Married Couple Identification

By Thomas FitzGerald & Mei Maddox

Executive Summary

Financial institutions gather and retain large amounts of customer data, including spending and banking habits. Married couples handle finances in a variety of ways. If it were possible to create a typology of how married couples handle their money, it should be possible to use that typology to identify otherwise unknown married couples via their banking habits. To test this, banking and demographic data was used to group pairs of bank customers. Accomplishing this requires overcoming several challenges in terms of dimension reduction to make the problem practically computable. Given a set of known married couples, these methods could be used to identify groups, and, with further refinement, identify married couples based purely on spending habits.

Research Problem

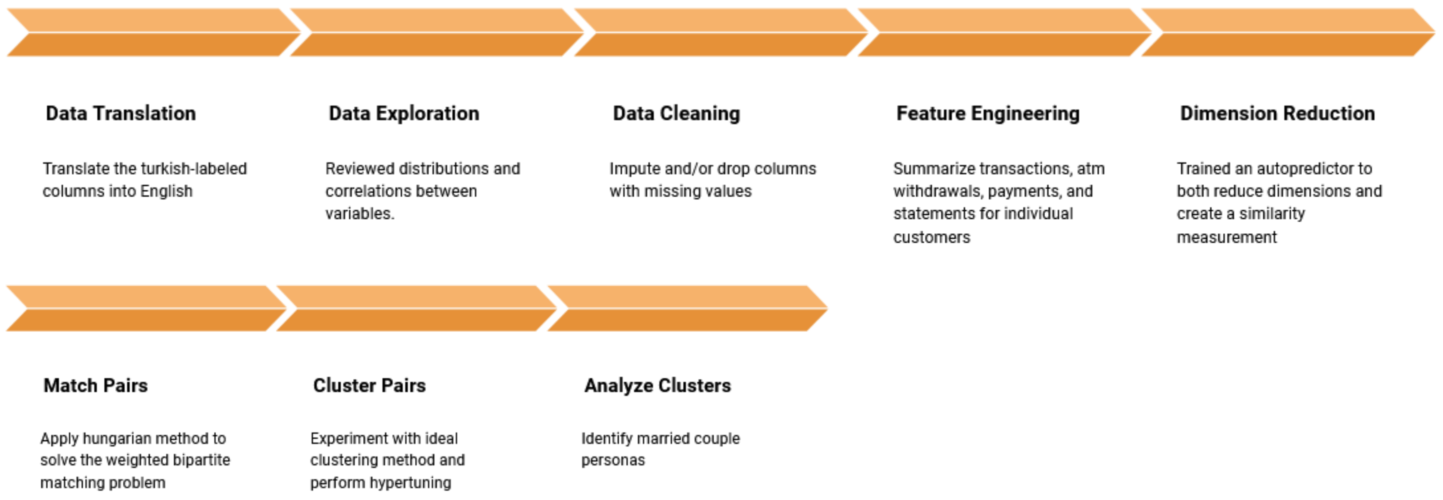
Given a set of banking data, identify married couples among a large set of account holders. Available information includes:

1. Marital status of customer
2. Basic demographic information
3. Banking behavior (transactions, payments, bank visits)

Although the customer's marital status is available, the identity of their partner (if applicable) is *not* provided. In other words, there is no "ground truth" so the solution will involve developing a typology for use when or if additional information becomes available.

Primary applications for a solution would most likely either fall into marketing or identifying potential credit risks, on the theory that married couples share a degree of monetary burden, and identifying high risk in one partner may be a way of preemptively identifying risk in the other.

The following graphic illustrates the data science pipeline of this project


Data Translation
Translate the turkish-labeled columns into English
Data Exploration
Reviewed distributions and correlations between variables.
Data Cleaning
Impute and/or drop columns with missing values
Feature Engineering
Summarize transactions, atm withdrawals, payments, and statements for individual customers
Dimension Reduction
Trained an autopredictor to both reduce dimensions and create a similarity measurement
Match Pairs
Apply hungarian method to solve the weighted bipartite matching problem
Cluster Pairs
Experiment with ideal clustering method and perform hypertuning
Analyze Clusters
Identify married couple personas

The Data

The 12-month financial data comes from a Turkish bank, Akbank, in the form of seven tables:

Table	Dimensions	Summary
Customer Demographics(kitle_orneklem_100k)	103,210 rows 28 columns	Customer Demographic Information
Transfers (havale_bil)	345,037 rows 7 columns	Inter-bank transfers
Statements(kk_ekstre_bil)	1,237,510 rows 8 columns	Customer bank statements
Payments(kk_ekstre_odm_bil)	1,434,800 rows 5 columns	Payments made on credit card
Branch Visits(sube_isl_ziy)	539,428 rows 7 columns	Visits made to bank branches by customer
Transactions(kk_har_bilgi)	9,334,625 rows 11 columns	Transactions made, along with spending category and method
ATM(atm_islem)	2,726,611 rows 9 columns	Customer withdrawals and deposits

Data Translation

As the source of the data was Turkish, the first task upon receiving the dataset was to translate the variable names to English. This would be useful to both avoid errors and to disambiguate some of the column names. Translations were drawn from the included schema, with supplemental translations via google translate. Much of the data in each table was compressed as part of feature engineering, so those values were only translated as needed.

Data Cleaning

The initial data had few apparent typos/errors or missing values, with only 13 of the 75 columns containing missing values. The following adjustments were made as part of preprocessing:

1. The Transaction table contained data on merchant x/y coordinates, approximately 60% of which was missing. These columns were dropped.
2. The 'trans_spending_category' column in the Transaction table was missing labels on approximately 12.8% of entries. These were replaced with an 'unknown' category.
3. In the Customer Demographics table, approximately 2.3% of entries in the customer_median_income column were empty. These were replaced with the median income value.
4. All other missing data was left as-is.

Feature Engineering

The dataset contains a large amount of multi-row data on each individual customer. To do pairwise comparisons of each potential married couple, it is necessary to summarize this information at the customer level, which was accomplished via a variety of feature engineering methods. This put additional features at a premium, so when there was ambiguity about whether to keep or drop a feature, it was generally dropped.

Table	Feature Engineering Methods
Customer Demographics	12 risk code columns converted into 6 summary variables Each variable counted frequency customer was flagged at that risk level
Transfers	No features retained
Statements	Converted into 2 summary variables Mean statement amount* Standard deviation of statement amount*
Payments	Converted into 3 summary variables Mean payment amount* Standard Deviation of payment amount* Monthly payment frequency*
Branch Visits	Converted into 11 variables 7 counting visits on each day of the week 4 counting visit time (very early, early, late, very late) Branch visit x/y coordinate was converted into 3 summary variables Mean x coordinate Mean y coordinate Variance of distance between branch coordinate to mean coordinate (representing travel distance)
Transactions	Converted into 10 variables Average amount spent in 5 most common spending categories* Average monthly frequency of spending in 5 most common spending categories *
ATM	Converted into 2 variables Mean withdrawal amount* Standard deviation of withdrawal amount*

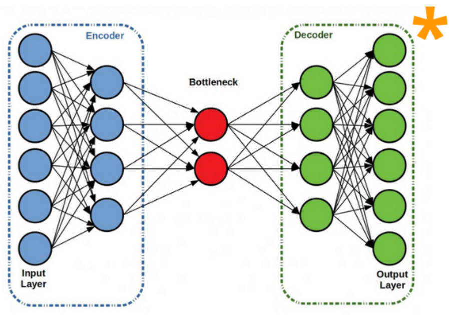
*Only transactions in Turkish Lira were retained. This represented >95% of total transactions.

Dimension Reduction

Through the preprocessing techniques mentioned above, 65,640 customer demographics and financial habits were represented via 42 columns, with each row representing a single customer. Fortytwo features is still a sizable amount considering the number of customers. Therefore, further dimension reduction was deemed necessary to avoid data scarcity and thus issues regarding overfitting and poor generalization. Reduced training times from the lower-dimensional space was an added bonus.

Principle component analysis (PCA) is a common method of dimension reduction which performs linear transformations on the features to project them into a lower dimension. Most of the data, however, was *not* highly linearly correlated with itself except for statement mean and standard deviation to payment mean and standard deviation, respectively. Even said relationships were better represented non-linearly. Therefore, instead of using PCA we opted to use an autopredictor for self-supervised dimension reduction.

An autopredictor is an extension of an autoencoder, a type of neural network where the input and target are identical. The number of nodes in subsequent hidden layers within an autoencoder decrease and then



increase, allowing for a ‘learned’ method of data compression and reconstruction (see image below).

In order to be able to accurately compress and reconstruct data, the autoencoder must ‘learn’ a similarity algorithm. In other words, an autoencoder is a self-supervised method for creating a similarity algorithm, eliminating the need to manually devise a similarity measurement for couple matching. Unforteunetely, an autoencoder does not leverage domain knowledge for educated inferences (i.e. two customers with identical home coordinates are more likely to be married than two customers with identical spending habits). An autopredictor, theoretically, can indirectly.

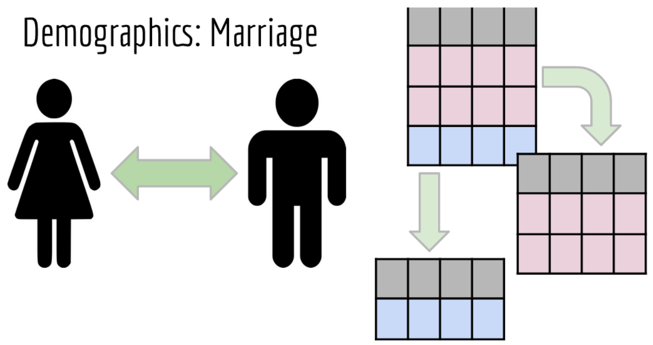
With an autopredictor, the target is a subset of features instead of all features. This allows for certain features to be designated as more significant. In this particular case banking age, bank visit date and time, average spending and frequency, and job status were all removed from the predicted features - leaving 19 of the 42 input features untouched.

Feature(s)	Rationale For Predictive Exclusion
Banking Age	Except for rare cases where they signed up together, most partnerships have one member being longer-established at the bank
Bank Visit Date/Time and Frequency	Several people visit banks at the same time; married couples do not necessarily have the same visitation patterns.
Spending Categories	Married couples do not necessarily have the same spending patterns (i.e. one might buy all groceries,
Job Status	In this day and age there are several types of working dynamics between married couples; in particular, the specific job status’s not insightful (e.g working abroad, paid (special), paid (public), etc.)

Couple Identification

The original customer table had over 100,000 customers; not all of these customers, however, are relevant to the research question. To group all potential married couples, all customers not flagged as ‘married’ in the customer demographics table were filtered out.

At the time of this project, Turkey *only* recognizes marriage as legally between a male and a female. As banks are legal entities we extended this definition to the ‘married’ marital status, splitting the dataset into one for females (13,372 customers) and one for males (52,268 customers). The large discrepancy between the married genders indicates that some married couples do *not* necessarily bank together at Akbank.



The datasets were then split into training, validation, and test sets using a 70:10:20 split. The training and validation sets from *both* genders were fed into the *same* autopredictor. The test set was used as a final measurement of how well the autopredictor performed. Afterwards, all customer data was fed through the autopredictor to obtained the compressed version of the data.

Couples were then generated via cartesian product of the two gendered compressed datasets and their similarity measured. Although pairwise cosine distance is faster to calculate than the euclidean counterpart, the latter was used for weighting similarity as it easily allowed for the use of the hungarian method to solve the weighted bipartite matching problem.

Although calculating similarity (distance) for the approximately 700 million potential couples was feasible, running the matching algorithm was *not* - the hungarian method runs in $O(n^3)$ time. A newer, more optimized method ($O(n^{3/2}log(n))$) was mostly implemented, but dismissing non-viable admissible paths was not finalized. Without access to the more optimized method, we were forced to run the couple matching algorithm on a subset of the data.

With no ground truth it is ideal to construct a “probable” couple dataset to act as a ground truth. Initial work on this front manually matched couples based on similarity of home location, age, and how long they were members with the bank. Although this forms an intuitive basis for what would define a married couple, at least in terms of demographic information, it was not rooted in objective information, but rather in supposition. Unfortunately, generating a dataset of substantial size was time consuming and timeline restraints forced us to abandon this effort in favor of a random subset of 1,000 males and 1,000 females.

Alternate Methods for Couple Identification

Research was done in social network analysis as a solution to the similarity measurement calculation, but it became clear early on that there was a major barrier regarding knowledge to implement such method. Deep matching autoencoders were also researched and mostly implemented. This approach is similar to the method described earlier, only that the self-supervised learning of the similarity measurement *and* matching occur *simultaneously*. The gendered datasets are run through two separate identical-architecture, but different weighted, autopredictors. The sum of loss values for a training instance i for females and j for males is then subtracted by a function which compares the similarity of the compressed formats for instance i and j. The subtractive element of this approach was not fully flushed out in the paper and we struggled to try and implement our own method along the same lines as they did. Due to time constraints, we abandoned this approach in favor of the method(s) outlined earlier.

Couple Segmentation

Clustering the matched couples provides insight into the personas or dynamics of the matched couples. To cluster our potential pairs, four clustering methods were tested: DBScan, Spectral, Agglomerative, and K-Means. Initial testing was conducted on single customers to determine the viability of each measure, specifically encoded female customers. All clustering was run using the sklearn library.

DBScan returned consistently poor results, either by failing to cluster the majority of the dataset, or by generating an excessive number of clusters. This is mostly likely a result of the extremely open structure of the data. DBScan is better suited to densely clustered data, and based on our test results, was poorly suited to this task.

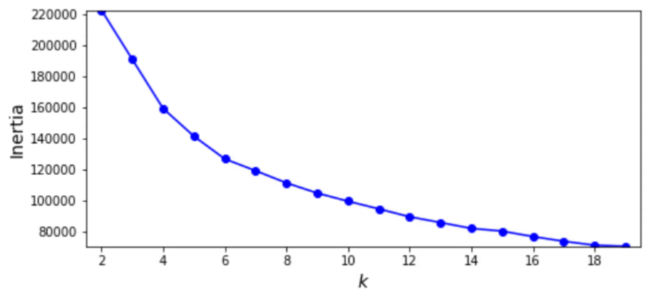
Epsilon	Min_samples	Silhouette Score	Cluster Count	Unclustered
2	2	-0.023	2	13370
100	5	-0.215	8	13328
1000	10	-0.596	18	13179
10000	10	.271	257	1841

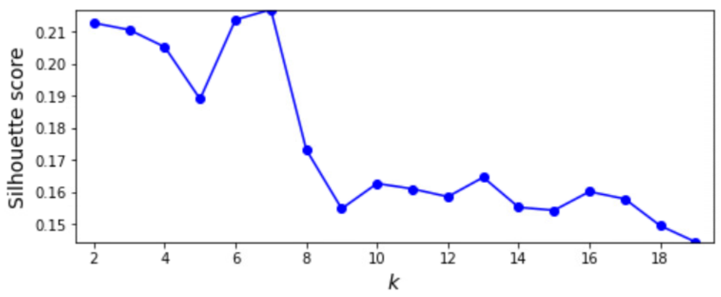
Spectral clustering ran too slowly for practical use. Although it would complete with <200 entries in a reasonable amount of time, running even 1,000 test entries took 15-20 minutes. For practical purposes, it was decided to focus on more efficient methods.

In initial testing, agglomerative produced competitive silhouette scores with Kmeans. However, when we scaled up to the full model, it experienced a significant drop in performance compared to Kmeans. This, combined with lower familiarity with the model, led us to drop agglomerative as a clustering method.

The clustering method we finally settled on for our dataset was Kmeans. Although initial testing was conducted using the individual customers, Kmeans was run against a subset of potential pairs. Each subset was generated by randomly selecting 1,000 males and 1,000 females from the dataset, then creating a dataset of all potential matches. The results below were run on these randomly generated subsets. Because of this, there was significant variation in results: the charts below were chosen from a run on a fairly typical subset.

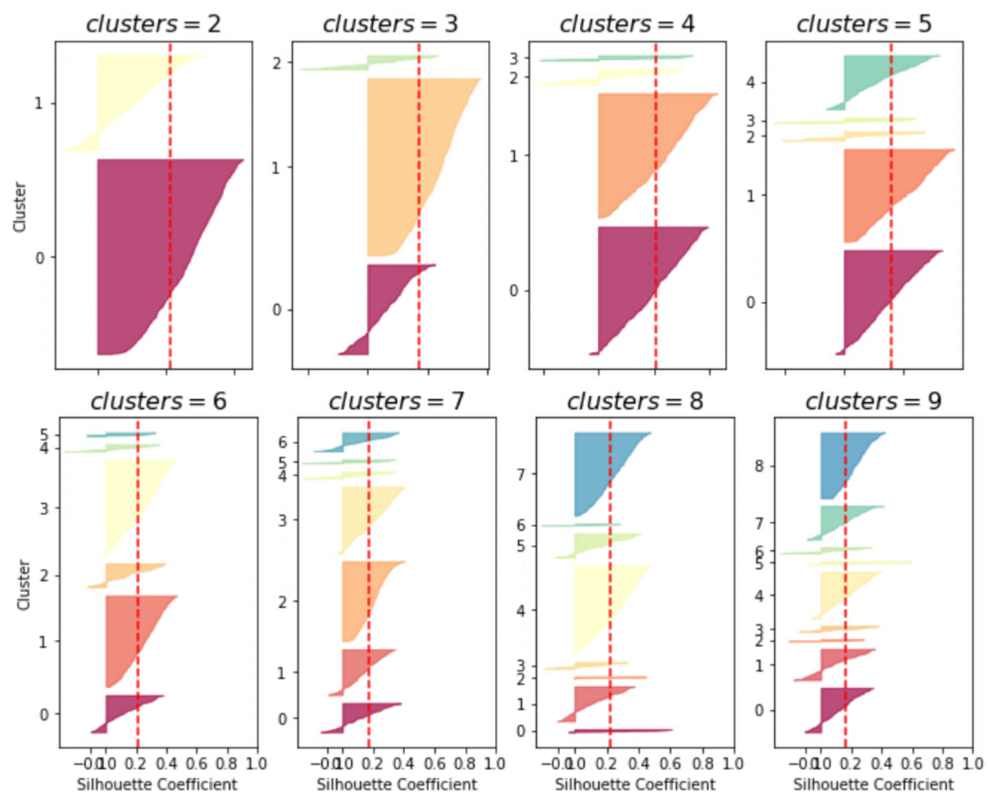
For hyperparameter tuning, we focused on the number of clusters, testing values between 2 and 20, with results below:





In the above images, k represents the number of clusters for our Kmeans algorithm.

Based on these results, we decided to focus on models with 2 to 10 clusters, and switched our focus to the profile of individual clusters generated. The silhouette scores generated in the 6-7 range here are fairly typical of test cases: low values were in the .1 range, with highs in the .5-.6 range.



From the silhouette coefficient graphs above, we can see that, for this particular test set, a kmeans models with 5 to 7 clusters produces a reasonably balanced set of clusters. Although it varied between subsets, 5 was the most consistent across multiple subsets. Therefore 5 clusters was used in the kmeans model for our final test.

Couple Personas

In our final test, we matched couples from a random subset of 1,000 males and 1,000 females and then proceeded to cluster the matchings into 5 groups using Kmeans. Matchings were clustered based on the *similarity* between the pair (difference between encoded values). or example, if both members had similar education levels, and visited branches on similar days of the week, they would be high similarity on both of those dimensions.

Surprisingly enough, most (if not all sometimes) matchings had similar home coordinates, supporting the notion that the autopredictor successfully prioritized location, but surprising nonetheless as it was only 2 out of 42 compared features. Although an exhaustive chart would not conveniently fit in this paper, the table below includes the strongest (or weakest) two dimensions of each cluster, to represent its particular persona.

Cluster	Primary characteristics
0	Similar education level/branch visit habits
1	Similar age, different payment frequencies
2	Different age and spending habits
3	Different age and income
4	Similar income and spending patterns

The clusters produced seem to represent distinct types of relationships, in terms of how each partner relates to the other. The features of each cluster primarily used to make that distinction also seem to vary: demographic information, like age, income, and education level, spending habits, and payment habits all feature prominently, although demographic information, which made up close to half of the input variables, features most prominently.

One interesting thing to note is that the two clusters primarily defined by dissimilarity between each member of the pair are also the smallest two, clusters #2 and 3, while the 3 clusters primarily defined by similarity are comparatively much larger.

Limitations

The lack of ground truth for this problem is the biggest limitation faced. Without knowing who, if anyone, in this dataset are married couples, there's no way to tell if particular types of couples fall into any of the clusters developed by this method.

An additional layer of missing information is other banking/spending behavior. In looking at how two couples spend money, make payments, etc. there is an implicit assumption that all spending made by that couple is made through using the bank in our dataset. All transactions through other financial institutions are absent, so there's an unknown set of transactions for an unknown subset of customers that we cannot account for.

In terms of technical limitations, the biggest problem faced was the inability to run the full dataset simultaneously. Although which and how many potential pairs are true married couples is unavailable information, any married couples that are in the dataset are even less likely to be found from randomly generated subsets of male and female customers. Running random subsets compounds the central difficulty of this problem.

Another major technical limitation was the loss of data during feature engineering. Some information, particularly regarding inter-account transfers, or usage of non-Lira currency, would likely be strong factors in identifying married couples, but were omitted in an attempt to minimize the amount of data being processed. Once other technical limitations are resolved, adding these features would be a good next step.

A group of likely married couples was developed as a test case, but was not finalized and thus utilized. Time-permitting, a larger and less biased "plausible" couples dataset would be able to serve as a ground truth.

Conclusion

Overall, it's difficult to provide meaningful analysis without a ground truth to test against, however, our initial work seems to successfully match couples with distinct relationship dynamics. The autopredictor used was able to significantly reduce the amount of data to process whilst learning a similarity algorithm that prioritized important features like customer home location. Matched couples were clustered into discernable types groups, but conclusions about who is married or unmarried cannot be drawn from this without additional information.