In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import pylab as pl
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn import linear_model
import warnings
import dateutil
from datetime import datetime
import calendar
from pandas.api.types import CategoricalDtype
```

In [2]:

```
data_Consumers = pd.read_csv(r'C:\Study\Data Science\Learn Data Science with Python\Task\Co
```

In [3]:

```
data_Pos = pd.read_csv(r'C:\Study\Data Science\Learn Data Science with Python\Task\Pos.csv'
```

In [4]:

```
data_Products = pd.read_csv(r'C:\Study\Data Science\Learn Data Science with Python\Task\Pro
```

In [5]:

```
data=pd.merge(data_Pos,data_Products, on = 'pid' )
```

In [6]:

```
data.head()
```

Out[6]:

	pid	cid	rid	date	time	price	discount	price_addedvat	marginal	qu
0	4401628	107430927	7	18-01-01	09:01:19	10.437785	0.0	11.690267	2.784990	
1	4401628	136502829	46	18-01-01	10:02:00	9.567943	0.0	10.716078	2.867688	
2	4401628	159696310	226	18-01-02	09:02:07	10.147801	0.0	11.365538	2.495006	
3	4401628	130016198	162	18-01-02	09:03:02	40.591206	0.0	45.462151	9.980025	
4	4401628	122455719	145	18-01-02	09:05:24	9.567943	0.0	10.716078	2.867688	

In [7]:

```
datad=pd.merge(data,data_Consumers, on = 'cid' )
```

In [8]:

```
datad.head()
```

Out[8]:

	pid	cid	rid	date	time	price	discount	price_addedvat	marginal
0	4401628	107430927	7	18-01-01	09:01:19	10.437785	0.0	11.690267	2.784990
1	4401628	107430927	6783	18-02-15	16:04:23	10.437785	0.0	11.690267	2.784990
2	4401628	107430927	12046	18-03-22	10:30:16	10.437785	0.0	11.690267	2.784990
3	4423732	107430927	7	18-01-01	09:01:19	12.467345	0.0	13.963375	6.059347
4	4423732	107430927	24384	18-06-07	11:09:03	12.467345	0.0	13.963375	6.048523

In [9]:

```
datad['purchase']=datad['discount']+datad['price_addedvat']
```

In [10]:

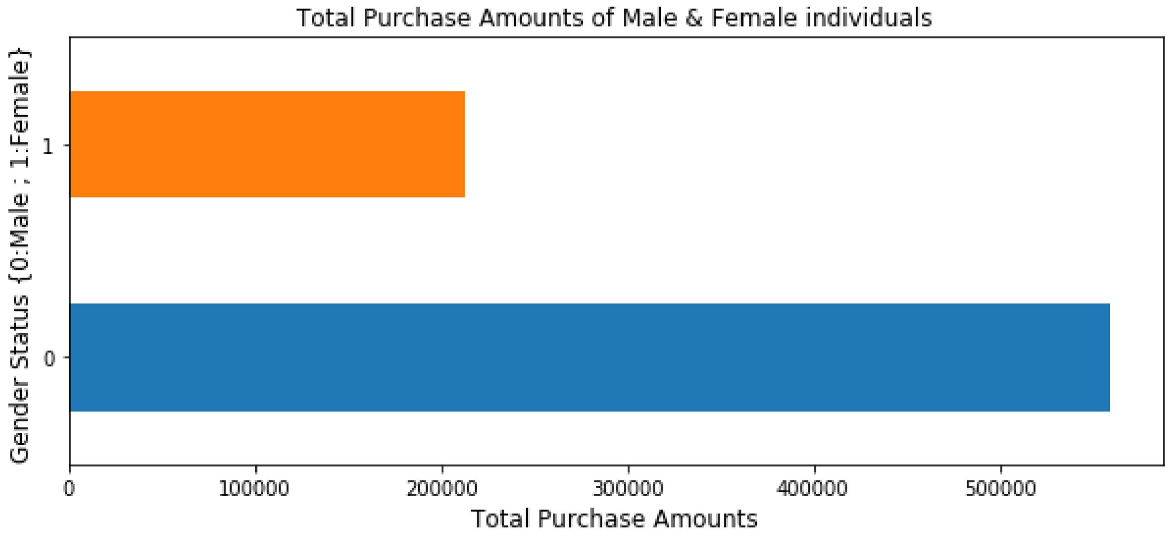
```
datad.head()
```

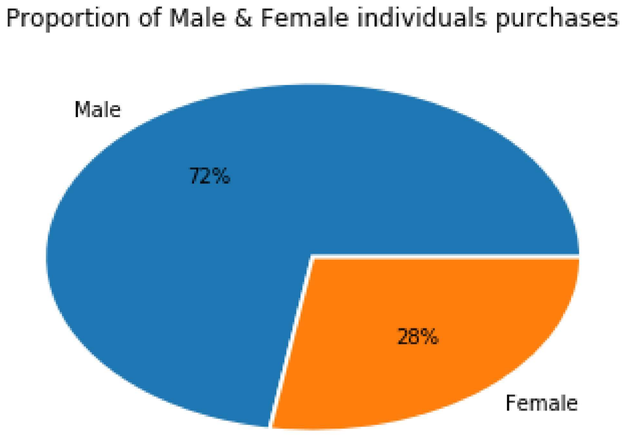
Out[10]:

	pid	cid	rid	date	time	price	discount	price_addedvat	marginal
0	4401628	107430927	7	18-01-01	09:01:19	10.437785	0.0	11.690267	2.784990
1	4401628	107430927	6783	18-02-15	16:04:23	10.437785	0.0	11.690267	2.784990
2	4401628	107430927	12046	18-03-22	10:30:16	10.437785	0.0	11.690267	2.784990
3	4423732	107430927	7	18-01-01	09:01:19	12.467345	0.0	13.963375	6.059347
4	4423732	107430927	24384	18-06-07	11:09:03	12.467345	0.0	13.963375	6.048523

In [11]:

```
#Comparing total purchase amounts of Male & Female individuals   #   Not working
pl.figure(figsize =(10,4))
#dataNew.groupby('gender').purchase.sum().plot('barh')
datad.groupby('gender').purchase.count().plot('barh')

#dataNew[dataNew['price_addedvat']+ data[' discount ']].groupby([' category ']).purchase.co
pl.ylabel('Gender Status {0:Male ; 1:Female}', fontsize=12)
pl.xlabel('Total Purchase Amounts', fontsize=12)
pl.title('Total Purchase Amounts of Male & Female individuals', fontsize=12)
plt.show()

plt.pie(datad["gender"].value_counts().values, labels=["Male","Female"], autopct="%1.0f%%",
plt.title("Proportion of Male & Female individuals purchases")
plt.show()
```





In [12]:

```
datag=datad
```

In [13]:

```
datag.describe()
```

Out[13]:

	pid	cid	rid	price	discount	price_addedv
count	7.721820e+05	7.721820e+05	772182.000000	772182.000000	772182.000000	772182.0000
mean	6.346639e+06	1.328194e+08	28802.188901	24.065330	-1.496217	28.6591
std	3.555690e+06	1.816107e+07	16697.548298	32.700340	8.257963	40.4456
min	8.000000e+00	1.014566e+08	1.000000	-710.501434	-1082.432160	-795.7608
25%	4.349421e+06	1.182611e+08	14320.000000	11.549226	0.000000	13.9633
50%	4.414560e+06	1.315419e+08	28731.000000	18.266043	0.000000	21.5404
75%	9.890323e+06	1.478355e+08	43259.000000	26.964143	0.000000	31.9317
max	1.443733e+07	1.692695e+08	57563.000000	5412.160800	0.000000	5412.1608

In [14]:

```
datag['date']=datag['date'].apply(dateutil.parser.parse,yearfirst=True)
```

In [15]:

```
datag.head()
```

Out[15]:

	pid	cid	rid	date	time	price	discount	price_addedvat	marginal
0	4401628	107430927	7	2018-01-01	09:01:19	10.437785	0.0	11.690267	2.784990
1	4401628	107430927	6783	2018-02-15	16:04:23	10.437785	0.0	11.690267	2.784990
2	4401628	107430927	12046	2018-03-22	10:30:16	10.437785	0.0	11.690267	2.784990
3	4423732	107430927	7	2018-01-01	09:01:19	12.467345	0.0	13.963375	6.059347
4	4423732	107430927	24384	2018-06-07	11:09:03	12.467345	0.0	13.963375	6.048523

In [16]:

```
datad=datag
```

In [17]:

```
type(datad['date'][0])
```

Out[17]:

```
pandas._libs.tslibs.timestamps.Timestamp
```

In [18]:

```
datad['month']=[d.month for d in datad['date']]
```

In [19]:

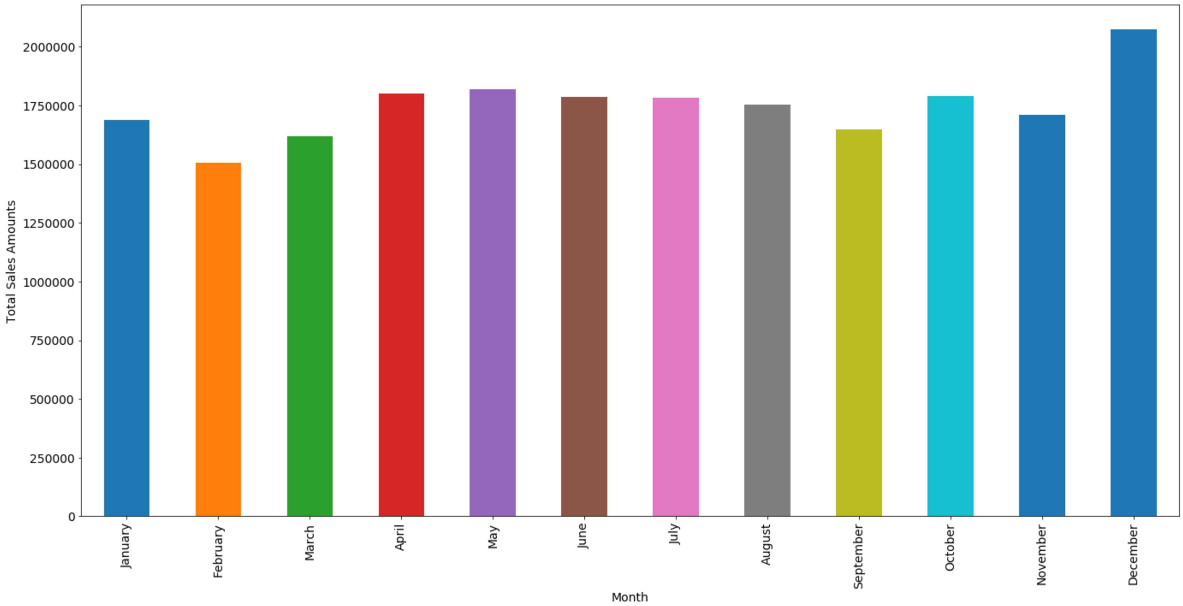
```
datad.head()
```

Out[19]:

	pid	cid	rid	date	time	price	discount	price_addedvat	marginal
0	4401628	107430927	7	2018-01-01	09:01:19	10.437785	0.0	11.690267	2.784990
1	4401628	107430927	6783	2018-02-15	16:04:23	10.437785	0.0	11.690267	2.784990
2	4401628	107430927	12046	2018-03-22	10:30:16	10.437785	0.0	11.690267	2.784990
3	4423732	107430927	7	2018-01-01	09:01:19	12.467345	0.0	13.963375	6.059347
4	4423732	107430927	24384	2018-06-07	11:09:03	12.467345	0.0	13.963375	6.048523

In [20]:

```
#Comparing total purchase amounts of different age individuals  # Need correction
pl.figure(figsize =(23,11))
datad.groupby('month').purchase.sum().plot('bar')
#pl.ylabel('Age', fontsize=12)
#pl.xlabel('Total Purchase Amounts', fontsize=12)
pl.xlabel('Month', fontsize=14)
pl.ylabel('Total Sales Amounts', fontsize=14)
plt.xticks(datad['month'].unique()-1,[calendar.month_name[i] for i in datad['month'].unique
plt.yticks(fontsize=14)
#plt.xticks('Jan','Feb','')
#pl.title('Total Purchase Amounts of different month individuals', fontsize=14)
plt.show()
```



In [21]:

```
[calendar.month_name[i] for i in datad['month'].unique()]
```

Out[21]:

```
['January',
 'February',
 'March',
 'June',
 'August',
 'September',
 'April',
 'May',
 'July',
 'October',
 'November',
 'December']
```

In [22]:

```
datad.groupby('month').purchase.sum()
```

Out[22]:

```
month
1     1.686786e+06
2     1.504427e+06
3     1.619381e+06
4     1.799255e+06
5     1.820790e+06
6     1.784829e+06
7     1.783200e+06
8     1.753694e+06
9     1.648940e+06
10    1.789489e+06
11    1.709657e+06
12    2.074289e+06
Name: purchase, dtype: float64
```

In [23]:

```
datad['week']=[calendar.day_name[d.weekday()] for d in datad['date']]
```

In [24]:

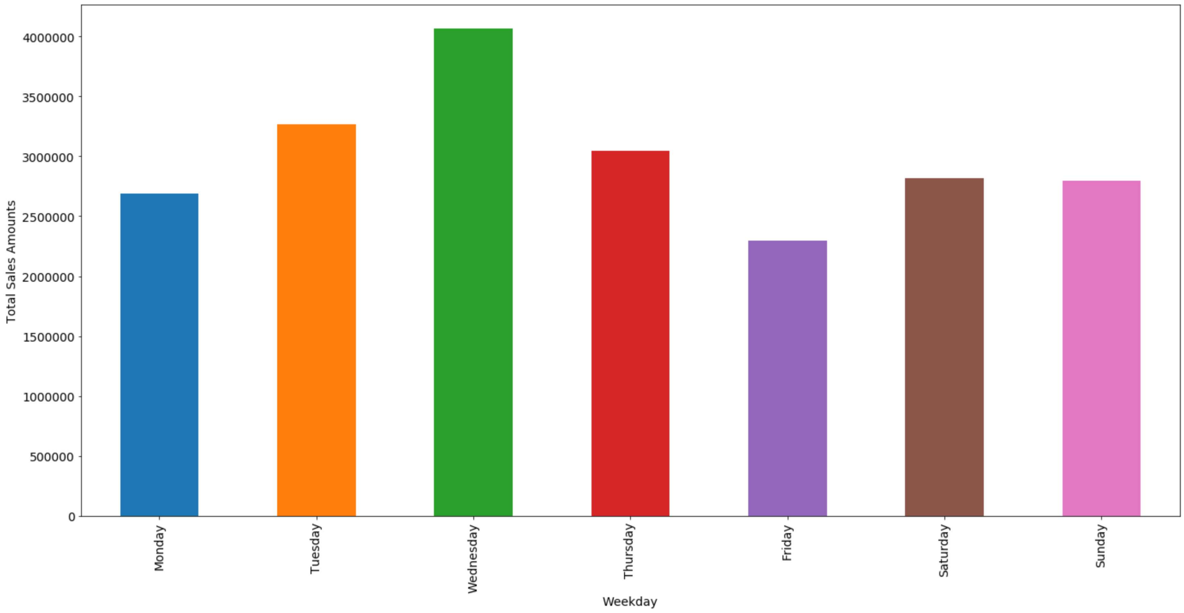
```
datad.head()
```

Out[24]:

	pid	cid	rid	date	time	price	discount	price_addedvat	marginal
0	4401628	107430927	7	2018-01-01	09:01:19	10.437785	0.0	11.690267	2.784990
1	4401628	107430927	6783	2018-02-15	16:04:23	10.437785	0.0	11.690267	2.784990
2	4401628	107430927	12046	2018-03-22	10:30:16	10.437785	0.0	11.690267	2.784990
3	4423732	107430927	7	2018-01-01	09:01:19	12.467345	0.0	13.963375	6.059347
4	4423732	107430927	24384	2018-06-07	11:09:03	12.467345	0.0	13.963375	6.048523

In [25]:

```
#Comparing total purchase amounts of different age individuals  # Need correction
cats = [ 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
pl.figure(figsize =(23,11))
datad.groupby('week').purchase.sum().reindex(cats).plot('bar')
#pl.ylabel('Age', fontsize=12)
#pl.xlabel('Total Purchase Amounts', fontsize=12)
pl.xlabel('Weekday', fontsize=14)
pl.ylabel('Total Sales Amounts', fontsize=14)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)

#plt.xticks('Jan','Feb','')
#pl.title('Total Purchase Amounts of different week individuals', fontsize=14)
plt.show()
```

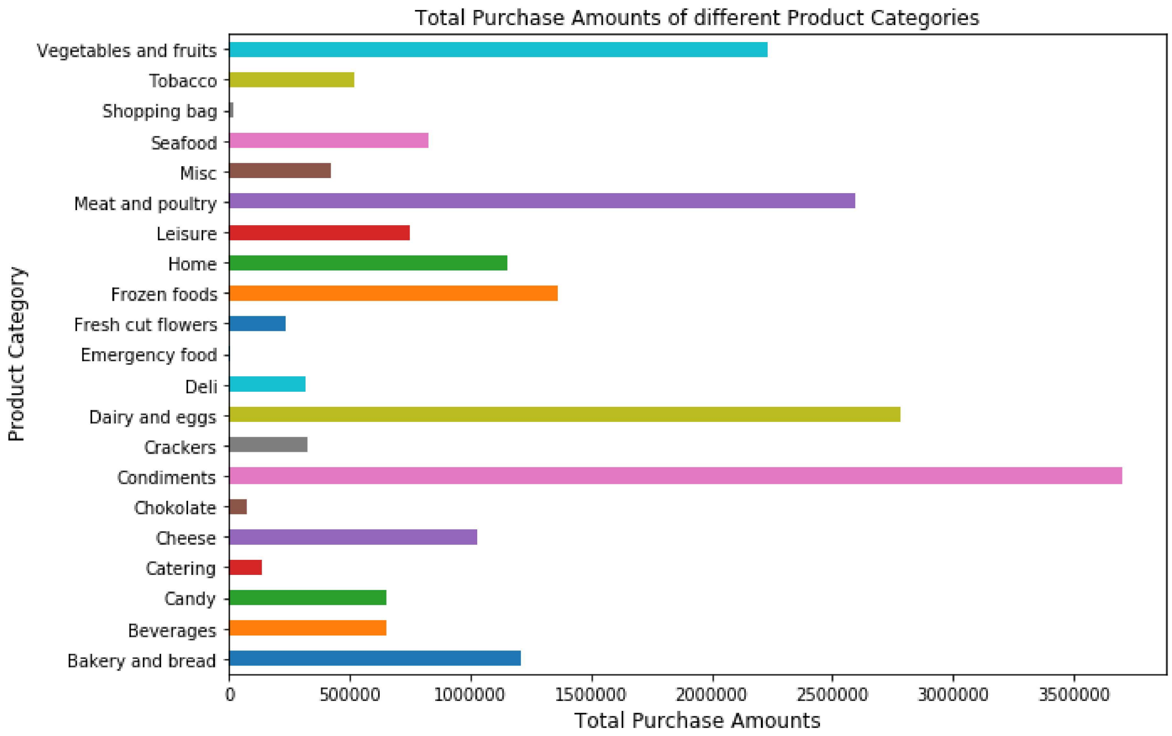


In [26]:

```
dataT=datad
```

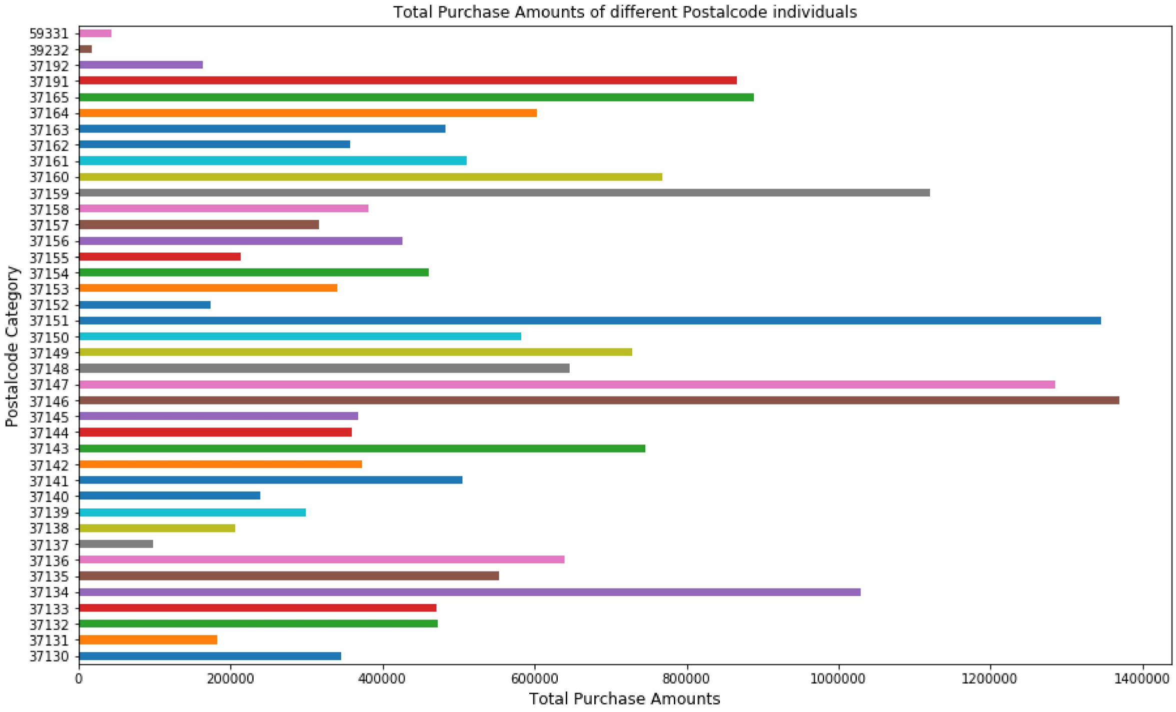
In [27]:

```
#Comparing total purchase amounts  Product catagory
pl.figure(figsize =(10,7))
dataT.groupby('category').purchase.sum().plot('barh')
pl.ylabel(' Product Category', fontsize=12)
pl.xlabel('Total Purchase Amounts', fontsize=12)
pl.title('Total Purchase Amounts of different Product Categories', fontsize=12)
plt.show()
```



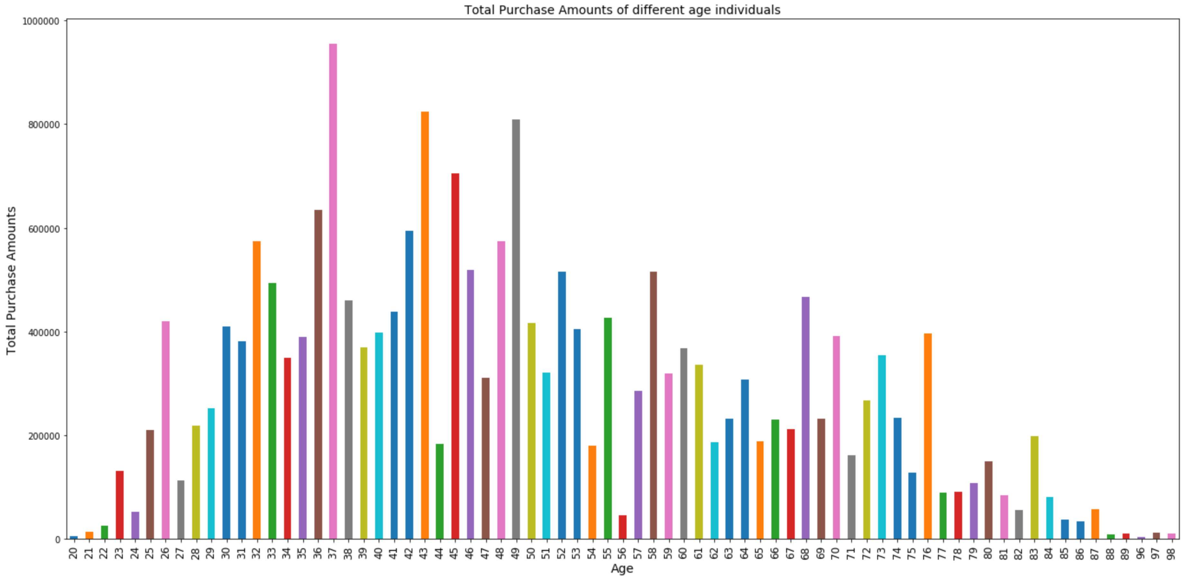
In [28]:

```
#Comparing total purchase amounts postalcode
pl.figure(figsize =(15,9))
dataT.groupby('postalcode').purchase.sum().plot('barh')
pl.ylabel('Postalcode Category', fontsize=12)
pl.xlabel('Total Purchase Amounts', fontsize=12)
pl.title('Total Purchase Amounts of different Postalcode individuals', fontsize=12)
plt.show()
```



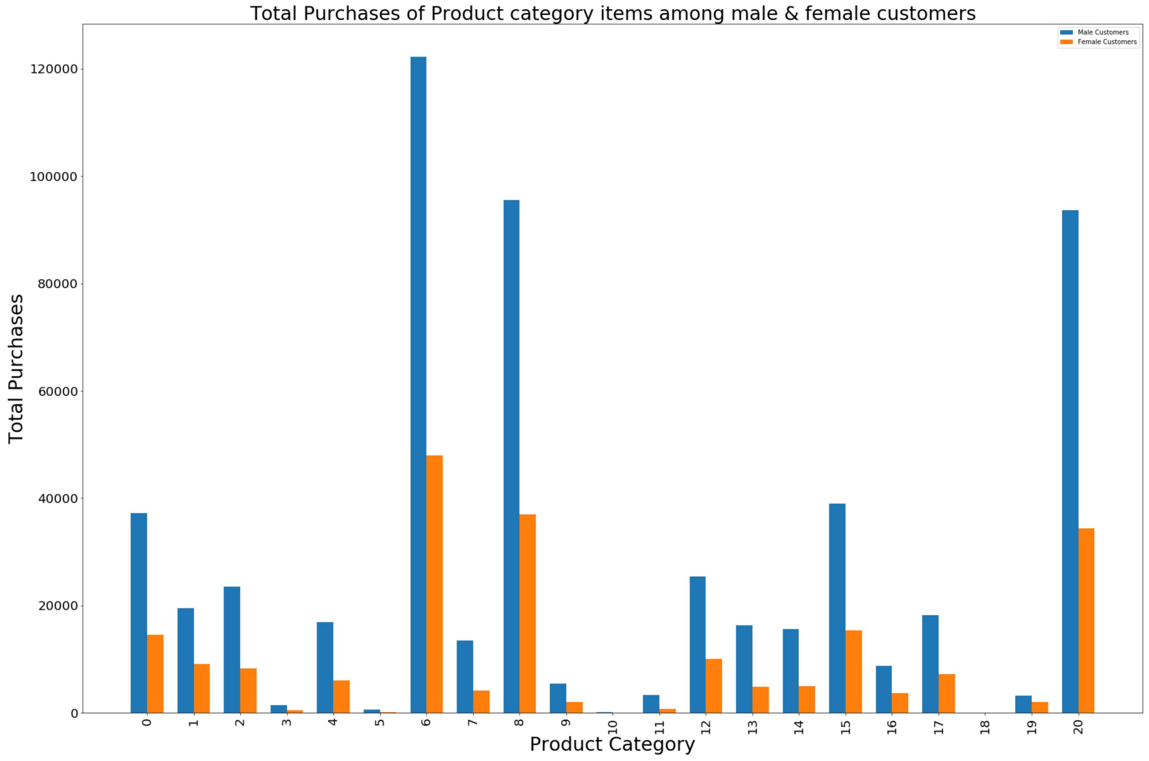
In [29]:

```
#Comparing total purchase amounts of different age individuals  # Need correction
pl.figure(figsize =(23,11))
dataT.groupby('age').purchase.sum().plot('bar')
#pl.ylabel('Age', fontsize=12)
#pl.xlabel('Total Purchase Amounts', fontsize=12)
pl.xlabel('Age', fontsize=14)
pl.ylabel('Total Purchase Amounts', fontsize=14)
plt.xticks(fontsize=12)

pl.title('Total Purchase Amounts of different age individuals', fontsize=14)
plt.show()
```



In [30]:

```
data=dataT#Data Transformation
encode = LabelEncoder()
#encode.fit(['0-17','18-25','26-35','36-45','46-50','51-55', '55+'])
encode.fit(dataT['category'].unique())
dataT['category'] = encode.transform(dataT['category'])

#encode.fit(['M','F'])
#data['Gender'] = encode.transform(data['Gender'])

#Total Purchases of Specific Product category 1 items among male & female customers
males_spp = data[data['gender']==0].groupby(['category']).purchase.count()
females_spp = data[data['gender']==1].groupby(['category']).purchase.count()

pl.figure(figsize =(30,20))
N = 21
ind = np.arange(N)
width = .35
plt.bar(ind, males_spp, width, label='Male Customers')
plt.bar(ind + width, females_spp, width, label='Female Customers')





plt.ylabel('Total Purchases', fontsize=30)
plt.xlabel('Product Category', fontsize=30)
plt.title('Total Purchases of Product category items among male & female customers', fontsi

#plt.xticks(ind + width / 2)


plt.xticks(ind + width / 2,males_spp.index,rotation='vertical',fontsize=20)
plt.yticks(fontsize=20)
plt.legend(loc='best')
plt.show()
```

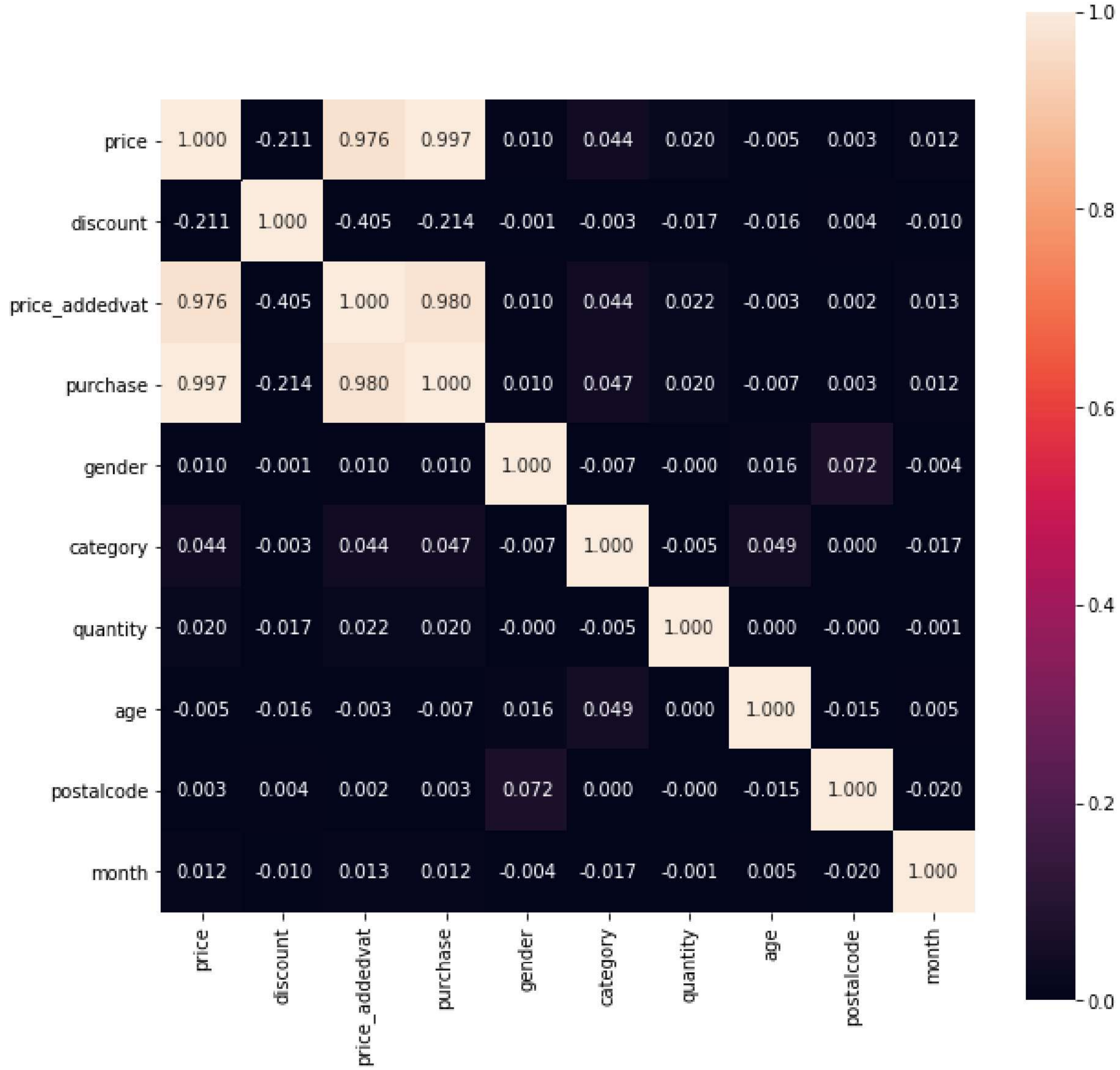


In [31]:

```
#Data Transformation
encode = LabelEncoder()
#encode.fit(['0-17','18-25','26-35','36-45','46-50','51-55', '55+'])
encode.fit(dataT['category'].unique())
dataT['category'] = encode.transform(dataT['category'])

#encode.fit(['M','F'])
#data['Gender'] = encode.transform(data['Gender'])
```

In [32]:

```
#Correlation matrix & Heatmap - Finding correlation
pl.figure(figsize =(10,10))
corrmat = dataT[['price','discount','price_addedvat','purchase','gender','category','quanti
sns.heatmap(corrmat, annot=True, fmt='.3f', vmin=0, vmax=1, square=True);
plt.show()
```



	price	discount	price_addedvat	purchase	gender	category	quantity	age	postalcode	month
price	1.000	-0.211	0.976	0.997	0.010	0.044	0.020	-0.005	0.003	0.012
discount	-0.211	1.000	-0.405	-0.214	-0.001	-0.003	-0.017	-0.016	0.004	-0.010
price_addedvat	0.976	-0.405	1.000	0.980	0.010	0.044	0.022	-0.003	0.002	0.013
purchase	0.997	-0.214	0.980	1.000	0.010	0.047	0.020	-0.007	0.003	0.012
gender	0.010	-0.001	0.010	0.010	1.000	-0.007	-0.000	0.016	0.072	-0.004
category	0.044	-0.003	0.044	0.047	-0.007	1.000	-0.005	0.049	0.000	-0.017
quantity	0.020	-0.017	0.022	0.020	-0.000	-0.005	1.000	0.000	-0.000	-0.001
age	-0.005	-0.016	-0.003	-0.007	0.016	0.049	0.000	1.000	-0.015	0.005
postalcode	0.003	0.004	0.002	0.003	0.072	0.000	-0.000	-0.015	1.000	-0.020
month	0.012	-0.010	0.013	0.012	-0.004	-0.017	-0.001	0.005	-0.020	1.000

In [*]:

```
data_csv=dataT

data_csv.to_csv('DataNew.csv')
```

In [ ]: