# Predicting user features from Social Media data

Presented by User 08: The Maze


François Mercier 20167322
Nicolas Sauthier 932337
Zicong Mo 20141760
Andrew Kristensen 20119706
Yifan(Andy) Bai 20153885


# What is the problem?

# Problem

We have user data from various social media accounts, and we'd like to be able to predict:

- Age
- Gender
- 5 personality traits

We have access to :

- Oxford facial features
- LIWC and NRC features from text posts
- Relation data from liked pages

# How can we use this data to achieve the goal?

## Page Relations

Drew

## Age and Gender

Nicolas and Andy

## Personality Predictions

Zicong and François

# Early Approach: Page Relations

"Similar people like similar things"

Our earliest idea for estimating user values was utilizing the relations data

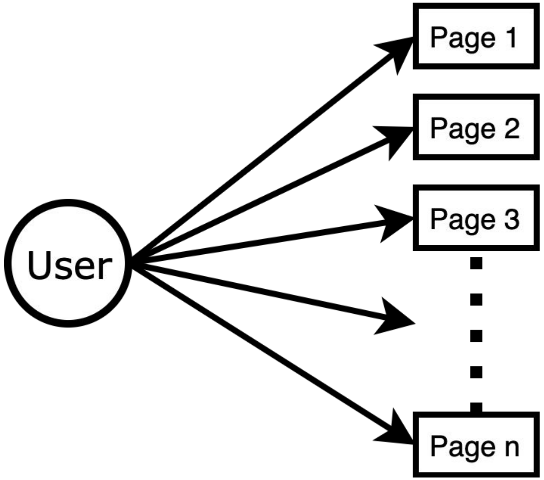
Graph structure might be exploitable

Easy and fast to get working for comparisons against the baselines

# Approaches: Page Relations

General idea: If we know which kinds of users have liked which pages, we can predict a new user's values based on which pages they have liked

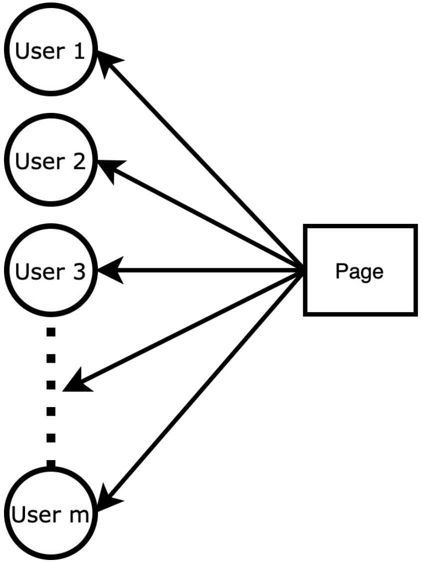
For each user, we hold onto their id and all the pages that they've liked



# Approaches: Page Relations

General idea: If we know which kinds of users have liked which pages, we can predict a new user's values based on which pages they have liked

For each page, we hold onto the pageid and all the users who like them

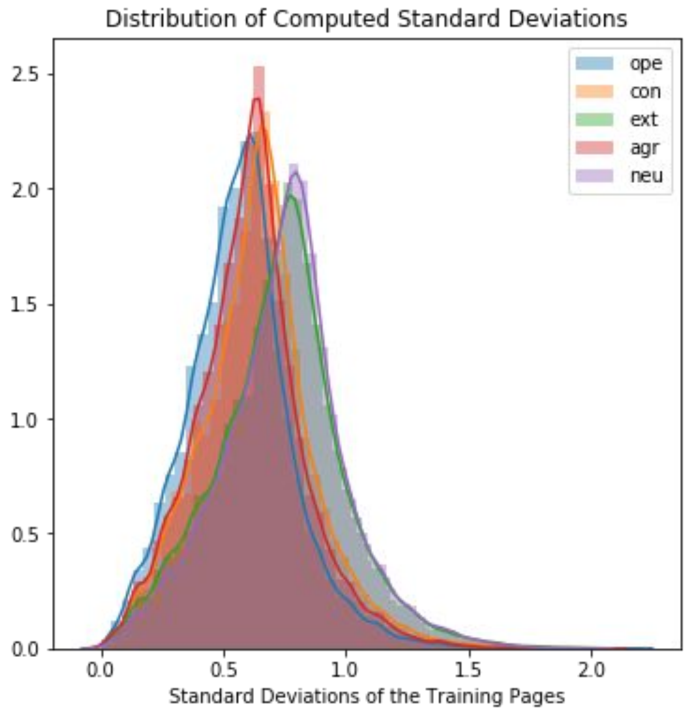


# Page Relations - Regression

We can predict the values for the personality traits (ope, con, etc) by taking a weighted average from all the pages the user likes

Why a weighted average?

The more we trust an estimate, the more credit we should give to it




| Trait | OPE | CON | EXT | AGR | NEU |
| --- | --- | --- | --- | --- | --- |
| Mean | 0.5683 | 0.6386 | 0.7306 | 0.6093 | 0.7469 |

# Page Relations - Regression

How did this go?

| Trait | ope | con | ext | agr | neu |
|---:|---:|---:|---:|---:|---:|
| Model RMSE | 0.613 | 0.7086 | 0.7973 | 0.6565 | 0.7896 |
| Baseline | 0.652 | 0.798 | 0.788 | 0.665 | 0.734 |
| Difference | -0.039 | -0.0894 | 0.0093 | -0.0085 | 0.0556 |

Ok, but not great

# Page Relations - Classification

We approached the problem of classification for the ages and genders in a binned regression manner

Just like the personalities, we would take a weighted mean

Bin the output value given some thresholds (.5 for the gender, and age limits for the ages)

# Page Relations - Classification

We approached the problem of classification for the ages and genders in a binned regression manner

Just like the personalities, we would take a weighted mean

Bin the output value given some thresholds (.5 for the gender, and age limits for the ages)

....This didn’t go great

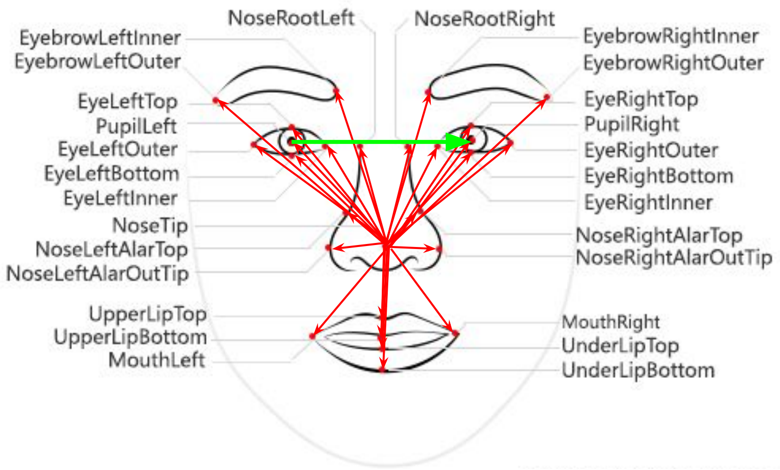
# Page Relations - Where is it?

The approaches that we took in the next parts of this presentation out performed it, so it has gone the way of the dodo

We knew this would be the case, since we needed something to give ourselves a slightly improved baseline that we could then beat again.

Since we never implemented any of the Node2Vec work, the model lacked sophistication

# Gender Prediction



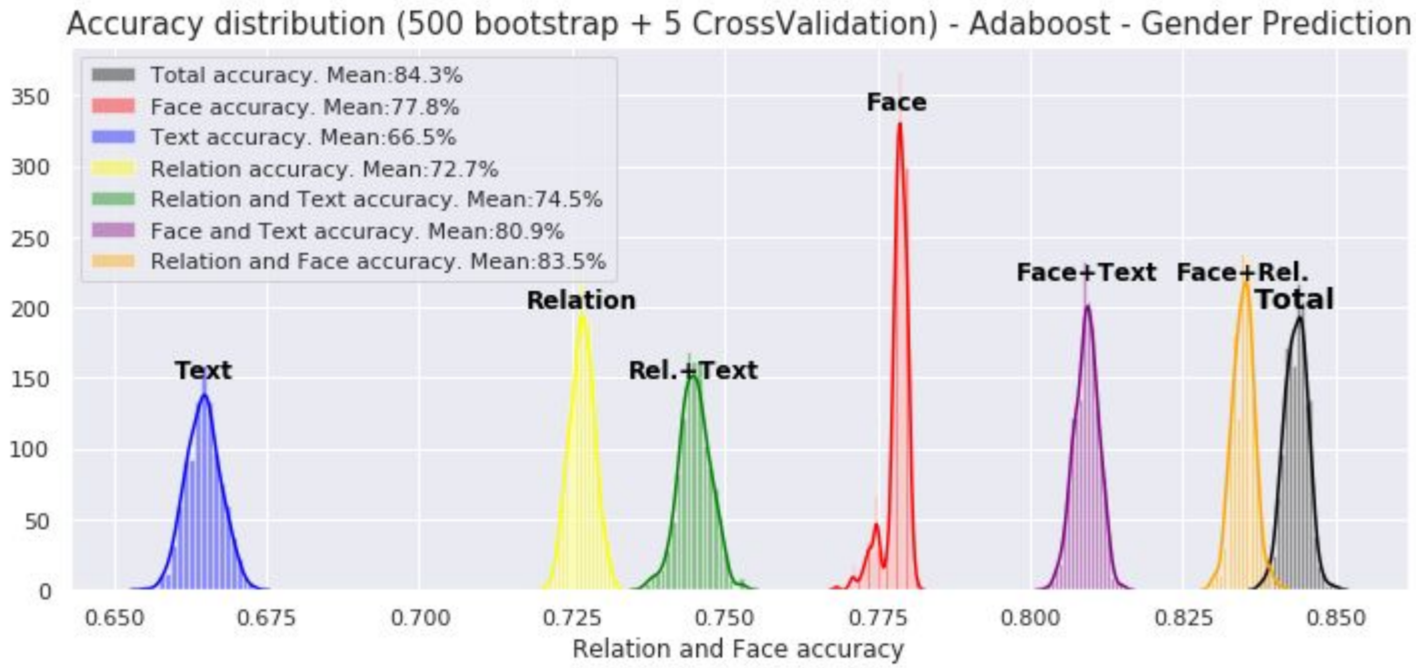

Data source: all three

1. Face data:
   a. Most facing or mean imputation
   b. Correction for yaw and pitch for "front face"
   c. Distance from nose, resized for eye_dist = 1
   d. Removing of very highly correlated (>0.99)
   **TOTAL 25 features**
2. NRC/LIWC data
   a. Removing of very highly correlated (>0.99)
   **TOTAL 88 features**
3. Relational data
   a. Co-occurrence matrix (same userID for two likeids), first 10'000 likeid
   b. Reduced with SVD to 15 dimensions.
   c. Average of all user's likeid. Imputations with 0
   **TOTAL 15 features**

Algorithm: Ada Boost

- Grid search for hyperparameters

# Gender Prediction - Results



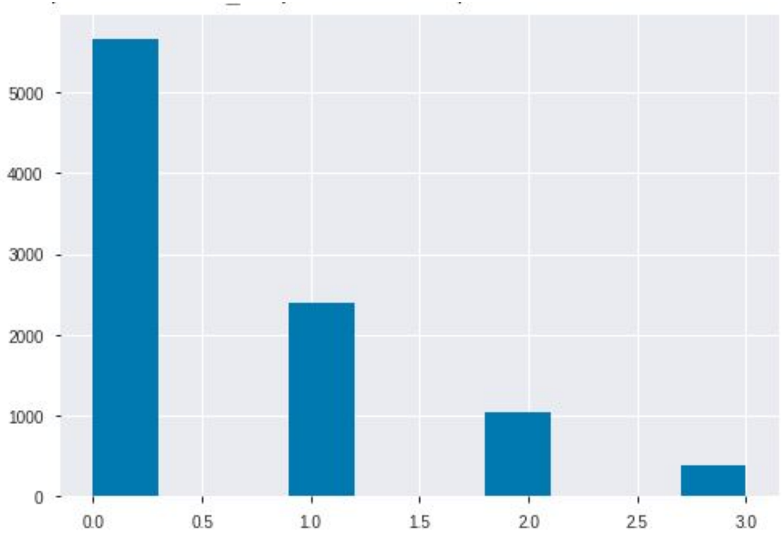
# Age Prediction

- Naive Bayes' wasn't successful with facial data only
- Tried ensemble method
    - With text data (nrc + liwc after feature selection):
        - No resampling: Gradient Boosting
        - Resampling: Random Forest/Extra Tree
    - With facial + text data :
        - AdaBoost

Photo Credit: https://www.pinterest.ca/pin/463800461600069721/?lp=true

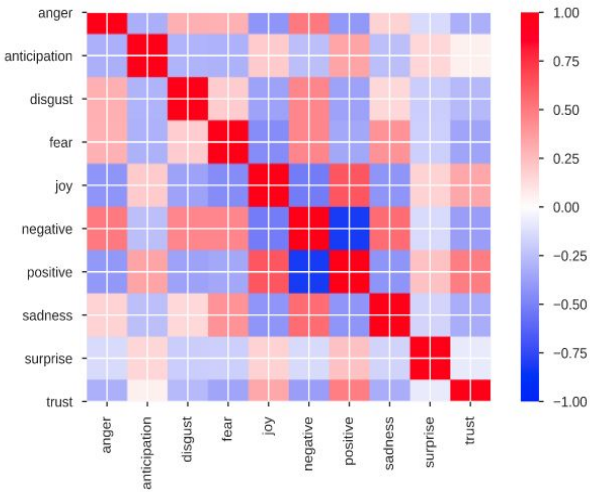
# Training Data

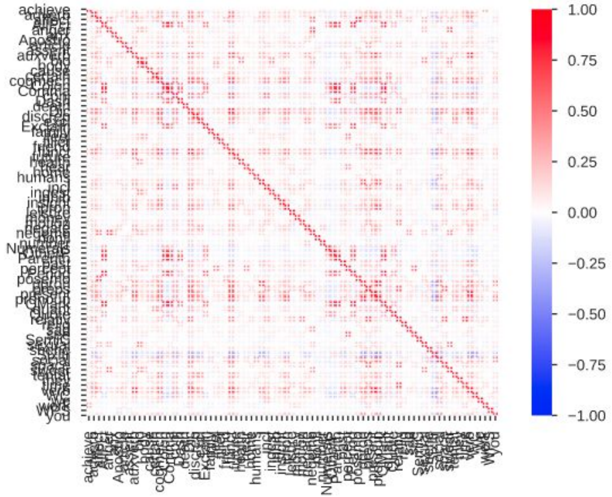
- Encode based on age groups: `Xx-24: 0, 25-34: 1, 35-49: 2, 50+: 3`
- Data highly imbalanced



# Feature Engineering with Text Data

- Merge NRC and LIWC by user ID after analyzing features, rejecting unuseful ones (e.g. highly correlated/skewed)



# Feature Engineering with Text Data

- Both have many features with high number of zeros
- Rejected variables (LIWC):

```
'Comma': Highly correlated with ‘AllPct’ - 0.9408211702

‘Funct’: Highly correlated with ‘Dic’ - 0.9371835644

‘QMark’: Highly correlated with ‘Comma’ - 0.9437018647
```

# Gradient Boost

- Baseline setup results (random state = 42), 5-fold CV: 0.611
- Random Search: 0.615
- Some over-fitting exist, public test accuracy: 0.612 - 0.616

# Resampling Efforts

- Training data highly imbalanced, consider resampling
- Extra Trees and Random Forest
- RandomOverSampler:
  - 5-fold CV, train/test split = 0.8/0.2
    - Random Forest = 0.942
    - Extra Trees = 0.961
- But failed miserably with public test (<0.6), due to overfitting in training

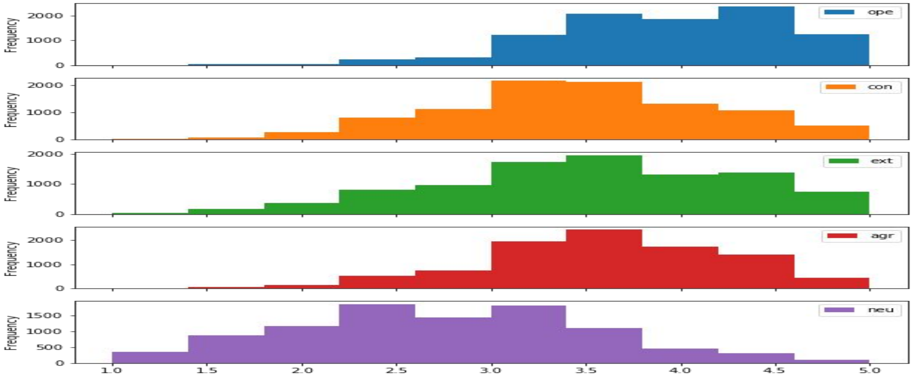
# AdaBoost

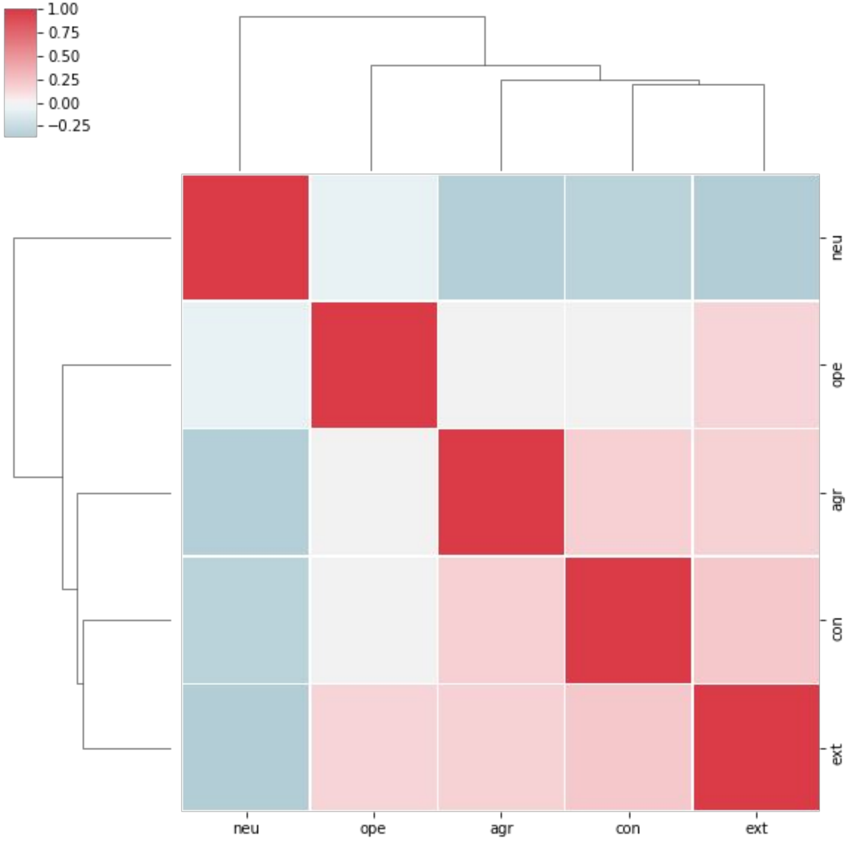
- Adopted Nicolas' feature set
- Grid Search: acc = 0.657 with public test

# Personality Predictions

Exploratory Data Analysis

- No orphans
- No missing data
- Skewed distribution for labels **and correlated**

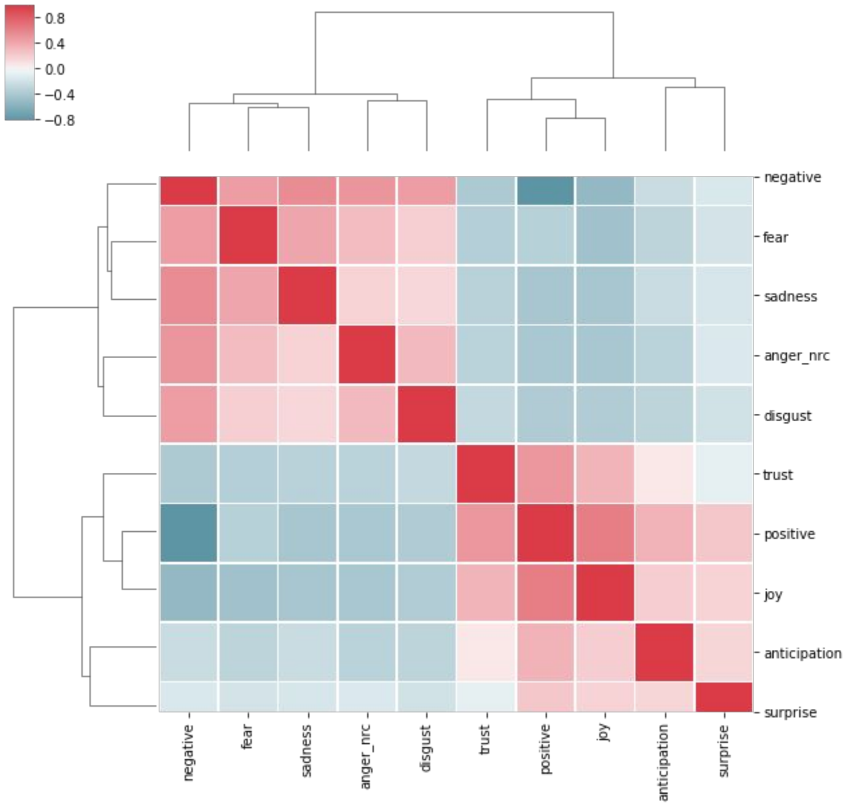


# Personality Predictions

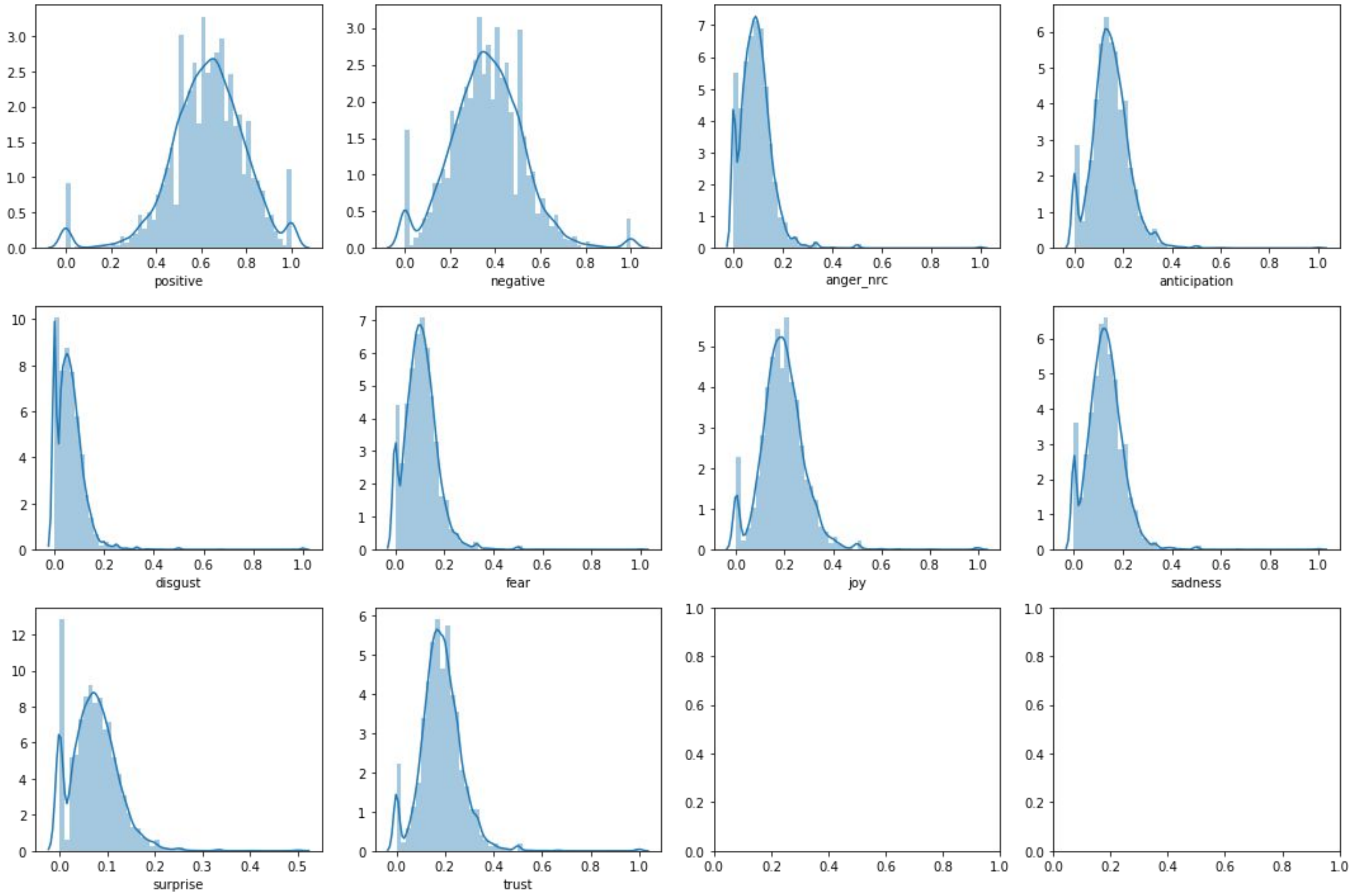
Exploratory Data Analysis

- No orphans
- No missing data
- Skewed distribution for labels **and correlated**
- NRCC features
    - Two group: Positive vs negative ones
    - Strong correlation
    - **Presence of outliers (missing data?)**
-



0.8
0.4
0.0
−0.4
−0.8
negative
fear
sadness
anger_nrc
disgust
trust
positive
joy
anticipation
surprise

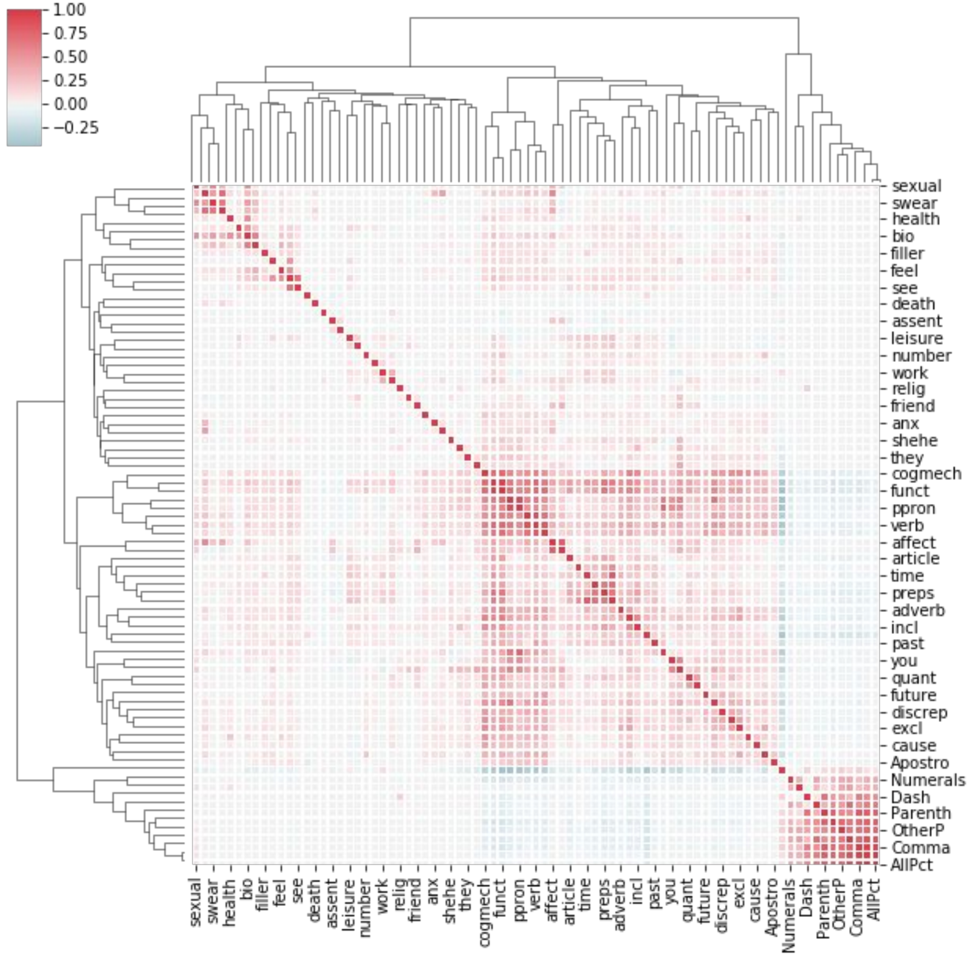
# NRCC features distributions



# Personality Predictions

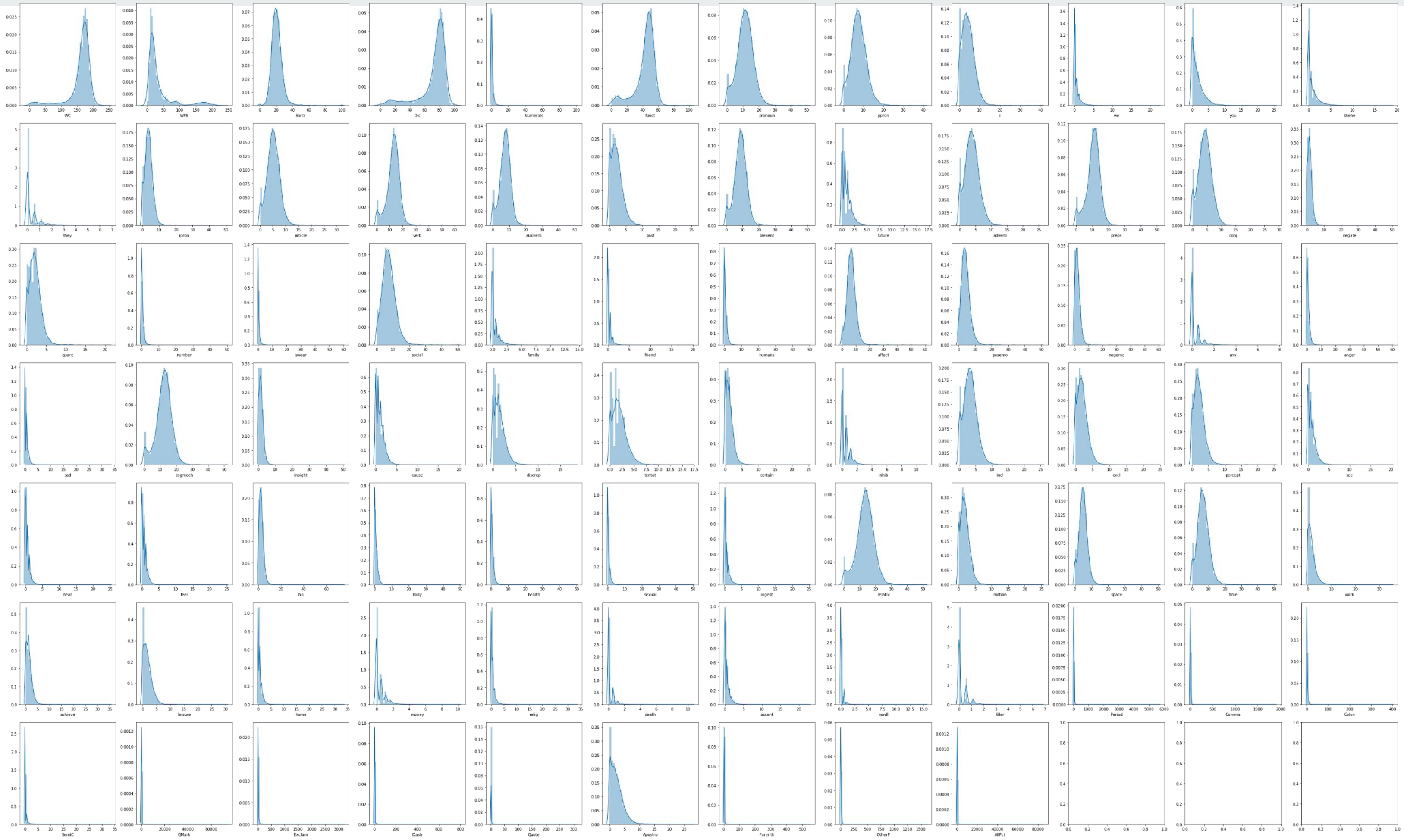
Exploratory Data Analysis

- No orphans
- No missing data
- Skewed distribution for labels **and correlated**
- NRCC features
  - Two group: Positive vs negative ones
  - Strong correlation
  - **Presence of outliers (missing data?)**
- LIWC features
  - Groups of features with strong correlations
  - **Presence of outliers (missing data?)**
  -



1.00
0.75
0.50
0.25
0.00
−0.25
sexual
swear
health
bio
filler
feel
see
death
assent
leisure
number
work
relig
friend
anx
shehe
they
cogmech
funct
ppron
verb
affect
article
time
preps
adverb
incl
past
you
quant
future
discrep
excl
cause
Apostro
Numerals
Dash
Parenth
OtherP
Comma
AllPct

# LIWC features distributions



# Personality Predictions

Exploratory Data Analysis

**⇒ With skewed labels, focus on decision tree family model and ensemble models using it**

**⇒ Focus on removing features to simplify models**

**⇒ Presence of outliers (replacing by median, mean)**

# How our models worked out

# Personality Predictions

| Model | Training errors (80%) | Validation errors (20%) | Comments |
|---|---|---|---|
| Mean baseline | Open RMSE 0.62775<br>Neurotic RMSE 0.7904<br>Extrovert RMSE 0.8077<br>Agreeable RMSE 0.6605<br>Conscientious RMSE 0.7169 | Open RMSE 0.6474<br>Neurotic RMSE 0.8045<br>Extrovert RMSE 0.8146<br>Agreeable RMSE 0.6651<br>Conscientious RMSE 0.7299 | |
| Decision tree | Open RMSE 0.6299<br>Neurotic RMSE 0.7819<br>Extrovert RMSE 0.799<br>Agreeable RMSE 0.6487<br>Conscientious RMSE 0.7078 | Open RMSE 0.6203<br>Neurotic RMSE 0.8001<br>Extrovert RMSE 0.8170<br>Agreeable RMSE 0.6688<br>Conscientious RMSE 0.7119 | 91 features<br>Overfitting |
| Decision tree with L1 features selection | Open RMSE 0.6284<br>Neurotic RMSE 0.7917<br>Extrovert RMSE 0.8030<br>Agreeable RMSE 0.6618<br>Conscientious RMSE 0.7161 | Open RMSE 0.6175<br>Neurotic RMSE 0.7947<br>Extrovert RMSE 0.8053<br>Agreeable RMSE 0.6417<br>Conscientious RMSE 0.7143 | 6 features only<br>No overfitting<br>Better performance than with all features<br>**Not enough to beat baselines** |

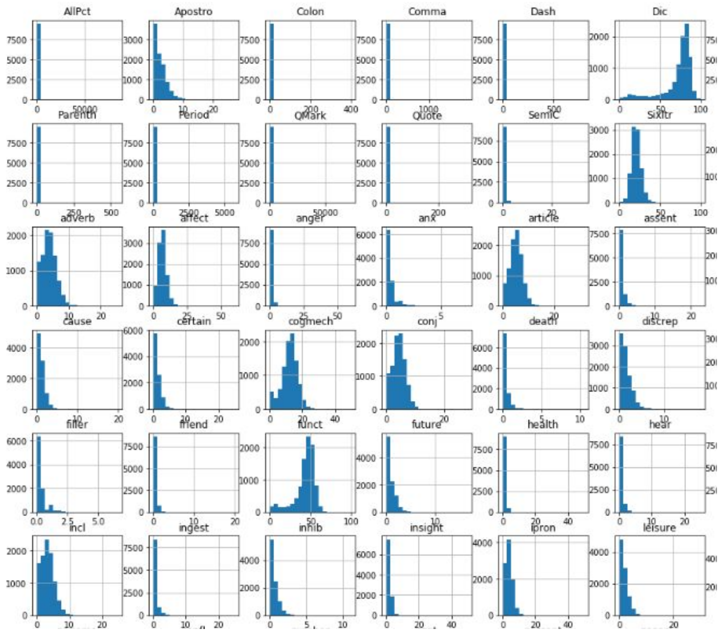
# Personality Predictions

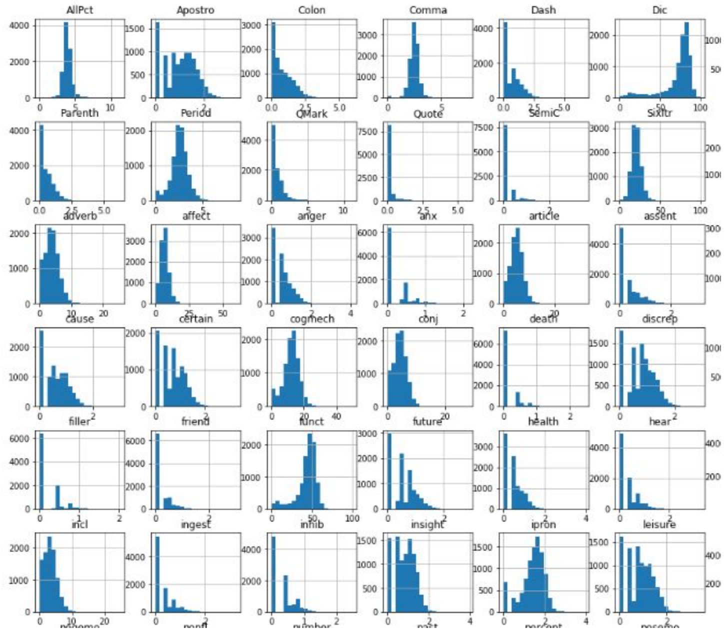
| Model | Training errors (80%) | Validation errors (20%) | Comments |
|---|---|---|---|
| Random Forest | Open RMSE 0.6193<br>Neurotic RMSE 0.7642<br>Extrovert RMSE 0.7794<br>Agreeable RMSE 0.6404<br>Conscientious RMSE 0.6917 | Open RMSE 0.6200<br>Neurotic RMSE 0.7966<br>Extrovert RMSE 0.8081<br>Agreeable RMSE 0.6543<br>Conscientious RMSE 0.7065 | All features (as sampling features doesn't help)<br>OOB R^2 score: 2.3% (very poor) |
| Gradient Boosting | Open RMSE 0.6284<br>Neurotic RMSE 0.7917<br>Extrovert RMSE 0.8030<br>Agreeable RMSE 0.6618<br>Conscientious RMSE 0.7161 | Open RMSE 0.6175<br>Neurotic RMSE 0.7947<br>Extrovert RMSE 0.8053<br>Agreeable RMSE 0.6417<br>Conscientious RMSE 0.7143 | All features (as sampling features doesn't help)<br>No specific management for outliers<br>1 model per score<br>Using some scores to predict others doesn't help (fusion approach)<br>Manual hyperparameters selection<br>Best model on valid set<br>**Bootstrap RMSE to get confidence interval on RMSE (30 tries)**<br>**Beat the baselines slightly** |

# Personality Predictions

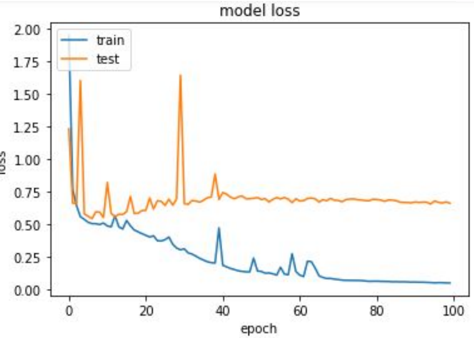
| Gradient Boosting model confidence interval Using 80/20 split | Error means On valid set | Low (2 * std dev) | High (2 * std dev) | Z test 1 tailed test | p-value for beating the baseline |
|---|---|---|---|---|---|
| **opn_rmse** | 0.620502 | 0.597431 | 0.643572 | -14.955 | **0.00%** |
| **neu_rmse** | 0.792591 | 0.775389 | 0.809794 | -3.444 | **0.03%** |
| **ext_rmse** | 0.810185 | 0.790275 | 0.830096 | 12.206 | **100%** |
| **agr_rmse** | 0.662342 | 0.639646 | 0.685039 | -1.282 | **9.98%** |
| **con_rmse** | 0.706747 | 0.684471 | 0.729023 | -13.401 | **0.00%** |

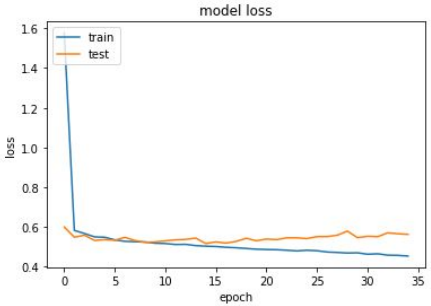
# Deep Learning

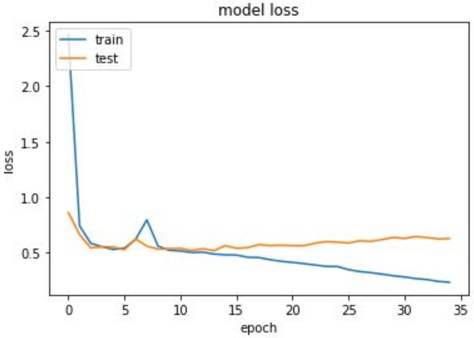


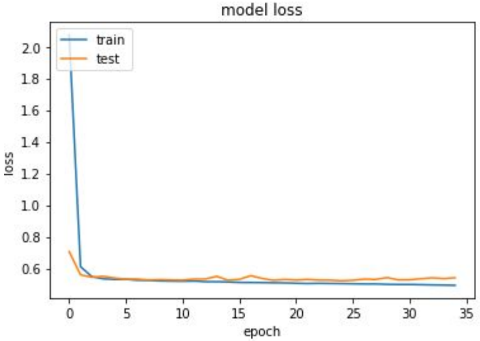



```
Train on 7600 samples, validate on 1900 samples
Epoch 1/35
7600/7600 [==============================] - 3s 458us/step - loss: 1.6652 - val_loss: 0.8075
Epoch 2/35
7600/7600 [==============================] - 1s 154us/step - loss: 0.7577 - val_loss: 0.6665
```




```
model = Sequential()
model.add(Dense(364, input_dim=91, activation="relu"))
model.add(Dropout(0.2))
model.add(Dense(1000, activation="relu"))
model.add(Dense(1000, activation="relu"))
model.add(Dense(500, activation="relu"))
model.add(Dense(250, activation="relu"))
model.add(Dense(100, activation="relu"))
model.add(Dense(20, activation="relu"))
model.add(Dense(5 ,activation='linear'))
model.compile(loss="mean_squared_error", optimizer="Nadam")
```

```
model = Sequential()
model.add(Dense(40, in
model.add(Dropout(0.2)
model.add(Dense(300, a
model.add(Dense(300, a
model.add(Dense(300, a
model.add(Dense(100, a
model.add(Dense(5 ,act
model.compile(loss="me
self._model = model
```

```
model = Sequential()
model.add(Dense(500, i
model.add(Dropout(0.2)
model.add(Dense(1000,
model.add(Dense(1000,
model.add(Dense(500, a
model.add(Dense(205, a
model.add(Dense(125, a
model.add(Dense(50, ac
model.add(Dense(25, ac
model.add(Dense(25, ac
model.add(Dense(5 ,act
model.compile(loss="me
self._model = model
```

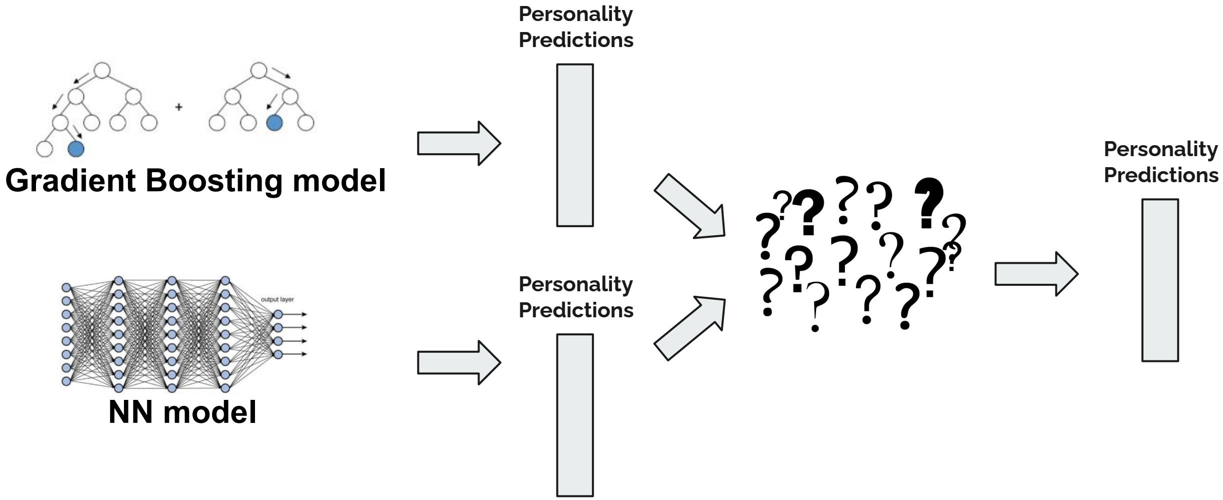
```
model = Sequential()
model.add(Dense(20, in
model.add(Dropout(0.2)
model.add(Dense(40, ac
model.add(Dense(40, ac
model.add(Dense(40, ac
model.add(Dense(10, ac
model.add(Dense(5 ,act
model.compile(loss="me
```

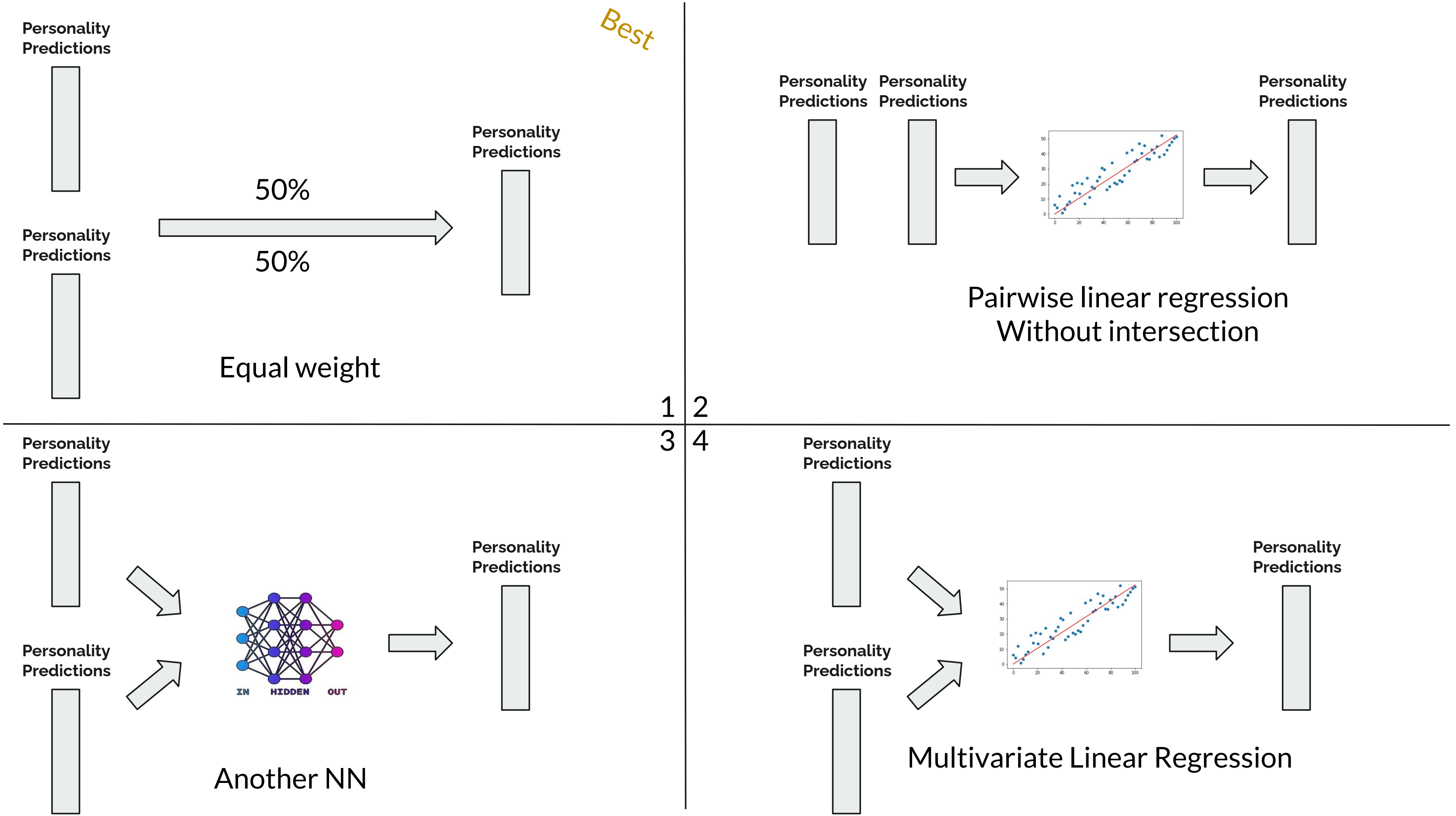
# Personality Predictions

```
model = Sequential()
model.add(Dense(200, input_dim=91, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(200, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(100, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(50, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(20, activation="linear"))
model.add(Dropout(0.1))
model.add(Dense(5 ,activation='linear'))
model.compile(loss="mean_squared_error", optimizer="Nadam")
```

| NN model confidence interval<br>Using 80/20 split | Error means<br><br>On valid set | Low<br><br>(2 * std dev) | High<br><br>(2 * std dev) | Z test<br><br>1 tailed test | p-value for beating the baseline |
|---|---|---|---|---|---|
| **opn_rmse** | 0,6198 | 0.6053 | 0.6428 | -15.40 | 0% |
| **neu_rmse** | 0,7907 | 0.7726 | 0.8231 | -4.909 | 0.029% |
| **ext_rmse** | 0,8041 | 0.7819 | 0.8240 | 7.256 | 100% |
| **agr_rmse** | 0,6578 | 0.6360 | 0.6721 | -3.853 | 0.006% |
| **con_rmse** | 0,7068 | 0.6839 | 0.7225 | -20.106 | 0% |

# Ensemble Model





Best
Personality Predictions
Personality Predictions
50%
50%
Personality Predictions
Equal weight
1
2
Personality Predictions
Personality Predictions
Personality Predictions
Pairwise linear regression
Without intersection
3
4
Personality Predictions
Personality Predictions
IN
HIDDEN
OUT
Personality Predictions
Another NN
Personality Predictions
Personality Predictions
Personality Predictions
Multivariate Linear Regression

# Personality Predictions

| Ensemble Model confidence interval Using 80/20 split | Error means On valid set | Low (2 * std dev) | High (2 * std dev) | Z test 1 tailed test | p-value for beating the baseline |
|---|---|---|---|---|---|
| **opn_rmse** | 0.62069 ↑ | 0.599249 ↓ | 0.642132 ↓ | -15.99 ↓ | 0% |
| **neu_rmse** | 0.789116 ↓ | 0.770993 ↓ | 0.807239 ↓ | -5.369 ↓ | 0% |
| **ext_rmse** | 0.798725 ↓ | 0.774492 ↓ | 0.822958 ↓ | 4.8482 ↓ | 100% |
| **agr_rmse** | 0.655121 ↓ | 0.637848 ↑ | 0.672393 ↑ | -6.265 ↓ | 0% |
| **con_rmse** | 0.703591 ↓ | 0.682387 ↓ | 0.724796 ↓ | -15.70 ↓ | 0% |