



E-COMMERCE DATASET- CUSTOMER SEGMENTATION

Author : Saurabh Anand Date: Dec, 20 2017



Contents

☐ Know the Client – Problem Statement and Solution

☐ What's the client current approach?

☐ Project Objective

☐ Data Acquisition and Description

☐ Data Preparation

☐ Data Wrangling and Exploratory Data Analysis

☐ Dimensionality Reduction

☐ Principle Component Analysis

☐ Cluster Analysis- Machine Learning Algorithm

☐ Conclusion and Summary

Know the Client – Problem Statement and Required Solution

Client Introduction

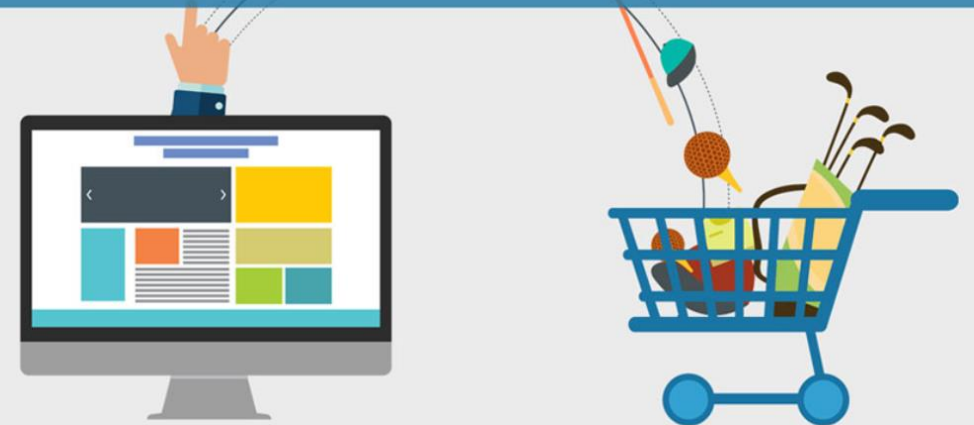
Our client is an online retailer based in the UK. They sell all-occasion gifts, and many of their customers are wholesalers.



Problem statement and Required Solution

- ✓ Most of their customers are from the UK, but they have a small percent of customers from other countries.
- ✓ They want to create groups of these international customers based on their previous purchase patterns.
- ✓ Their goal is to provide more tailored services and improve the way they market to these international customers

Transform into a Modernized
Ecommerce Contender



Contents

☐ Know the Client – Problem Statement and Solution

☐ What's the client current approach?

☐ Project Objective

☐ Data Acquisition and Description

☐ Data Preparation

☐ Data Wrangling and Exploratory Data Analysis

☐ Dimensionality Reduction

☐ Principle Component Analysis

☐ Cluster Analysis- Machine Learning Algorithm

☐ Conclusion and Summary

What's the client current approach?

Currently, the retailer simply groups their international customers by country. As you'll see in the project, this is quite inefficient because:

- ✓ There's a large number of countries (which defeats the purpose of creating groups)
- ✓ Some countries have very few customers.
- ✓ This approach treats large and small customers the same, regardless of their purchase patterns



Contents

☐ Know the Client – Problem Statement and Solution

☐ What's the client current approach?

☐ Project Objective

☐ Data Acquisition and Description

☐ Data Preparation

☐ Data Wrangling and Exploratory Data Analysis

☐ Dimensionality Reduction

☐ Principle Component Analysis

☐ Cluster Analysis- Machine Learning Algorithm

☐ Conclusion and Summary

Project Objective

The retailer has hired us to help them create customer clusters, i.e. "customer segments," through a data-driven approach.

- ✓ They've provided us a dataset of past purchase data at the transaction level.
- ✓ Our task is to build a clustering model using that dataset.
- ✓ Our clustering model should factor in both **aggregate sales patterns** and **specific items purchased**.

Customer Segmentation



Contents

- ☐ Know the Client – Problem Statement and Solution
- ☐ What's the client current approach?
- ☐ Project Objective
- ☐ Data Acquisition and Description
- ☐ Data Preparation
- ☐ Data Wrangling and Exploratory Data Analysis
- ☐ Dimensionality Reduction
- ☐ Principle Component Analysis
- ☐ Cluster Analysis- Machine Learning Algorithm
- ☐ Conclusion and Summary

Data Acquisition and Description

Data Acquisition

This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers



Data Description

- ✓ **InvoiceNo:** Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter 'c', it indicates a cancellation.
- ✓ **StockCode:** Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.
- ✓ **Description:** Product (item) name. Nominal.
- ✓ **Quantity:** The quantities of each product (item) per transaction. Numeric.
- ✓ **InvoiceDate:** Invoice Date and time. Numeric, the day and time when each transaction was generated.
- ✓ **UnitPrice:** Unit price. Numeric, Product price per unit in sterling.
- ✓ **CustomerID:** Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.
- ✓ **Country:** Country name. Nominal, the name of the country where each customer resides

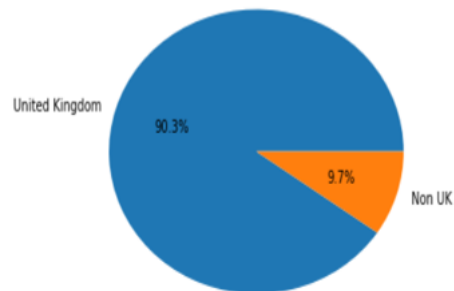
Contents

- ☐ Know the Client – Problem Statement and Solution
- ☐ What's the client current approach?
- ☐ Project Objective
- ☐ Data Acquisition and Description
- ☐ Data Preparation
- ☐ Data Wrangling and Exploratory Data Analysis
- ☐ Dimensionality Reduction
- ☐ Principle Component Analysis
- ☐ Cluster Analysis- Machine Learning Algorithm
- ☐ Conclusion and Summary

Data Preparation

Acquired data was loaded and was prepared for analysis. As, we can see the dataset contains most of the transactional record from UK, so we will create a dataframe for international transaction i.e other than United Kingdom for our analysis and solution

```
1 pie_data = []  
2 pie_data.append(len(df1))  
3 pie_data.append(len(df2))  
4 plt.pie(pie_data, labels=['United Kingdom', 'Non UK'], autopct='%1.1f%%',)  
5 plt.show()
```



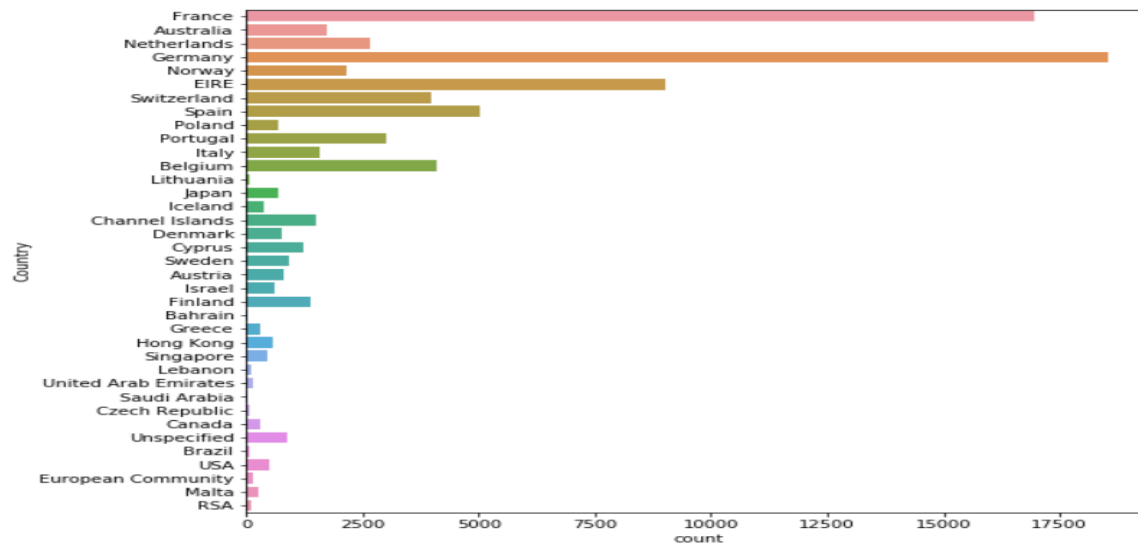
Contents

- ☐ Know the Client – Problem Statement and Solution
- ☐ What's the client current approach?
- ☐ Project Objective
- ☐ Data Acquisition and Description
- ☐ Data Preparation
- ☐ Data Wrangling and Exploratory Data Analysis
- ☐ Dimensionality Reduction
- ☐ Principle Component Analysis
- ☐ Cluster Analysis- Machine Learning Algorithm
- ☐ Conclusion and Summary

Data Wrangling and Exploratory Data Analysis

Even though we eventually want customer-level data, it's still helpful to do some basic exploratory analysis at the transaction level

```
1 # Make figsize 9x9
2 plt.figure(figsize=(9,9))
3
4 # Bar plot by country
5 sns.countplot(y = 'Country', data=df)
6 plt.show()
```



Data Cleaning

Data Cleaning-

- ✓ Drop observations with missing customer ID's
- ✓ Convert the CustomerID's and Invoice IDs from floats into strings
- ✓ Saved the cleaned dataset into CSV

Display the number of missing observations for each feature.

```
In [73]: 1 df.isnull().sum()
```

```
Out[73]: InvoiceNo      0
StockCode      0
Description     0
Quantity        0
InvoiceDate     0
UnitPrice       0
CustomerID    2898
Country         0
dtype: int64
```

Drop observations with missing customer ID's.

```
In [74]: 1 df.dropna(inplace=True)
```

```
In [75]: 1 df.isnull().sum()
```

```
Out[75]: InvoiceNo      0
StockCode      0
Description     0
Quantity        0
InvoiceDate     0
UnitPrice       0
CustomerID      0
Country         0
dtype: int64
```

Data Aggregation

For the better analysis purpose all the possible data combination were done and data was rolled up at customer level following below steps

- ✓ Aggregating product data by customer
- ✓ Aggregate sales data by customer
- ✓ Aggregating Cart data to Customer level
- ✓ Joining All data frames created above and converted into a csv '*Customer_df_analysis.csv*'

```
In [90]: 1 customer_df = invoice_data.join([product_data, sales_data, agg_cart_data])
        2 customer_df.head()
```

```
Out[90]:
```

	total_transactions	total_products	total_unique_products	total_sales	avg_product_value	avg_cart_value	min_cart_value	max_cart_value
CustomerID								
12347	7	364	103	8620.00	23.681319	1231.428571	449.64	2588.64
12348	4	62	22	3594.48	57.975484	898.620000	454.88	1785.60
12349	1	146	73	3515.10	24.076027	3515.100000	3515.10	3515.10
12350	1	34	17	668.80	19.670588	668.800000	668.80	668.80
12352	11	180	59	4051.45	22.508056	368.313636	-463.80	1680.60

Contents

- ☐ Know the Client – Problem Statement and Solution
- ☐ What's the client current approach?
- ☐ Project Objective
- ☐ Data Acquisition and Description
- ☐ Data Preparation
- ☐ Data Wrangling and Exploratory Data Analysis
- ☐ Dimensionality Reduction
- ☐ Principle Component Analysis
- ☐ Cluster Analysis- Machine Learning Algorithm
- ☐ Conclusion and Summary

Dimensionality Reduction

As, our client wishes to incorporate information about **specific item purchases** into the clusters. For example, our model should be more likely to group together customers who buy similar items.

- ✓ Now, we prepared individual item features for our clustering algorithms.
- ✓ We'll introduce a simple way to reduce the number of dimensions by applying thresholds

Below are the steps used as a part of dimensionality reduction-

Import Cleaned dataset

- ✓ Applied Toy example on sample data for reduce dimensionality
- ✓ Applied Toy example to entire dataset- High Dimensionality
- ✓ Created a Threshold to reduce dimensionality of entire dataframe

Here, take a look:

```
In [15]: 1 # Display first 5 rows of top_20_item_data  
        2 top_20_item_data.head(5)
```

```
Out[15]:
```

	23245	22961	21080	22630	20726	20719	20750	86099B	23084	20725	21212	22551	22629	21731	22328	22556	22554	22423	22326	POST
CustomerID																				
12347	0	0	0	0	0	8	0	0	6	0	0	0	0	10	0	0	0	8	0	0
12348	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8
12349	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2
12350	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	2
12352	4	0	0	2	0	0	0	0	0	0	0	0	0	2	0	0	0	4	0	10

Contents

- ☐ Know the Client – Problem Statement and Solution
- ☐ What's the client current approach?
- ☐ Project Objective
- ☐ Data Acquisition and Description
- ☐ Data Preparation
- ☐ Data Wrangling and Exploratory Data Analysis
- ☐ Dimensionality Reduction
- ☐ Principle Component Analysis
- ☐ Cluster Analysis- Machine Learning Algorithm
- ☐ Conclusion and Summary

Principle Component Analysis

Principal Component Analysis, or PCA, is a popular dimensionality reduction technique. PCA seeks to create new features by finding linear combinations of your original ones. These new features, called **principal components**, are meant to maximize the "**explained variance**," which we'll explain further in the module.

- ✓ Here, we'll prepare individual item features for our clustering algorithms, except this time we'll use PCA instead of thresholding.
- ✓ PCA is especially effective when you have many correlated features.

Note- PCA creates new features that replace the original ones.

Below are the steps performed

- ✓ Item data - Principal Component Analysis
- ✓ Explained Variance
- ✓ Dimensionality Reduction

Principle Component Analysis(Contd)

Item data - Principal Component Analysis

- ✓ Scaled item_data, which we saved before and named it 'item_data_scaled'
- ✓ Next, initialize and fit an instance of the PCA transformation and generate new "principal component features" from item_data_scaled.

Initialize and fit an instance of the PCA transformation.

- Keeping all of the components for now, not passing any argument

```
In [37]: 1 # Initialize and fit a PCA transformation
         2 pca = PCA()
```

Generating new "principal component features" from item_data_scaled.

```
In [38]: 1 # Generate new features
         2 PC = pca.fit_transform(item_data_scaled)
         3
         4 # Display first 5 rows
         5 PC[:5]
```

```
Out[38]: array([[ 5.18063695e+00,  7.18817541e-01,  1.54869085e+00, ...,
                  -2.14468752e-05,  9.72318157e-15,  2.89688263e-17],
                 [-3.36799957e+00, -3.44660478e+00,  7.61253706e-01, ...,
                  -1.62949609e-04,  9.72318157e-15,  2.89688263e-17],
```

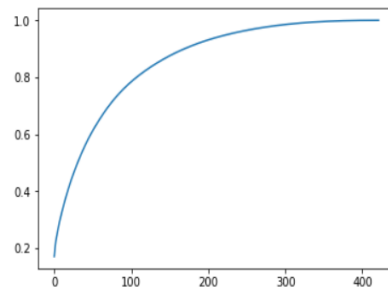
Principle Component Analysis(Contd)

Explained Variance

It's very helpful to calculate and plot the explained variance. This will tell us the total amount of variance we'd capture if we kept up to the n-th component.

- ✓ First, we'll use `np.cumsum()` to calculate the cumulative explained variance.
- ✓ Then, we'll plot it so we can see how many features we'd need to keep in order to capture most of the original variance.
- ✓ Initialize and fit another PCA transformation with 150 components and save it as 'pca_item_data.csv'

```
In [39]: 1 # Cumulative explained variance
          2 cumulative_explained_variance = np.cumsum(pca.explained_variance_ratio_)
          3
          4 # Plot cumulative explained variance
          5 plt.plot(range(len(cumulative_explained_variance)), cumulative_explained_variance)
          6 plt.show()
```



```
In [40]: 1 # How much variance we'd capture with the first 150 components
          2 cumulative_explained_variance[150]
```

Out[40]: 0.87588010485156753

```
9
10 items_pca.head(5)
```

Out[42]:

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10	...	PC141	PC142	PC143	PC14
CustomerID															
12347	5.180637	0.718818	1.548891	-0.007762	-2.314253	-2.536359	0.967863	-0.735692	0.997183	6.819645	...	-0.042104	0.093376	0.351750	-0.21
12348	-3.368000	-3.446605	0.761256	0.806926	-1.031881	-1.086750	-0.431494	0.157940	-0.439192	-0.363441	...	-0.358197	1.824972	-2.023484	4.60
12349	-0.569100	-1.702421	0.993412	0.784203	0.098077	-1.821577	-1.916209	-0.307812	-0.832751	1.634552	...	-0.976433	-0.931319	-4.301407	-1.62
12350	-3.942596	-3.817902	1.186881	-0.839754	-1.408507	0.811212	-0.541209	0.313549	0.416172	0.503025	...	-1.509877	0.895428	0.005363	0.12

Contents

- ☐ Know the Client – Problem Statement and Solution
- ☐ What's the client current approach?
- ☐ Project Objective
- ☐ Data Acquisition and Description
- ☐ Data Preparation
- ☐ Data Wrangling and Exploratory Data Analysis
- ☐ Dimensionality Reduction
- ☐ Principle Component Analysis
- ☐ Cluster Analysis- Machine Learning Algorithm
- ☐ Conclusion and Summary

Cluster Analysis- Machine Learning Algorithm

Cluster Analysis

For clustering problems, the chosen input features are usually more important than which algorithm you use. Here, we'll apply the K-Means algorithm to 3 different feature sets for comparison

Import 3 CSV files we've saved before

- ✓ 'Customer_df_analysis.csv' as base_df.
- ✓ 'threshold_item_data.csv' as threshold_item_data.
- ✓ 'pca_item_data.csv' as pca_item_data.

Feature1- Result after applying K-Means algorithm to base_df with random_seed=126 and three cluster

Feature2- Result after applying K-Means algorithm to threshold_item_data with random_seed=126 and three cluster

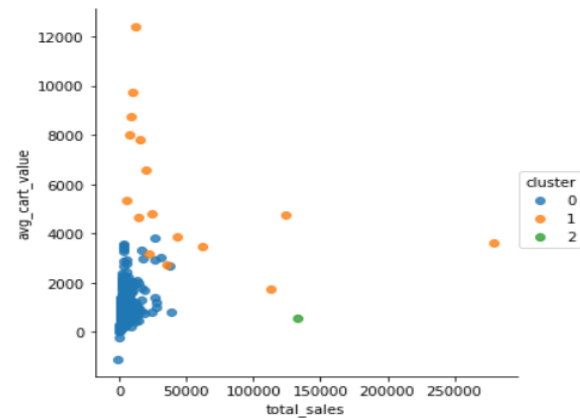
Feature3- Result after applying K-Means algorithm to pca_item_data with random_seed=126 and three cluster

Cluster Analysis- Machine Learning Algorithm(Contd)

Cluster Analysis- Feature 1

Result after applying K-Means algorithm to base_df with random_seed=126 and three cluster

```
: 1 # Scatterplot, colored by cluster  
2 sns.lmplot(x='total_sales', y='avg_cart_value', data=base_df, hue='cluster', fit_reg=False)  
3 plt.show()
```

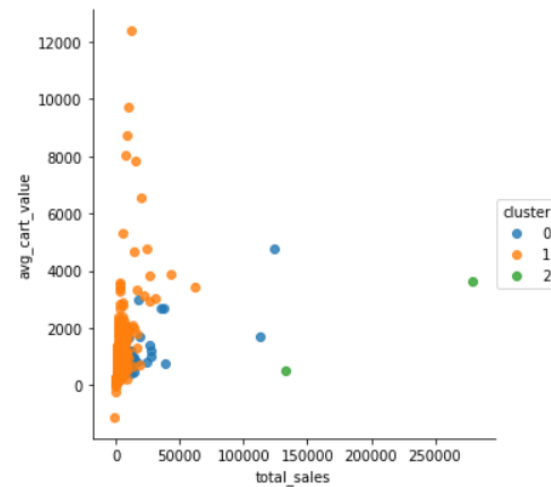


Cluster Analysis- Machine Learning Algorithm(Contd)

Cluster Analysis- Feature 2

Result after applying K-Means algorithm to threshold_item_data with random_seed=126 and three cluster

```
: 1 # Scatterplot, colored by cluster  
2 sns.lmplot(x='total_sales', y = 'avg_cart_value', data=threshold_df, hue='cluster', fit_reg=False)  
3 plt.show()
```

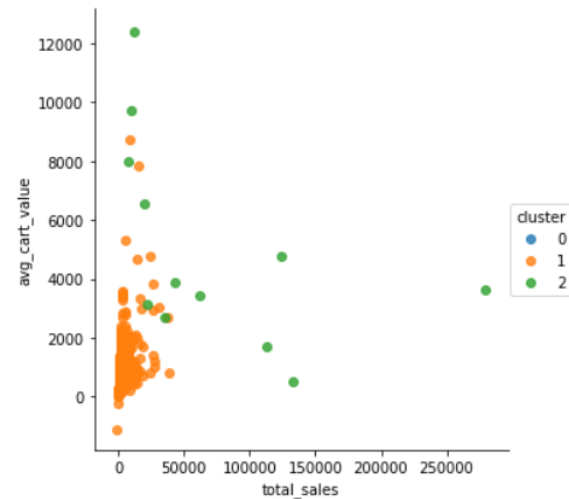


Cluster Analysis- Machine Learning Algorithm(Contd)

Cluster Analysis- Feature 3

Result after applying K-Means algorithm to `pca_item_data` with `random_seed=126` and three cluster

```
: 1 # Scatterplot, colored by cluster
  2 sns.lmplot(x='total_sales', y='avg_cart_value', data=pca_df, hue='cluster', fit_reg=False)
  3 plt.show()
```

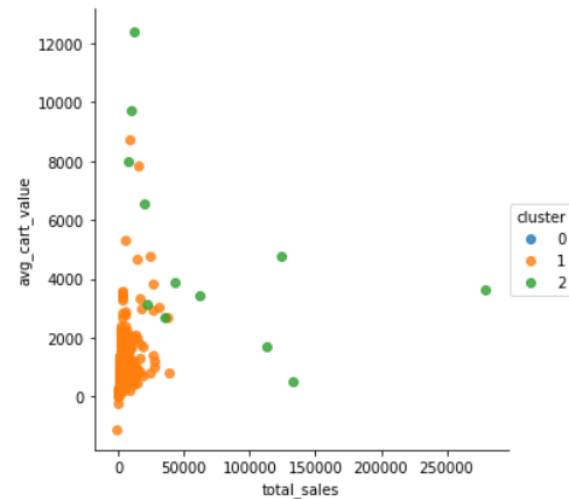


Cluster Analysis- Machine Learning Algorithm(Contd)

Cluster Analysis- Feature 3

Result after applying K-Means algorithm to `pca_item_data` with `random_seed=126` and three cluster

```
: 1 # Scatterplot, colored by cluster
  2 sns.lmplot(x='total_sales', y='avg_cart_value', data=pca_df, hue='cluster', fit_reg=False)
  3 plt.show()
```



Model Comparison

Comparing features developed before using scikit learn- adjusted random score as below-

1. Compare base_df.cluster and threshold_df.cluster

We can see the adjusted Rand index between `base_df.cluster` and `threshold_df.cluster` like so:

```
In [153]: 1 # Similary between base_df.cluster and threshold_df.cluster  
          2 adjusted_rand_score(base_df.cluster, threshold_df.cluster)
```

```
Out[153]: 0.11352673471033736
```

2. Compare base_df.cluster and pca_df.cluster

Finally, display the adjusted Rand index between `base_df.cluster` and `pca_df.cluster`.

```
In [154]: 1 # Similary between base_df.cluster and pca_df.cluster  
          2 adjusted_rand_score(base_df.cluster, pca_df.cluster)
```

```
Out[154]: 0.78161752101988669
```

Contents

- ☐ Know the Client – Problem Statement and Solution
- ☐ What's the client current approach?
- ☐ Project Objective
- ☐ Data Acquisition and Description
- ☐ Data Preparation
- ☐ Data Wrangling and Exploratory Data Analysis
- ☐ Dimensionality Reduction
- ☐ Principle Component Analysis
- ☐ Cluster Analysis- Machine Learning Algorithm
- ☐ Conclusion and Summary

- [illegible]

THANK
YOU

