Hw1 & hw2

A screen shot of a computer code

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screen shot of a computer code

Description automatically generated

A screenshot of a phone

Description automatically generatedA screenshot of a graph

Description automatically generatedA screenshot of a computer

Description automatically generated

Hw3

A screenshot of a graph

Description automatically generatedA graph with blue and white bars

Description automatically generatedA screenshot of a graph

Description automatically generated

Hw1

*import* pandas *as* pd

*import* numpy *as* np

*import* featuretools *as* ft

*import* warnings

warnings.filterwarnings('ignore')

ft.\_\_version\_\_

'0.16.0'

clients = pd.read\_csv('./clients.csv', parse\_dates = ['joined'])

clients.head()

|  | **client\_id** | **joined** | **income** | **credit\_score** |
| --- | --- | --- | --- | --- |
| 0 | 46109 | 2002-04-16 | 172677 | 527 |
| 1 | 49545 | 2007-11-14 | 104564 | 770 |
| 2 | 41480 | 2013-03-11 | 122607 | 585 |
| 3 | 46180 | 2001-11-06 | 43851 | 562 |
| 4 | 25707 | 2006-10-06 | 211422 | 621 |

clients.isna().sum()

client\_id 0

joined 0

income 0

credit\_score 0

dtype: int64

loans = pd.read\_csv('./loans.csv', parse\_dates = ['loan\_start', 'loan\_end'])

loans.head()

|  | **client\_id** | **loan\_type** | **loan\_amount** | **repaid** | **loan\_id** | **loan\_start** | **loan\_end** | **rate** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 46109 | home | 13672 | 0 | 10243 | 2002-04-16 | 2003-12-20 | 2.15 |
| 1 | 46109 | credit | 9794 | 0 | 10984 | 2003-10-21 | 2005-07-17 | 1.25 |
| 2 | 46109 | home | 12734 | 1 | 10990 | 2006-02-01 | 2007-07-05 | 0.68 |
| 3 | 46109 | cash | 12518 | 1 | 10596 | 2010-12-08 | 2013-05-05 | 1.24 |
| 4 | 46109 | credit | 14049 | 1 | 11415 | 2010-07-07 | 2012-05-21 | 3.13 |

loans.isna().sum()

client\_id 0

loan\_type 0

loan\_amount 0

repaid 0

loan\_id 0

loan\_start 0

loan\_end 0

rate 0

dtype: int64

payments = pd.read\_csv('./payments.csv', parse\_dates = ['payment\_date'])

payments.head()

|  | **loan\_id** | **payment\_amount** | **payment\_date** | **missed** |
| --- | --- | --- | --- | --- |
| 0 | 10243 | 2369 | 2002-05-31 | 1 |
| 1 | 10243 | 2439 | 2002-06-18 | 1 |
| 2 | 10243 | 2662 | 2002-06-29 | 0 |
| 3 | 10243 | 2268 | 2002-07-20 | 0 |
| 4 | 10243 | 2027 | 2002-07-31 | 1 |

payments.isna().sum()

loan\_id 0

payment\_amount 0

payment\_date 0

missed 0

dtype: int64

es = ft.EntitySet(id='clients')

es = es.entity\_from\_dataframe(entity\_id='clients',

dataframe=clients,

index = 'client\_id',

time\_index='joined')

es = es.entity\_from\_dataframe(entity\_id='loans',

dataframe=loans,

index = 'loan\_id',

time\_index='loan\_start')

es = es.entity\_from\_dataframe(entity\_id='payments',

dataframe=payments,

variable\_types={'missed':

ft.variable\_types.Categorical},

make\_index=True,

index='payment\_id',

time\_index='payment\_date')

es

Entityset: clients

Entities:

clients [Rows: 25, Columns: 4]

loans [Rows: 443, Columns: 8]

payments [Rows: 3456, Columns: 5]

Relationships:

No relationships

es['loans']

Entity: loans

Variables:

loan\_id (dtype: index)

client\_id (dtype: numeric)

loan\_type (dtype: categorical)

loan\_amount (dtype: numeric)

repaid (dtype: numeric)

loan\_start (dtype: datetime\_time\_index)

loan\_end (dtype: datetime)

rate (dtype: numeric)

Shape:

(Rows: 443, Columns: 8)

stats = loans.groupby('client\_id')['loan\_amount'].agg(['sum'])

stats.columns = ['total\_loan\_amount']

stats = clients.merge(stats, left\_on='client\_id',right\_index=True,how='left')

stats.head(10)

|  | **client\_id** | **joined** | **income** | **credit\_score** | **total\_loan\_amount** |
| --- | --- | --- | --- | --- | --- |
| 0 | 46109 | 2002-04-16 | 172677 | 527 | 179032 |
| 1 | 49545 | 2007-11-14 | 104564 | 770 | 205786 |
| 2 | 41480 | 2013-03-11 | 122607 | 585 | 157897 |
| 3 | 46180 | 2001-11-06 | 43851 | 562 | 154017 |
| 4 | 25707 | 2006-10-06 | 211422 | 621 | 159279 |
| 5 | 39505 | 2011-10-14 | 153873 | 610 | 148481 |
| 6 | 32726 | 2006-05-01 | 235705 | 730 | 126032 |
| 7 | 35089 | 2010-03-01 | 131176 | 771 | 138784 |
| 8 | 35214 | 2003-08-08 | 95849 | 696 | 129124 |
| 9 | 48177 | 2008-06-09 | 190632 | 769 | 141063 |

r\_client\_previous = ft.Relationship(es['clients']['client\_id'],es['loans']['client\_id'])

es = es.add\_relationship(r\_client\_previous)

r\_payments = ft.Relationship(es['loans']['loan\_id'],es['payments']['loan\_id'])

es = es.add\_relationship(r\_payments)

es

Entityset: clients

Entities:

clients [Rows: 25, Columns: 4]

loans [Rows: 443, Columns: 8]

payments [Rows: 3456, Columns: 5]

Relationships:

loans.client\_id -> clients.client\_id

payments.loan\_id -> loans.loan\_id

features, feature\_names = ft.dfs(entityset= es,

target\_entity='clients',

agg\_primitives=['mean', 'max','percent\_true','last'],

trans\_primitives=['year', 'month', 'subtract\_numeric', 'divide\_numeric'])

pd.DataFrame(features['MONTH(joined)'].head())

|  | **MONTH(joined)** |
| --- | --- |
| **client\_id** |  |
| 42320 | 4 |
| 39384 | 6 |
| 26945 | 11 |
| 41472 | 11 |
| 46180 | 11 |

pd.DataFrame(features['MEAN(payments.payment\_amount)'].head())

|  | **MEAN(payments.payment\_amount)** |
| --- | --- |
| **client\_id** |  |
| 42320 | 1021.483333 |
| 39384 | 1193.630137 |
| 26945 | 1109.473214 |
| 41472 | 1129.076190 |
| 46180 | 1186.550336 |

pd.DataFrame(features[['MEAN(loans.loan\_amount)','MEAN(loans.rate)','MAX(loans.loan\_amount)','MAX(loans.rate)','LAST(loans.loan\_type)','LAST(loans.loan\_amount)']].head())

|  | **MEAN(loans.loan\_amount)** | **MEAN(loans.rate)** | **MAX(loans.loan\_amount)** | **MAX(loans.rate)** | **LAST(loans.loan\_type)** | **LAST(loans.loan\_amount)** |
| --- | --- | --- | --- | --- | --- | --- |
| **client\_id** |  |  |  |  |  |  |
| 42320 | 7062.066667 | 2.457333 | 13887 | 6.74 | home | 8090 |
| 39384 | 7865.473684 | 3.538421 | 14654 | 9.23 | other | 14654 |
| 26945 | 7125.933333 | 2.855333 | 14593 | 5.65 | cash | 9249 |
| 41472 | 7510.812500 | 3.981250 | 13657 | 9.82 | cash | 10122 |
| 46180 | 7700.850000 | 3.502500 | 14081 | 9.26 | other | 3834 |

Hw2

*import* pandas *as* pd

*import* numpy *as* np

*import* featuretools *as* ft

*from* woodwork.logical\_types *import* Categorical

*import* warnings

warnings.filterwarnings('ignore')

ft.\_\_version\_\_

'1.13.0'

clients = pd.read\_csv('./clients.csv', parse\_dates = ['joined'])

clients.head()

|  | **client\_id** | **joined** | **income** | **credit\_score** |
| --- | --- | --- | --- | --- |
| 0 | 46109 | 2002-04-16 | 172677 | 527 |
| 1 | 49545 | 2007-11-14 | 104564 | 770 |
| 2 | 41480 | 2013-03-11 | 122607 | 585 |
| 3 | 46180 | 2001-11-06 | 43851 | 562 |
| 4 | 25707 | 2006-10-06 | 211422 | 621 |

clients.isna().sum()

client\_id 0

joined 0

income 0

credit\_score 0

dtype: int64

loans = pd.read\_csv('./loans.csv', parse\_dates = ['loan\_start', 'loan\_end'])

loans.head()

|  | **client\_id** | **loan\_type** | **loan\_amount** | **repaid** | **loan\_id** | **loan\_start** | **loan\_end** | **rate** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 46109 | home | 13672 | 0 | 10243 | 2002-04-16 | 2003-12-20 | 2.15 |
| 1 | 46109 | credit | 9794 | 0 | 10984 | 2003-10-21 | 2005-07-17 | 1.25 |
| 2 | 46109 | home | 12734 | 1 | 10990 | 2006-02-01 | 2007-07-05 | 0.68 |
| 3 | 46109 | cash | 12518 | 1 | 10596 | 2010-12-08 | 2013-05-05 | 1.24 |
| 4 | 46109 | credit | 14049 | 1 | 11415 | 2010-07-07 | 2012-05-21 | 3.13 |

loans.isna().sum()

client\_id 0

loan\_type 0

loan\_amount 0

repaid 0

loan\_id 0

loan\_start 0

loan\_end 0

rate 0

dtype: int64

payments = pd.read\_csv('./payments.csv', parse\_dates = ['payment\_date'])

payments.head()

|  | **loan\_id** | **payment\_amount** | **payment\_date** | **missed** |
| --- | --- | --- | --- | --- |
| 0 | 10243 | 2369 | 2002-05-31 | 1 |
| 1 | 10243 | 2439 | 2002-06-18 | 1 |
| 2 | 10243 | 2662 | 2002-06-29 | 0 |
| 3 | 10243 | 2268 | 2002-07-20 | 0 |
| 4 | 10243 | 2027 | 2002-07-31 | 1 |

payments.isna().sum()

loan\_id 0

payment\_amount 0

payment\_date 0

missed 0

dtype: int64

es = ft.EntitySet(id='clients')

es = es.add\_dataframe(dataframe\_name='clients',

dataframe=clients,

index = 'client\_id',

time\_index='joined')

es = es.add\_dataframe(dataframe\_name='loans',

dataframe=loans,

index = 'loan\_id',

time\_index='loan\_start')

es = es.add\_dataframe(dataframe\_name='payments',

dataframe=payments,

logical\_types={'missed':Categorical},

make\_index=True,

index='payment\_id',

time\_index='payment\_date')

es

Entityset: clients

DataFrames:

clients [Rows: 25, Columns: 4]

loans [Rows: 443, Columns: 8]

payments [Rows: 3456, Columns: 5]

Relationships:

No relationships

es['loans']

|  | **client\_id** | **loan\_type** | **loan\_amount** | **repaid** | **loan\_id** | **loan\_start** | **loan\_end** | **rate** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 11140 | 39505 | home | 2274 | 1 | 11140 | 2000-01-26 | 2002-01-29 | 1.00 |
| 11251 | 26326 | home | 2847 | 1 | 11251 | 2000-03-06 | 2001-09-26 | 1.32 |
| 10816 | 49545 | home | 8354 | 1 | 10816 | 2000-03-08 | 2001-08-02 | 0.45 |
| 11965 | 29841 | credit | 6012 | 0 | 11965 | 2000-03-25 | 2002-07-10 | 4.63 |
| 10166 | 41472 | home | 13657 | 1 | 10166 | 2000-04-11 | 2001-09-08 | 5.68 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 11595 | 35089 | other | 773 | 1 | 11595 | 2014-09-26 | 2016-04-23 | 7.63 |
| 10985 | 26695 | other | 13918 | 1 | 10985 | 2014-10-03 | 2016-10-25 | 0.90 |
| 10684 | 48177 | credit | 659 | 1 | 10684 | 2014-10-05 | 2017-01-16 | 1.52 |
| 10131 | 49068 | other | 10082 | 1 | 10131 | 2014-10-10 | 2016-05-25 | 0.63 |
| 11072 | 26326 | credit | 5275 | 0 | 11072 | 2014-11-11 | 2016-07-17 | 1.45 |

stats = loans.groupby('client\_id')['loan\_amount'].agg(['sum'])

stats.columns = ['total\_loan\_amount']

stats = clients.merge(stats, left\_on='client\_id',right\_index=True,how='left')

stats.head(10)

|  | **client\_id** | **joined** | **income** | **credit\_score** | **total\_loan\_amount** |
| --- | --- | --- | --- | --- | --- |
| 42320 | 42320 | 2000-04-27 | 229481 | 563 | 105931 |
| 39384 | 39384 | 2000-06-18 | 191204 | 617 | 149444 |
| 26945 | 26945 | 2000-11-26 | 214516 | 806 | 106889 |
| 41472 | 41472 | 2001-11-06 | 152214 | 638 | 120173 |
| 46180 | 46180 | 2001-11-06 | 43851 | 562 | 154017 |
| 46109 | 46109 | 2002-04-16 | 172677 | 527 | 179032 |
| 32885 | 32885 | 2002-05-13 | 58955 | 642 | 148806 |
| 29841 | 29841 | 2002-08-17 | 38354 | 523 | 176634 |
| 38537 | 38537 | 2002-10-21 | 127183 | 643 | 152768 |
| 35214 | 35214 | 2003-08-08 | 95849 | 696 | 129124 |

r\_client\_previous = ft.Relationship(es, 'clients','client\_id','loans','client\_id')

es = es.add\_relationship(relationship=r\_client\_previous)

r\_payments = ft.Relationship(es, 'loans','loan\_id','payments','loan\_id')

es = es.add\_relationship(relationship=r\_payments)

es

Entityset: clients

DataFrames:

clients [Rows: 25, Columns: 4]

loans [Rows: 443, Columns: 8]

payments [Rows: 3456, Columns: 5]

Relationships:

loans.client\_id -> clients.client\_id

payments.loan\_id -> loans.loan\_id

features, feature\_names = ft.dfs(entityset= es,

target\_dataframe\_name='clients',

agg\_primitives=['mean', 'max','percent\_true','last'],

trans\_primitives=['year', 'month', 'subtract\_numeric', 'divide\_numeric'])

pd.DataFrame(features['MONTH(joined)'].head())

|  | **MONTH(joined)** |
| --- | --- |
| **client\_id** |  |
| 42320 | 4 |
| 39384 | 6 |
| 26945 | 11 |
| 41472 | 11 |
| 46180 | 11 |

pd.DataFrame(features['MEAN(payments.payment\_amount)'].head())

|  | **MEAN(payments.payment\_amount)** |
| --- | --- |
| **client\_id** |  |
| 42320 | 1021.483333 |
| 39384 | 1193.630137 |
| 26945 | 1109.473214 |
| 41472 | 1129.076190 |
| 46180 | 1186.550336 |

pd.DataFrame(features[['MEAN(loans.loan\_amount)','MEAN(loans.rate)','MAX(loans.loan\_amount)','MAX(loans.rate)','LAST(loans.loan\_type)','LAST(loans.loan\_amount)']].head())

|  | **MEAN(loans.loan\_amount)** | **MEAN(loans.rate)** | **MAX(loans.loan\_amount)** | **MAX(loans.rate)** | **LAST(loans.loan\_type)** | **LAST(loans.loan\_amount)** |
| --- | --- | --- | --- | --- | --- | --- |
| **client\_id** |  |  |  |  |  |  |
| 42320 | 7062.066667 | 2.457333 | 13887.0 | 6.74 | home | 8090 |
| 39384 | 7865.473684 | 3.538421 | 14654.0 | 9.23 | other | 14654 |
| 26945 | 7125.933333 | 2.855333 | 14593.0 | 5.65 | cash | 9249 |
| 41472 | 7510.812500 | 3.981250 | 13657.0 | 9.82 | cash | 10122 |
| 46180 | 7700.850000 | 3.502500 | 14081.0 | 9.26 | other | 3834 |

Hw3

*import* pandas *as* pd

*import* numpy *as* anp

*import* sklearn

*from* sklearn *import* datasets

*from* sklearn.feature\_selection *import* SelectKBest, f\_regression

*import* matplotlib.pyplot *as* plt

*from* sklearn.ensemble *import* ExtraTreesRegressor

*import* seaborn *as* sns

dataset = datasets.fetch\_california\_housing(data\_home=None,

download\_if\_missing=True,

return\_X\_y=False,

as\_frame=True)

dataset

{'data': MedInc HouseAge AveRooms AveBedrms Population AveOccup Latitude \

0 8.3252 41.0 6.984127 1.023810 322.0 2.555556 37.88

1 8.3014 21.0 6.238137 0.971880 2401.0 2.109842 37.86

2 7.2574 52.0 8.288136 1.073446 496.0 2.802260 37.85

3 5.6431 52.0 5.817352 1.073059 558.0 2.547945 37.85

4 3.8462 52.0 6.281853 1.081081 565.0 2.181467 37.85

... ... ... ... ... ... ... ...

20635 1.5603 25.0 5.045455 1.133333 845.0 2.560606 39.48

20636 2.5568 18.0 6.114035 1.315789 356.0 3.122807 39.49

20637 1.7000 17.0 5.205543 1.120092 1007.0 2.325635 39.43

20638 1.8672 18.0 5.329513 1.171920 741.0 2.123209 39.43

20639 2.3886 16.0 5.254717 1.162264 1387.0 2.616981 39.37

Longitude

0 -122.23

1 -122.22

2 -122.24

3 -122.25

4 -122.25

... ...

20635 -121.09

20636 -121.21

20637 -121.22

20638 -121.32

20639 -121.24

[20640 rows x 8 columns],

'target': 0 4.526

1 3.585

2 3.521

3 3.413

4 3.422

...

20635 0.781

20636 0.771

20637 0.923

20638 0.847

20639 0.894

Name: MedHouseVal, Length: 20640, dtype: float64,

'frame': MedInc HouseAge AveRooms AveBedrms Population AveOccup Latitude \

0 8.3252 41.0 6.984127 1.023810 322.0 2.555556 37.88

1 8.3014 21.0 6.238137 0.971880 2401.0 2.109842 37.86

2 7.2574 52.0 8.288136 1.073446 496.0 2.802260 37.85

3 5.6431 52.0 5.817352 1.073059 558.0 2.547945 37.85

4 3.8462 52.0 6.281853 1.081081 565.0 2.181467 37.85

... ... ... ... ... ... ... ...

20635 1.5603 25.0 5.045455 1.133333 845.0 2.560606 39.48

20636 2.5568 18.0 6.114035 1.315789 356.0 3.122807 39.49

20637 1.7000 17.0 5.205543 1.120092 1007.0 2.325635 39.43

20638 1.8672 18.0 5.329513 1.171920 741.0 2.123209 39.43

20639 2.3886 16.0 5.254717 1.162264 1387.0 2.616981 39.37

Longitude MedHouseVal

0 -122.23 4.526

1 -122.22 3.585

2 -122.24 3.521

3 -122.25 3.413

4 -122.25 3.422

... ... ...

20635 -121.09 0.781

20636 -121.21 0.771

20637 -121.22 0.923

20638 -121.32 0.847

20639 -121.24 0.894

[20640 rows x 9 columns],

'target\_names': ['MedHouseVal'],

'feature\_names': ['MedInc',

'HouseAge',

'AveRooms',

'AveBedrms',

'Population',

'AveOccup',

'Latitude',

'Longitude'],

'DESCR': '.. \_california\_housing\_dataset:\n\nCalifornia Housing dataset\n--------------------------\n\n\*\*Data Set Characteristics:\*\*\n\n:Number of Instances: 20640\n\n:Number of Attributes: 8 numeric, predictive attributes and the target\n\n:Attribute Information:\n - MedInc median income in block group\n - HouseAge median house age in block group\n - AveRooms average number of rooms per household\n - AveBedrms average number of bedrooms per household\n - Population block group population\n - AveOccup average number of household members\n - Latitude block group latitude\n - Longitude block group longitude\n\n:Missing Attribute Values: None\n\nThis dataset was obtained from the StatLib repository.\nhttps://www.dcc.fc.up.pt/~ltorgo/Regression/cal\_housing.html\n\nThe target variable is the median house value for California districts,\nexpressed in hundreds of thousands of dollars ($100,000).\n\nThis dataset was derived from the 1990 U.S. census, using one row per census\nblock group. A block group is the smallest geographical unit for which the U.S.\nCensus Bureau publishes sample data (a block group typically has a population\nof 600 to 3,000 people).\n\nA household is a group of people residing within a home. Since the average\nnumber of rooms and bedrooms in this dataset are provided per household, these\ncolumns may take surprisingly large values for block groups with few households\nand many empty houses, such as vacation resorts.\n\nIt can be downloaded/loaded using the\n:func:`sklearn.datasets.fetch\_california\_housing` function.\n\n.. topic:: References\n\n - Pace, R. Kelley and Ronald Barry, Sparse Spatial Autoregressions,\n Statistics and Probability Letters, 33 (1997) 291-297\n'}

X = dataset['data']

y = dataset['target']

X.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 20640 entries, 0 to 20639

Data columns (total 8 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 MedInc 20640 non-null float64

1 HouseAge 20640 non-null float64

2 AveRooms 20640 non-null float64

3 AveBedrms 20640 non-null float64

4 Population 20640 non-null float64

5 AveOccup 20640 non-null float64

6 Latitude 20640 non-null float64

7 Longitude 20640 non-null float64

dtypes: float64(8)

memory usage: 1.3 MB

best\_features = SelectKBest(score\_func=f\_regression, k=8)

fit = best\_features.fit(X,y)

feature\_scores = pd.concat([pd.DataFrame(X.columns),pd.DataFrame(fit.scores\_)],axis=1)

feature\_scores.columns = ['Specs','Score']

feature\_scores.nlargest(8,'Score')

|  | **Specs** | **Score** |
| --- | --- | --- |
| 0 | MedInc | 18556.571631 |
| 2 | AveRooms | 487.757462 |
| 6 | Latitude | 438.005453 |
| 1 | HouseAge | 232.841479 |
| 3 | AveBedrms | 45.108576 |
| 7 | Longitude | 43.698976 |
| 4 | Population | 12.547410 |
| 5 | AveOccup | 11.635342 |

model = ExtraTreesRegressor()

model.fit(X,y)

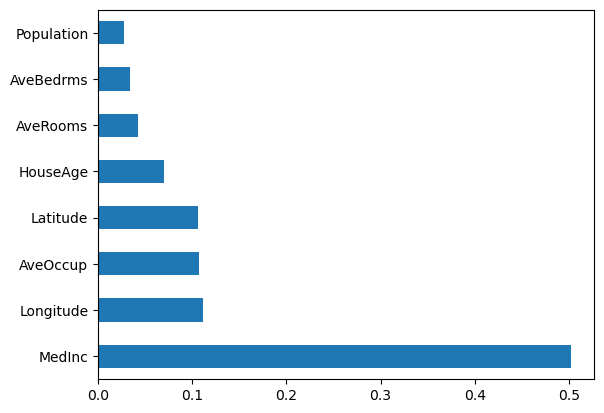
model.feature\_importances\_

array([0.50186861, 0.07035609, 0.04174722, 0.03423284, 0.02754407,

0.10723049, 0.10572781, 0.11129287])

feat\_importances = pd.Series(model.feature\_importances\_, index=X.columns)

feat\_importances.nlargest(8).plot(kind='barh')



data = pd.concat([X,y],axis=1)

corrmat = data.corr()

top\_corr\_features = corrmat.index

plt.figure(figsize=(20,20))

g=sns.heatmap(data[top\_corr\_features].corr(),annot=True,cmap="RdYlGn")

A screenshot of a graph

Description automatically generated