchase'], rotation='horizontal')
plt.show()
```

```
count    537577.00
mean       9334.00
std        4981.00
min         185.00
25%        5866.00
50%        8062.00
75%       12073.00
max       23961.00
Name: Purchase, dtype: float64
```



Black Friday Study - Approach EDA

jupyter exploratory_data_analysis_final (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

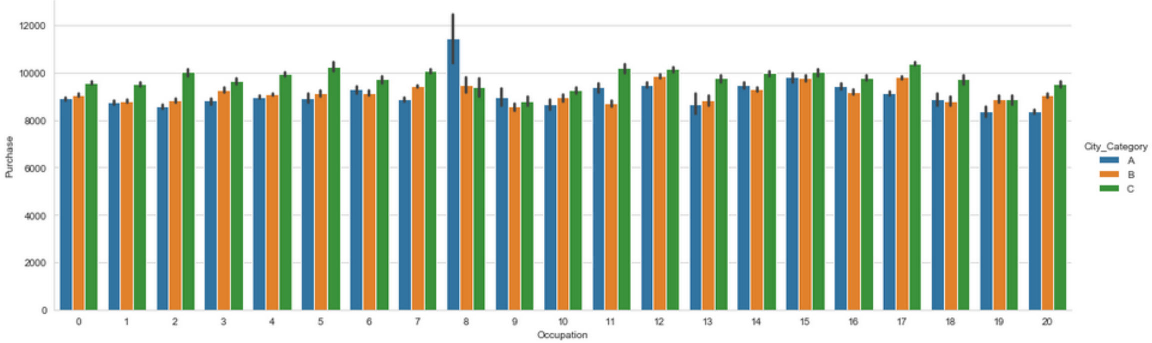
Business Perspective

"An approximate answer to the right question is worth a great deal more than a precise answer to the wrong question." John Tukey

In [43]:

```
# Cat plot to show the distribution between Occupation x Purchase
sns.catplot(x = "Occupation",
            y = "Purchase",
            hue = "City_Category",
            hue_order = ["A", "B", "C"],
            aspect = 3,
            kind = "bar",
            data = df)
```

Out[43]: <seaborn.axisgrid.FacetGrid at 0x7f4032344668>



Black Friday Study - Approach ML Model

jupyter machine_learning_model Last Checkpoint: 2019-05-28 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

Run | Code

Feature Engineering

In [4]:

```
# TODO: create a loop to transform the categorical columns to numerical
for col in ['Gender', 'Age', 'Occupation', 'City_Category', 'Stay_In_Current_City_Years', 'Marital_Status', 'Product_Cate
    df_dummies = pd.get_dummies(df[col], prefix=col)
    df = pd.concat([df, df_dummies], axis=1)
    # Remove the original columns
    del df[col]
df.head()
```

Out[4]:

	User_ID	Product_ID	Purchase	Gender_F	Gender_M	Age_0-17	Age_18-25	Age_26-35	Age_36-45	Age_46-50	...	Product_Category_3_9	Product_Cate
0	1000001	P00069042	8370	1	0	1	0	0	0	0	...	0	
1	1000001	P00248942	15200	1	0	1	0	0	0	0	...	0	
2	1000001	P00087842	1422	1	0	1	0	0	0	0	...	0	
3	1000001	P00085442	1057	1	0	1	0	0	0	0	...	0	
4	1000002	P00285442	7969	0	1	0	0	0	0	0	...	0	

5 rows × 95 columns

In [5]:

```
df.dtypes
```

Out[5]:

```
User_ID                     int64
Product_ID                 object
Purchase                    int64
Gender_F                    uint8
Gender_M                    uint8
Age_0-17                    uint8
Age_18-25                   uint8
```

Black Friday Study - Approach ML Model

jupyter machine_learning_model Last Checkpoint: 2019-05-28 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Model Training

In [7]:

```
threshold = 0.8
X = df[X_columns]
y = df[y_column]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=1.0-threshold, shuffle=True)

print('X_train', X_train.shape)
print('y_train', y_train.shape)
print('X_test', X_test.shape)
print('y_test', y_test.shape)
```

```
X_train (430061, 93)
y_train (430061, 1)
X_test (107516, 93)
y_test (107516, 1)
```

Model Evaluation

Linear Regression

In [8]:

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

In [9]:

```
mae = mean_absolute_error(y_test, y_pred)
print('MAE', round(mae, 2))
```

Black Friday Study – Results Experiments

jupyter machine_learning_model_ricardo Last Checkpoint: an hour ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Experiments

In [9]:

```
def model_training(model_name, model, X_train, y_train):
    model.fit(X_train, y_train)
    return model

def model_prediction(model, X_test):
    y_pred = model.predict(X_test)
    return y_pred

def model_evaluation(model_name, y_test, y_pred):
    print(model_name)
    print('MAE', mean_absolute_error(y_test, y_pred))
    print('RMSE', np.sqrt(mean_squared_error(y_test, y_pred)))
    plt.scatter(y_test, y_pred, alpha=0.3)
#     plt.plot(range(0,5000000, 100), range(0,5000000, 100), '--r', alpha=0.3, label='Line1')
    plt.plot(range(0,10), range(0,10), '--r', alpha=0.3, label='Line1')
    plt.title(model_name)
    plt.xlabel('True Value')
    plt.ylabel('Predict Value')
#     plt.xlim([0, 5000000])
#     plt.ylim([0, 5000000])
    plt.show()
    print('')

def run_experiment(model_name, model, X_train, y_train, X_test):
    train_model = model_training(model_name, model, X_train, y_train)
    predictions = model_prediction(train_model, X_test)
    model_evaluation(model_name, y_test, predictions)

run_experiment('Linear Regression', LinearRegression(), X_train, y_train, X_test)
run_experiment('KNN 5', KNeighborsRegressor(5), X_train, y_train, X_test)
run_experiment('KNN 2', KNeighborsRegressor(2), X_train, y_train, X_test)
run_experiment('Decision Tree', DecisionTreeRegressor(), X_train, y_train, X_test)
run_experiment('Random Forest 10', RandomForestRegressor(10), X_train, y_train, X_test)
```

Black Friday Study - Results Error Analysis

jupyter machine_learning_model_ricardo Last Checkpoint: 2 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

Error Analysis

In [11]:
```
model = RandomForestRegressor(100)
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
```

```
/home/coastrock/anaconda3/lib/python3.7/site-packages/ipykernel_launcher.py:2: DataConversionWarning: A column-vector y
was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
```

In [12]:
```
#for i in range(len(X_test.columns)):
fi = []
for i, col in enumerate(X_test.columns):
    fi.append([col, model.feature_importances_[i]])
pd.DataFrame(fi).sort_values(1, ascending=False)
```

4	Gender_M	0.008624
34	City_Category_B	0.008042
33	City_Category_A	0.007579
12	Occupation_0	0.007434
19	Occupation_7	0.006780
13	Occupation_1	0.006691
16	Occupation_4	0.006637
29	Occupation_17	0.006328
35	City_Category_C	0.006142
9	Age_46-50	0.006123
32	Occupation_20	0.006023

Black Friday Study - Results Clustering Kmeans

jupyter clustering-kmeans Last Checkpoint: 12 hours ago (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

Run Code

Model Training

In [4]:

```
k = 7
kmeans = KMeans(n_clusters=k).fit(df_norm.values)

print(set(kmeans.labels_))
print(collections.Counter(kmeans.labels_))

df_results = df.copy()
df_norm['cluster'] = kmeans.labels_
df_results['cluster'] = kmeans.labels_
```

```
{0, 1, 2, 3, 4, 5, 6}
Counter({2: 47619, 5: 17664, 0: 16651, 4: 12995, 1: 2426, 3: 1490, 6: 1154})
```

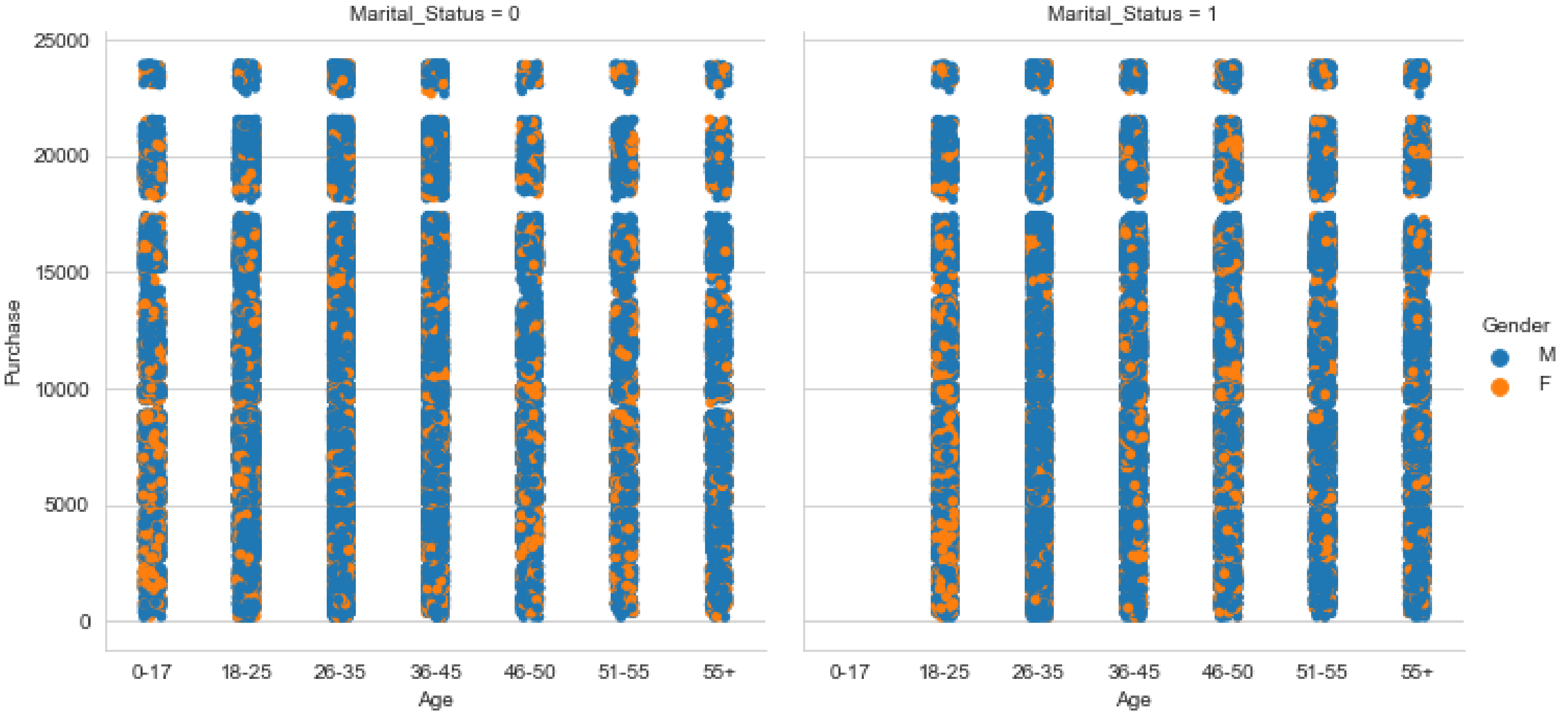
In [5]:

```
# Analyze the results
df_results = df_results.reset_index()
for cluster in sorted(set(kmeans.labels_)):
    print(collections.Counter(df_results[df_results['cluster']==cluster]['Age']).most_common(5))

n_clusters = len(set(kmeans.labels_))
for col in X_columns:
    print(col)
    i = 1
    plt.figure(figsize=(16,3))
    for cluster in sorted(set(kmeans.labels_)):
        plt.subplot(1, n_clusters, i)
        plt.xlim([0,df_results[col].max()])
        plt.hist(df_results[df_results['cluster']==cluster][col], label=str(cluster), alpha=0.3, bins=20)
        i += 1
    plt.show()
```

```
[('26-35', 7071), ('36-45', 3992), ('18-25', 2966), ('46-50', 1034), ('51-55', 976)]
[('0-17', 2086), ('18-25', 307), ('55+', 20), ('36-45', 10), ('26-35', 3)]
[('26-35', 20239), ('36-45', 9019), ('18-25', 8713), ('46-50', 4251), ('51-55', 3795)]
[('55+', 956), ('51-55', 357), ('46-50', 114), ('36-45', 60), ('0-17', 3)]
```

Black Friday Study – Discussion



Black Friday Study

Thank you!

Questions?