CSC311 - ML GROUP PROJECT

Aiden Lei Wang, Burak Unat, Craig William Katsube, David Lee

April 1, 2025

Data

Exploration

Based on the survey data, our target classes are three distinct food categories: pizza, shawarma, and sushi. The dataset consists of structured responses about food perceptions collected through an 8-question survey.

Here are the results from the answers to the survey; the dataset contains both numeric and text features:

Numeric Features:

- **Q1_complexity:** Complexity rating on a 1-5 scale.
  - Mainly has left-tail distributions that peak at the 5 complexity, except for pizza that not many participants voted 5 for. This data can be helpful as it would suggest that the item is likely not pizza if given a 5 rating.
- **Q2_ingredients:** Expected number of ingredients.
  - The types are normally distributed around a mean of approximately 5 for pizza and sushi and 7.5 for shawarma. Although pizza and sushi also have a larger variance, shawarma has a more dense probability distribution around the 10 complexity range, which could suggest that the food item is shawarma if near 10.
- **Q3_setting_:** Binary indicators for various eating settings (weekday lunch/dinner, weekend lunch/dinner, party, late night).
  - The distribution is fairly uniform in pizza, significantly less expected at parties and more at lunch for shawarma, and much more expected during dinner and less as a snack for sushi. There seems to be some indication that pizza is the most expected food item at a party, sushi may be more expected at dinner though it could be pizza as well, and shawarma is more likely as lunch though could still be the other two as well.
- **Q4_price:** Expected price per serving.
  - Pizza seems to be perceived to be less expensive than shawarma, which is less expensive than sushi. The expected price for pizza is around 7, 10 for shawarma, and 12 for sushi. Pizza seems to have a larger variance than shawarma, however, so on average, participants expected pizza to be cheaper, but many participants also thought it could be more expensive (most notably pizza priced at $20). Shawarma has a much smaller variance, and as such, it seems more likely that the lower-priced items may be pizza and the more expensive items, especially much more expensive items past $20, may be sushi.

- **Q7_has_:** Binary indicators of whom the food reminds respondents of (parents, siblings, friends, teachers, strangers).
    - Although friends are the most common setting to eat any of these items with, pizza was the only type that participants seemed to imagine eating with teachers and shawarma with strangers. Participants also seemed to perceive eating sushi as a more family endeavor than the other food types though participants also did seem to eat pizza and shawarma with family as well.
- **Q8_hot_sauce_level:** Numerical representation of hot sauce preference (0-3).
    - Participants seemed strongly inclined against adding hot sauce to sushi. Though people also did not tend to add hot sauce to pizza, around half the participants would add some level of hot sauce. Shawarma has the flipped distribution of pizza, where most participants would add hotter sauce to their shawarma and would not particularly expect non-spicy shawarma.

Text Features:

- **Q5_movie:** Movie associated with the food (converted to lowercase, with blank values replaced by "none")
    - Although there were many “none” responses, some movies appeared much more frequently for each food type. The top mentioned movies for pizza were *Teenage Mutant Ninja Turtles*, *Home Alone*, *Spiderman*, *Ratatouille*, and *Cloud with a Chance of Meatballs*; for Shawarma was primarily *Avengers*. Sushi did not have a clear frequently mentioned movie, and “none” was most frequently mentioned, but after that is *Jiro Dreams of Sushi* followed by *Spirited Away*, *Monster Inc*, *Your Name*, *Finding Nemo*, and *Kill Bill*. The movies are quite diverse and any input that contains these would have a strong probability of being the respective category. However, it’s difficult to use the tail-end categories.
- **Q6_drink:** Drink paired with the food (converted to lowercase)
    - The most commonly mentioned drink for pizza is Coke, followed far behind by generic soda and pop. For shawarma, both Coke and water were popular choices. For sushi, water and tea were the top choices. Tea seems to be indicative of sushi. Coke could be pizza or shawarma, potentially more likely to be pizza, and water could be sushi or shawarma with somewhat equal probability but much less likely to just be pizza.

Distribution graphs are included at the end of the report

Feature Selection

We included all transformed features from the survey questions in our models based on both domain knowledge and empirical analysis:

- Numeric features (Q1_complexity, Q2_ingredients, Q4_price, Q8_hot_sauce_level): Provided quantitative measures that showed clear discriminative patterns.
- Binary indicators (Q3_setting_, Q7_has_): Captured categorical preferences that helped distinguish between food types.
- Text features (Q5_movie, Q6_drink): Contained semantic associations that provided complementary signals.

We deliberately maintained the full feature set rather than performing feature elimination, as our analysis indicated that each feature contributed unique information to the classification task. The price feature (Q4) proved especially valuable, showing strong correlations with specific food classes.

Data Representation

Our data representation strategy involved careful preprocessing of the survey data to create a consistent and informative feature set for our models. Based on our analysis of the cleandata.py file, we applied the following transformations:

Numeric Features

We applied standardized processing to all numeric features:

- **Q1_complexity:** Converted responses directly to a 1-5 numerical scale using pd.to_numeric() with error handling.
- **Q2_ingredients:** Applied sophisticated text parsing to extract numeric values, handling ranges (e.g., "5-10") by taking their mean and processing expressions like "at least 7" appropriately.
- **Q4_price:** Besides extracting numerical values and handling ranges, we implemented domain-specific coding where mentions of "slice" were coded as -1 (implying pizza), "sushi" as -2, and "piece" as -3 (either pizza or sushi).
- **Q8_hot_sauce_level:** Mapped text descriptions to a 0-3 ordinal scale (none=0, mild=1, medium=2, hot=3).

Text Features

We implemented extensive text standardization processes:

- **Q5_movie:** We standardized movie mentions (e.g., all variants of "Ninja Turtles" mapped to "ninja turtles"), converted all to lowercase, and replaced missing values with "none". We also removed punctuation for consistency.
- **Q6_drink:** Applied comprehensive drink standardization through pattern matching, with 25+ drink categories identified through string matching (e.g., variants of cola → "coke", carbonated beverages → "pop").

Binary Features

We created multiple binary indicator variables:

- **Q3_setting_:** Generated six binary features from the settings question (weekday lunch/dinner, weekend lunch/dinner, party, late night).
- **Q7_has_:** Created five binary features indicating which relationships (parents, siblings, friends, teachers, strangers) were associated with the food.

Handling Missing Values

We implemented a consistent strategy for missing values:

- All numeric fields (Q1, Q2, Q4, Q8) had missing values replaced with -99.
- Text fields had missing values replaced with "none".
- Binary fields had missing values defaulted to 0 through the na=False parameter in string matching.

This preprocessing pipeline created a standardized feature set where each survey response was represented by 17 structured features (4 numeric, 2 text, and 11 binary).

Data Splitting

We employed a consistent train-test splitting strategy across all model implementations:

```
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y)
```

By using a fixed 80/20(80 training, 20 test) split with stratification, we ensured:

- Sufficient training data for robust model learning.
- Representative test data for reliable performance evaluation.
- Consistent class distributions in both splits.
- Reproducibility through a fixed random seed 42.

This approach allowed for fair comparisons between different modeling techniques.

Model

For this project, three models stood out as excellent candidates for evaluating the survey data: **softmax regression, multilayer perceptrons (MLPs), and random forests**. Each of these approaches offers a unique perspective and set of strengths when it comes to integrating both numeric responses and text-based inputs.

Softmax regression is highly attractive for multiclass classification because it offers clear interpretability through its coefficients, which reveal how each feature influences the class probabilities. After preprocessing the data by standardizing the numeric features and vectorizing the text features, we fed the transformed data into a Logistic Regression classifier (effectively a softmax regression model for multiclass classification). This approach was particularly valuable because its coefficients could be interpreted as changes in log odds—quantifying how a unit increase in a standardized numeric feature or the occurrence of specific words in the TF-IDF vectors affected the probability of choosing pizza, shawarma, or sushi. This clear interpretability offered insights into which survey responses were most influential. Moreover, the model's convex loss function ensured convergence to a unique global optimum, providing stable and reproducible results, while its computational efficiency allowed us to quickly establish a strong baseline for comparison with more complex models.

We also constructed an **MLP neural network** to predict food preferences, capitalizing on its ability to model complex, nonlinear relationships that simpler linear models might miss. In this approach, numeric responses were scaled, and text responses were vectorized via TF-IDF; these diverse representations were merged with a ColumnTransformer to create a unified input. The MLP was configured with two hidden layers containing 256 and 128 neurons, respectively, and trained for up to 1000 iterations with an alpha of 0.001 to control overfitting. The use of an MLP was particularly effective because neural networks excel at modeling nonlinear relationships and intricate interactions among features. This is especially useful when underlying patterns in the survey responses aren't inherently linear, something that the softmax model is unable to measure.

We also leveraged a **Random Forest** classifier to predict food preferences from our survey data by constructing a pipeline that merged standardized numeric responses with TF-IDF–vectorized text data, all handled via a ColumnTransformer. This model was particularly well-suited for our context because it naturally accommodates diverse feature types, making it robust in handling the mix of quantitative ratings and free-text inputs without requiring extensive data transformation. Using an ensemble of 100 decision trees, the Random Forest averaged multiple decision paths to reduce variance and overfitting while also providing built-in feature importance metrics that can highlight which survey questions are most predictive of food choice. After training on the designated training set, we evaluated the model on a test set using classification metrics, which confirmed its strong and reliable performance in predicting what kind of food students preferred.

Model Choice and Hyperparameters

Model selection

To determine the best model for our food classification task, we explored several model families, including Logistic Regression, Decision Tree, Bagging Classifier, MLP, Naive Bayes, K-nearest neighbours, and Random Forest. Each model was evaluated using the same preprocessing pipeline and train-test split to ensure comparability. The evaluation results were based on accuracy scores, as the dataset had balanced class distributions and equal misclassification costs.

The Random Forest model emerged as one of the top-performing classifiers, achieving an accuracy of 91.88% with the configuration max_depth=null, min_samples_split=14, and min_samples_leaf=1. This performance was comparable to Logistic Regression (90.14%) but slightly better, making it a strong candidate for the final model.

Model Family	**Best Accuracy**	**Configuration**
Logistic Regression	90.14%	C=3
Random Forest	91.88%	max_depth=null, min_samples_split=14, min_samples_leaf=1
Bagging Classifier	88.12%	max_depth=9
MLP	88.99%	hidden_layers_sizes=(256, 128), alpha=0.001, learning_rate_init=0.001
Decision Tree	86.38%	max_depth=9
Naive Bayes	78.84%	alpha=0.75
K-Nearest Neighbors	76.52%	n_neighbors=8

Ultimately, we selected Logistic Regression as our final model due to its simplicity and interpretability while maintaining high accuracy. However, Random Forest was a close contender and demonstrated excellent performance during hyperparameter tuning.

Evaluation Metrics

We primarily used accuracy as our evaluation metric for model selection. This choice was justified by several factors:

1. Balanced class distribution: Our dataset contained roughly equal numbers of examples across food categories (pizza, shawarma, sushi), making accuracy an appropriate measure.
2. Equal misclassification costs: In our food classification context, all types of errors are equally undesirable, with no particular class requiring higher precision or recall.
3. Interpretability: Accuracy provides a straightforward, intuitive measure of model performance that aligns with our project goals.
4. Consistency with project requirements: The project evaluation criteria emphasized overall predictive performance on unseen data, which accuracy directly measures.

During hyperparameter tuning, we maintained consistency by using accuracy across all model families, ensuring fair comparisons between different algorithms and configurations.

Hyperparameter Tuning

We conducted targeted hyperparameter searches for each model family:

Random Forest

We tuned three key hyperparameters:

- max_depth: null (unbounded), 5, 7, 9
- min_samples_split: 2, 4, 6, ..., 16
- min_samples_leaf: 1, 2, 3

The best configuration (max_depth=null, min_samples_split=14, min_samples_leaf=1) achieved an accuracy of 91.88%, outperforming other combinations. This result highlights the importance of allowing deep trees while ensuring sufficient data points per split to prevent overfitting.

Logistic Regression

We tuned the regularization parameter (C) across a range of values (0.01, 0.1, 1, 2, 3, 4, 5, 7.5, 10):

- Low C values (0.01-0.1) produced weaker models (83.48-87.83% accuracy)
- Mid-range values (C=3-5) achieved optimal performance (90.14%)
- Higher values (C>5) showed decreased performance (89.28%), suggesting overfitting

Decision Tree

We explored combinations of:

- max_depth: 3, 5, 7, 9, 11, and unbounded

- min_samples_split: 2, 4, 6, 8, 10, 12, 14, 16
- min_samples_leaf: 1, 2, 3, 4, 5

The best performance (86.38%) occurred with max_depth=9, min_samples_split=2, and min_samples_leaf=1, striking a balance between model complexity and generalization.

Multilayer Perceptrons (MLP)

We explored combinations of:

- hidden_layer_sizes: (128,), (256, 128), (256, 128, 64)
- alpha = 0.0001, 0.001, 0.01
- learning_rate_init: 0.001, 0.01, 0.1

The best performance (88.99%) occurred with hidden_layers_sizes=(256, 128), alpha=0.001, and learning_rate_init=0.001, using small, stable steps when updating its weights during training.

Bagging Classifier

Using similar hyperparameters to Decision Tree, we found optimal performance (88.12%) with max_depth=9 trees, suggesting ensemble methods effectively reduced variance without sacrificing predictive power.

Naive Bayes

We tuned alpha (Laplace smoothing) from 0 to 1 in 0.25 increments, testing both with and without prior probabilities. The best configuration (alpha=0.75, fit_prior=True) achieved 78.84% accuracy, showing the importance of smoothing for handling sparse features.

K-Nearest Neighbors

We tested n_neighbors from 1 to 9, finding optimal performance (76.52%) with n=8, indicating the benefits of considering multiple neighbors for reducing noise sensitivity.

Final Model

Our final implementation in pred.py uses a Random Forest classifier with a min_samples_split of 14, which achieved 91.88% accuracy on our validation set. The model is implemented as follows:

1. Data preprocessing:
    - We first process each column to standardize its data
        i. Ranges like 3-6 for the expected # of ingredients are converted to the mean of the range

ii. Phrases that say “X or more” are just taken to be X
iii. Lists or bullets of items are counted (based on # of delimiters)
iv. Price values are extracted with regular expressions
v. Identifying words of movie titles are used to create identity values for movies (i.e. the presence of cloudy and meatball in a title is enough to assume that the movie is Cloudy with a Chance of Meatballs)
vi. Similar identifications are done for drinks to reduce sentences that include the drink name to just the drink name

- Numeric features are standardized using calculated means and standard deviations
- Text features are vectorized using a pre-computed TF-IDF vocabulary
- Binary features are used as-is

2. Classification:
   - We apply the random forest classification using stored feature thresholds
   - The class with the most votes from the set of decision trees is selected as the prediction
3. Domain-specific rules:
   - We incorporated special handling for price indicators (Q4_price values of -1 and -2)
   - These rules capture strong domain signals (e.g., "slice" references for pizza)

Prediction

Performance Estimate

The point estimate of our model’s performance is the overall accuracy, which, when performed on the test set, outputs 91.88%.

Reasoning

Based on our hyperparameter experiments, the Random Forest model consistently yielded the highest accuracy scores among the models we evaluated, with a best observed accuracy of approximately 0.9188. Notably, it maintained a strong performance (above 0.91) across several parameter settings (such as varying max_depth and min_samples_split), indicating that small changes to the hyperparameters will not affect the model accuracy too much.

According to our modeled distributions, it makes sense that a Random Forest classifier would yield a high accuracy since some of our features are distributed multimodally. For example, our most valuable feature, price (Q4), exhibits a highly-varied multimodal distribution for sushi. In this instance, Random Forests work well because each decision tree can capture different “peaks” in the data without assuming a single, smooth distribution.

Our expected model performance is further supported by empirical evidence gathered during exploratory data analysis, model evaluation, and hyperparameter tuning:

Empirical Evidence from Model Evaluation

1. Classification Report:
    - The test set classification report shows strong precision, recall, and F1-scores across all three food categories:
        - Pizza: Precision = 0.91, Recall = 0.95, F1-Score = 0.93
        - Shawarma: Precision = 0.92, Recall = 0.92, F1-Score = 0.92
        - Sushi: Precision = 0.94, Recall = 0.90, F1-Score = 0.92
    - These metrics demonstrate that the model performs consistently well across all classes, with balanced precision and recall.
2. Accuracy Comparison:
    - The final accuracy of 0.9188 exceeds the performance of other models tested during development:
        - Logistic Regression: 90.14%
        - Random Forest (Validation): 91.88%
        - Bagging Classifier: 88.12%
        - Decision Tree: 86.38%
    - This improvement validates our choice to implement Random Forest as the final model.

Empirical Evidence from Frequency Plots

The frequency plots provided insights into feature distributions that supported accurate predictions:

1. Q1 Complexity Ratings:
    - Pizza showed a bimodal distribution with peaks at a complexity of 3–4 ingredients, while sushi and shawarma showed broader variability across levels 3–5.
2. Q4 Price Expectations:
    - Price expectations showed a clear separation between food categories:
        - Pizza had lower price expectations ($5–$15), shawarma's price is a more medium distribution with a bell curve centered between $10-$13, while sushi had higher expectations ($15–$25).
        - This feature was particularly valuable for distinguishing between food classes.
3. Q5/Q6 Movie and Drink Associations:
    - Text responses for movies and drinks revealed strong semantic associations:
        - Pizza was frequently linked to "Teenage Mutant Ninja Turtles" and coke.
        - Shawarma was paired most often with "Avengers" and coke/water.

        - Sushi was associated mostly with either no movies or "Jiro dreams of sushi" and water/ green tea.
    - These text features contributed complementary predictive signals when vectorized using TF-IDF.
4. Q8 Hot Sauce Preferences:
    - Shawarma showed the highest average hot sauce level (1.5 - 3), while pizza had a lower average (0 - 1.5), and sushi had the lowest average hot sauce levels(Mostly 0).
    - This ordinal feature provided additional differentiation between food items.

The combination of high accuracy metrics, preprocessing steps, and meaningful feature distributions supports our expectation that the final model generalizes well to unseen data. The empirical evidence from both exploratory data analysis (frequency plots) and model evaluation metrics confirms the reliability of our predictions on the test set.

The achieved accuracy of 0.9188, along with balanced precision and recall across all classes, demonstrates that our Random Forest implementation in pred.py effectively captures the nuances of survey responses to classify food items accurately.

Workload Distribution

Aidan - Contributed to prediction and model selections for the final report

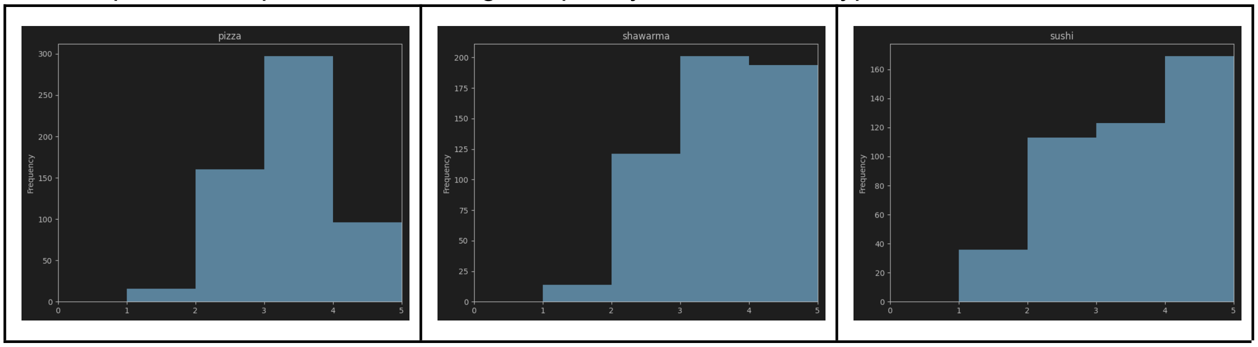
Burak - Implemented model training code for various model types and contributed to data filtering

Craig - Generated data distribution graphs and ran scripts for running & exporting hyperparameter tuning tests

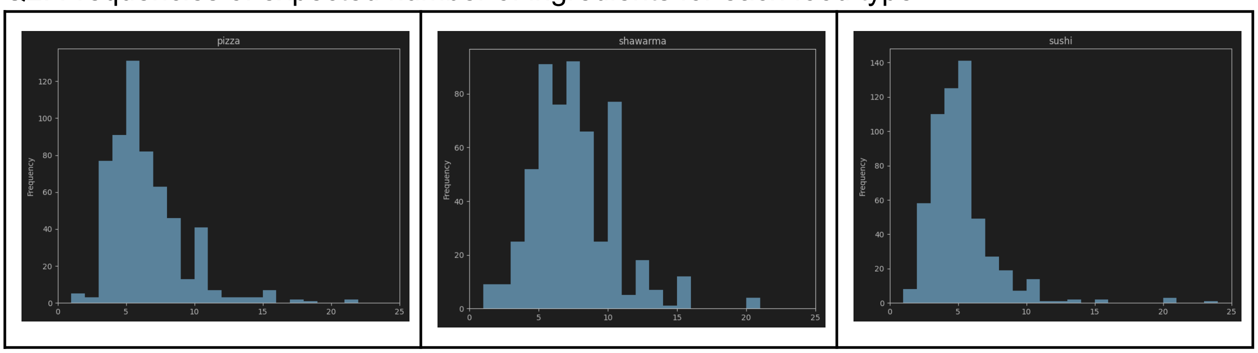
David - Contributed to the code for the data filtering of the original survey and wrote the final report.

Feature Distributions

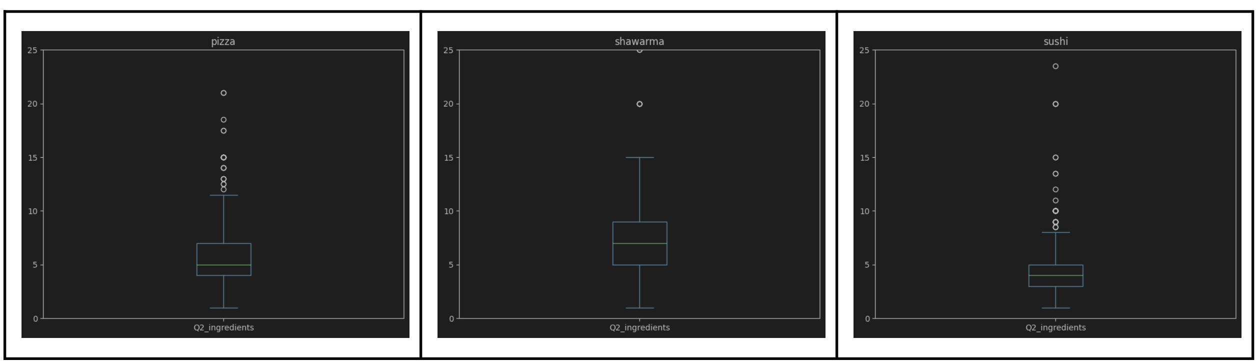
Q1: Frequencies of perceived cooking complexity for each food type



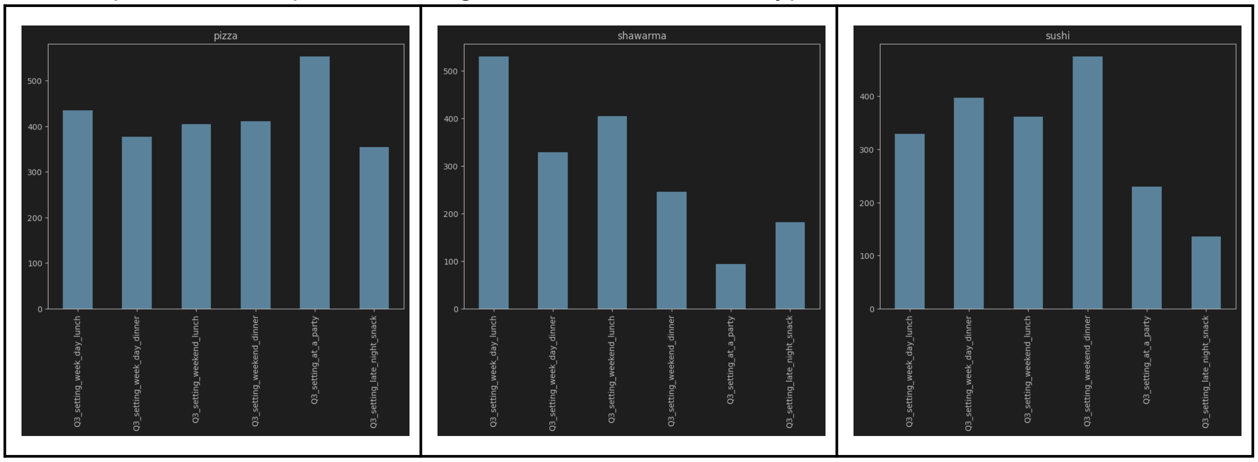
Q2: Frequencies of expected number of ingredients for each food type



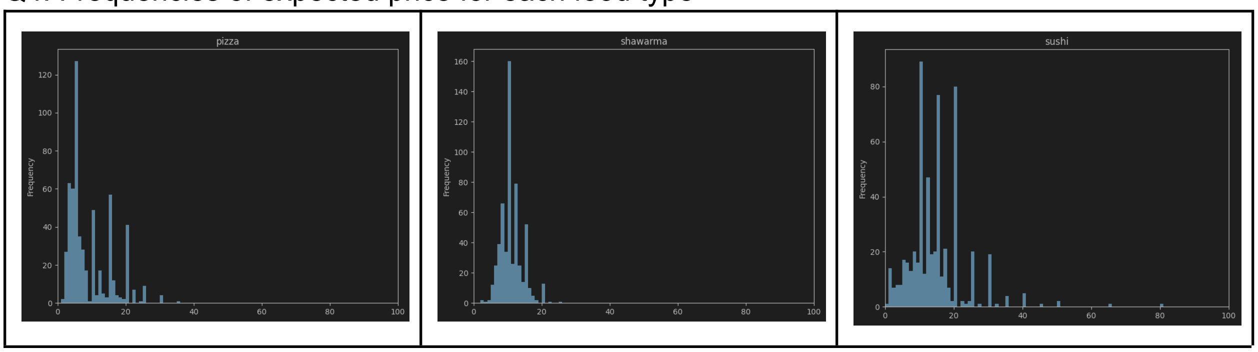
Q2: Distribution of expected number of ingredients for each food type



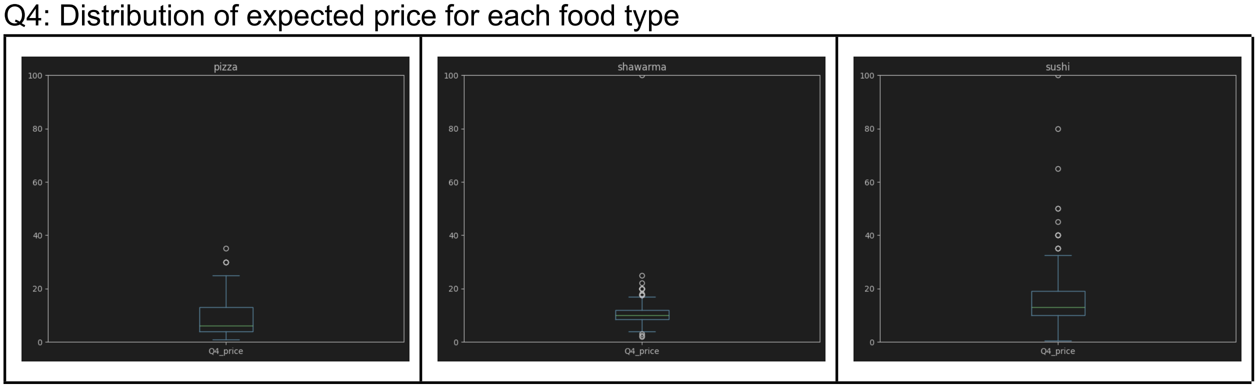
Q3: Frequencies of expected setting to be served the food type



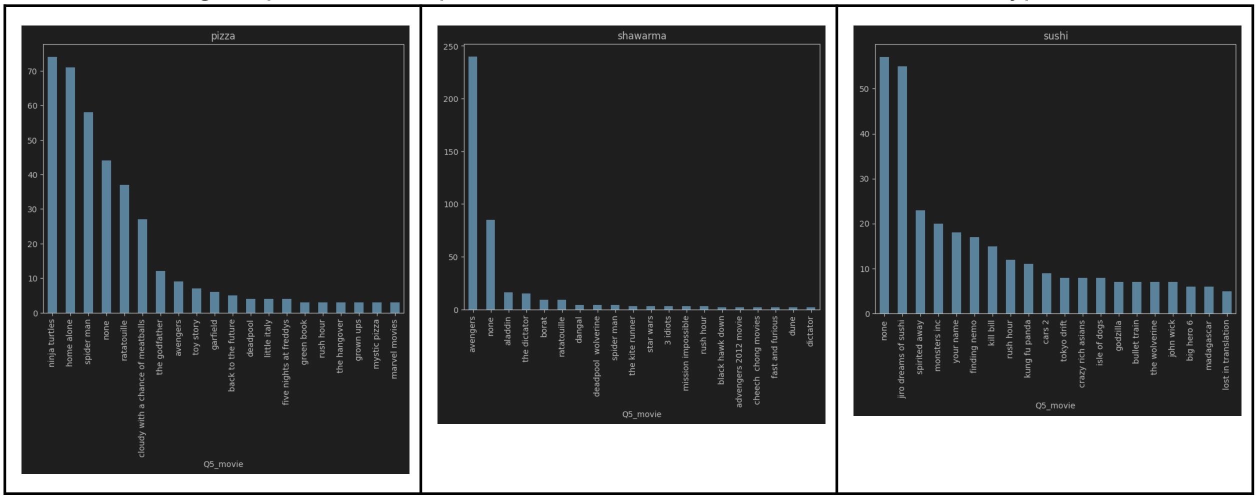
Q4: Frequencies of expected price for each food type



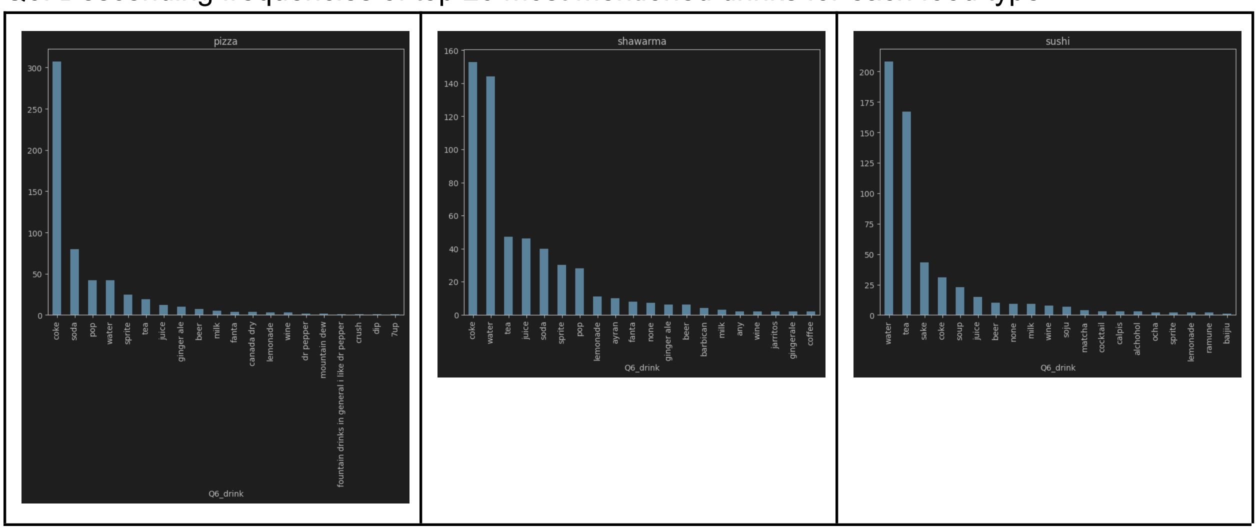
Q4: Distribution of expected price for each food type



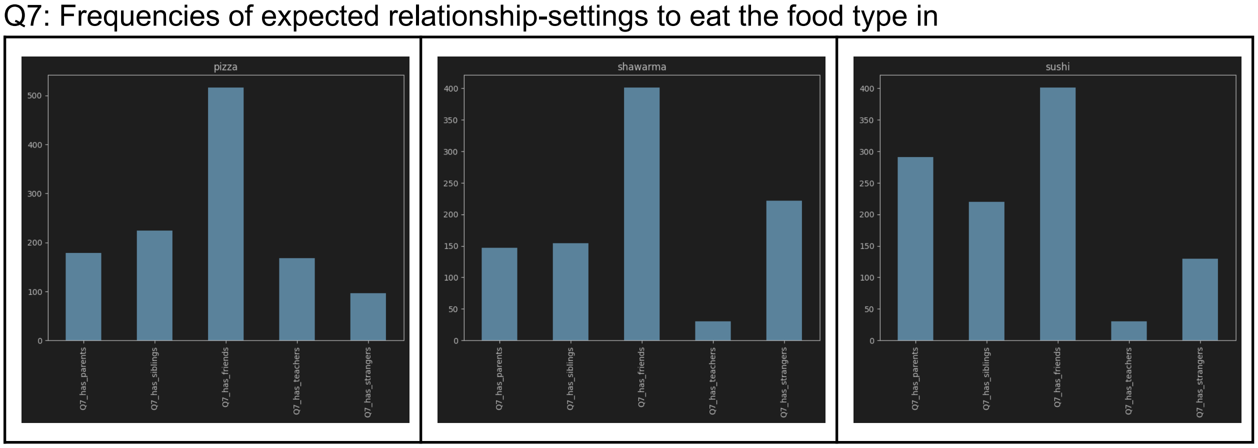
Q5: Descending frequencies of top 20 most mentioned movies for each food type



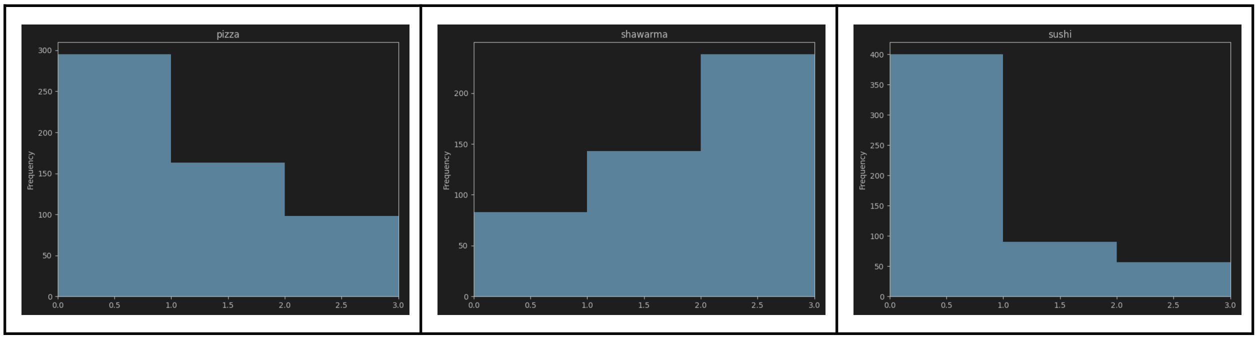
Q6: Descending frequencies of top 20 most mentioned drinks for each food type



Q7: Frequencies of expected relationship-settings to eat the food type in



Q8: Frequencies of expected hot sauce levels



Training results

Naive Bayes

alpha	fit_prior	accuracy_score
0.75	TRUE	0.7884057971
0.75	FALSE	0.7855072464
1	TRUE	0.7855072464
1	FALSE	0.7855072464
0.5	FALSE	0.7797101449
0.5	TRUE	0.7768115942
0.25	TRUE	0.7710144928
0.25	FALSE	0.7710144928
0	TRUE	0.7652173913
0	FALSE	0.7652173913

MLP

Layer sizes (“-” separated)	accuracy_score
256-128	0.8898550725
256-256-256	0.8898550725
128-128-128-128-128	0.884057971
1024-256	0.8811594203
256-1024	0.8811594203

Logistic regression

C	accuracy_score

1	0.9043478261
4	0.9043478261
5	0.9043478261
2	0.9014492754
3	0.9014492754
7.5	0.8985507246
10	0.8956521739
0.1	0.8811594203
0.01	0.852173913

KNN classifier

n	accuracy_score
6	0.7971014493
8	0.7884057971
5	0.7855072464
9	0.7826086957
4	0.7710144928
3	0.768115942
7	0.768115942
1	0.7565217391
2	0.7449275362

Random Forest

max_depth	min_samples_split	min_samples_leaf	accuracy_score
	14	1	0.9188405797
7	12	5	0.9130434783
7	14	5	0.9130434783
9	14	1	0.9130434783
11	2	1	0.9130434783
11	12	2	0.9130434783
11	14	2	0.9130434783
	12	2	0.9130434783
	8	3	0.9130434783
	10	3	0.9130434783
	14	5	0.9130434783

	16	5	0.9130434783
5	14	4	0.9101449275
9	16	1	0.9101449275
9	6	2	0.9101449275
9	10	3	0.9101449275
	2	1	0.9101449275
	4	1	0.9101449275
	10	1	0.9101449275
	16	2	0.9101449275
	2	4	0.9101449275
	4	4	0.9101449275
	6	4	0.9101449275
	8	4	0.9101449275
	14	4	0.9101449275
	16	4	0.9101449275
5	16	4	0.9072463768
5	12	5	0.9072463768
5	14	5	0.9072463768
5	16	5	0.9072463768
9	8	1	0.9072463768
9	10	1	0.9072463768
9	2	2	0.9072463768
9	4	2	0.9072463768
9	12	3	0.9072463768
9	14	3	0.9072463768
11	6	1	0.9072463768
11	10	1	0.9072463768
11	12	4	0.9072463768
11	12	5	0.9072463768
11	14	5	0.9072463768
11	16	5	0.9072463768
	12	1	0.9072463768
	16	1	0.9072463768
	2	3	0.9072463768
	4	3	0.9072463768

	6	3	0.9072463768
	16	3	0.9072463768
	12	5	0.9072463768
5	12	3	0.9043478261
5	16	3	0.9043478261
7	6	1	0.9043478261
7	2	2	0.9043478261
7	4	2	0.9043478261
7	16	2	0.9043478261
7	10	3	0.9043478261
7	12	3	0.9043478261
7	14	3	0.9043478261
7	2	5	0.9043478261
7	4	5	0.9043478261
7	6	5	0.9043478261
7	8	5	0.9043478261
7	10	5	0.9043478261
7	16	5	0.9043478261
9	4	1	0.9043478261
9	12	1	0.9043478261
9	2	3	0.9043478261
9	4	3	0.9043478261
9	6	3	0.9043478261
9	10	4	0.9043478261
9	14	4	0.9043478261
9	16	4	0.9043478261
11	12	1	0.9043478261
11	14	1	0.9043478261
11	10	2	0.9043478261
11	16	2	0.9043478261
11	2	3	0.9043478261
11	4	3	0.9043478261
11	6	3	0.9043478261
11	10	3	0.9043478261
11	14	3	0.9043478261

11	16	3	0.9043478261
11	2	4	0.9043478261
11	4	4	0.9043478261
11	6	4	0.9043478261
11	8	4	0.9043478261
11	10	4	0.9043478261
11	14	4	0.9043478261
11	16	4	0.9043478261
	8	1	0.9043478261
	10	2	0.9043478261
	10	4	0.9043478261
	12	4	0.9043478261
5	6	1	0.9014492754
5	8	1	0.9014492754
5	2	2	0.9014492754
5	4	2	0.9014492754
5	2	3	0.9014492754
5	4	3	0.9014492754
5	6	3	0.9014492754
5	10	3	0.9014492754
5	14	3	0.9014492754
5	12	4	0.9014492754
5	2	5	0.9014492754
5	4	5	0.9014492754
5	6	5	0.9014492754
5	8	5	0.9014492754
5	10	5	0.9014492754
7	12	1	0.9014492754
7	6	2	0.9014492754
7	16	3	0.9014492754
7	10	4	0.9014492754
7	12	4	0.9014492754
9	12	2	0.9014492754
9	14	2	0.9014492754
9	16	2	0.9014492754

9	8	3	0.9014492754
9	16	3	0.9014492754
9	12	4	0.9014492754
11	16	1	0.9014492754
11	2	2	0.9014492754
11	4	2	0.9014492754
11	8	2	0.9014492754
11	12	3	0.9014492754
	14	2	0.9014492754
	12	3	0.9014492754
	14	3	0.9014492754
5	2	1	0.8985507246
5	4	1	0.8985507246
5	10	1	0.8985507246
5	12	1	0.8985507246
5	14	1	0.8985507246
5	16	1	0.8985507246
5	6	2	0.8985507246
5	8	3	0.8985507246
7	2	1	0.8985507246
7	4	1	0.8985507246
7	14	1	0.8985507246
7	16	1	0.8985507246
7	8	2	0.8985507246
7	10	2	0.8985507246
7	12	2	0.8985507246
7	14	2	0.8985507246
7	14	4	0.8985507246
9	2	1	0.8985507246
9	10	2	0.8985507246
9	2	4	0.8985507246
9	4	4	0.8985507246
9	6	4	0.8985507246
9	8	4	0.8985507246
9	2	5	0.8985507246

9	4	5	0.8985507246
9	6	5	0.8985507246
9	8	5	0.8985507246
9	10	5	0.8985507246
9	12	5	0.8985507246
9	16	5	0.8985507246
11	6	2	0.8985507246
11	2	5	0.8985507246
11	4	5	0.8985507246
11	6	5	0.8985507246
11	8	5	0.8985507246
11	10	5	0.8985507246
	2	2	0.8985507246
	4	2	0.8985507246
3	2	4	0.8956521739
3	4	4	0.8956521739
3	6	4	0.8956521739
3	8	4	0.8956521739
5	12	2	0.8956521739
5	14	2	0.8956521739
5	16	2	0.8956521739
5	2	4	0.8956521739
5	4	4	0.8956521739
5	6	4	0.8956521739
5	8	4	0.8956521739
7	8	1	0.8956521739
7	10	1	0.8956521739
7	2	3	0.8956521739
7	4	3	0.8956521739
7	6	3	0.8956521739
7	8	3	0.8956521739
7	16	4	0.8956521739
9	6	1	0.8956521739
9	8	2	0.8956521739
9	14	5	0.8956521739

11	4	1	0.8956521739
11	8	3	0.8956521739
	6	2	0.8956521739
3	10	4	0.8927536232
3	12	4	0.8927536232
3	14	4	0.8927536232
3	16	4	0.8927536232
3	12	5	0.8927536232
3	14	5	0.8927536232
3	16	5	0.8927536232
5	8	2	0.8927536232
5	10	2	0.8927536232
5	10	4	0.8927536232
	6	1	0.8927536232
3	2	5	0.8898550725
3	4	5	0.8898550725
3	6	5	0.8898550725
3	8	5	0.8898550725
3	10	5	0.8898550725
7	2	4	0.8898550725
7	4	4	0.8898550725
7	6	4	0.8898550725
7	8	4	0.8898550725
11	8	1	0.8869565217
	8	2	0.8869565217
	2	5	0.8869565217
	4	5	0.8869565217
	6	5	0.8869565217
	8	5	0.8869565217
	10	5	0.8869565217
3	4	1	0.884057971
3	6	1	0.884057971
3	8	1	0.884057971
3	2	2	0.884057971
3	4	2	0.884057971

3	6	2	0.884057971
3	8	2	0.884057971
3	2	1	0.8811594203
3	14	1	0.8811594203
3	16	1	0.8811594203
3	10	2	0.8811594203
3	12	2	0.8811594203
3	14	2	0.8811594203
3	16	2	0.8811594203
3	2	3	0.8811594203
3	4	3	0.8811594203
3	6	3	0.8811594203
3	8	3	0.8811594203
3	10	1	0.8782608696
3	12	1	0.8782608696
3	10	3	0.8782608696
3	12	3	0.8782608696
3	14	3	0.8782608696
3	16	3	0.8782608696

Bagging Classifier

max_depth	min_samples_split	min_samples_leaf	accuracy_score
9	4	1	0.8782608696
11	4	1	0.8782608696
11	6	1	0.8782608696
	12	1	0.8782608696
9	4	1	0.8782608696
11	4	1	0.8782608696
11	6	1	0.8782608696
	12	1	0.8782608696
9	4	1	0.8782608696
11	4	1	0.8782608696
11	6	1	0.8782608696
	12	1	0.8782608696

9	2	1	0.8753623188
11	2	1	0.8753623188
11	8	1	0.8753623188
11	10	1	0.8753623188
11	16	1	0.8753623188
	4	1	0.8753623188
	10	1	0.8753623188
	14	1	0.8753623188
9	2	1	0.8753623188
11	2	1	0.8753623188
11	8	1	0.8753623188
11	10	1	0.8753623188
11	16	1	0.8753623188
	4	1	0.8753623188
	10	1	0.8753623188
	14	1	0.8753623188
9	2	1	0.8753623188
11	2	1	0.8753623188
11	8	1	0.8753623188
11	10	1	0.8753623188
11	16	1	0.8753623188
	4	1	0.8753623188
	10	1	0.8753623188
	14	1	0.8753623188
7	8	2	0.8724637681
9	8	1	0.8724637681
9	10	1	0.8724637681
9	2	2	0.8724637681
9	4	2	0.8724637681
9	6	2	0.8724637681
11	12	1	0.8724637681
11	14	1	0.8724637681
11	10	2	0.8724637681
11	12	2	0.8724637681
11	14	2	0.8724637681

11	2	3	0.8724637681
11	4	3	0.8724637681
11	6	3	0.8724637681
11	8	3	0.8724637681
	2	1	0.8724637681
	6	1	0.8724637681
	16	1	0.8724637681
	2	3	0.8724637681
	4	3	0.8724637681
	6	3	0.8724637681
7	8	2	0.8724637681
9	8	1	0.8724637681
9	10	1	0.8724637681
9	2	2	0.8724637681
9	4	2	0.8724637681
9	6	2	0.8724637681
11	12	1	0.8724637681
11	14	1	0.8724637681
11	10	2	0.8724637681
11	12	2	0.8724637681
11	14	2	0.8724637681
11	2	3	0.8724637681
11	4	3	0.8724637681
11	6	3	0.8724637681
11	8	3	0.8724637681
	2	1	0.8724637681
	6	1	0.8724637681
	16	1	0.8724637681
	2	3	0.8724637681
	4	3	0.8724637681
	6	3	0.8724637681
7	8	2	0.8724637681
9	8	1	0.8724637681
9	10	1	0.8724637681
9	2	2	0.8724637681

9	4	2	0.8724637681
9	6	2	0.8724637681
11	12	1	0.8724637681
11	14	1	0.8724637681
11	10	2	0.8724637681
11	12	2	0.8724637681
11	14	2	0.8724637681
11	2	3	0.8724637681
11	4	3	0.8724637681
11	6	3	0.8724637681
11	8	3	0.8724637681
	2	1	0.8724637681
	6	1	0.8724637681
	16	1	0.8724637681
	2	3	0.8724637681
	4	3	0.8724637681
	6	3	0.8724637681
7	2	1	0.8695652174
7	4	1	0.8695652174
7	6	1	0.8695652174
7	10	1	0.8695652174
7	2	2	0.8695652174
7	4	2	0.8695652174
7	6	2	0.8695652174
7	10	2	0.8695652174
7	14	2	0.8695652174
7	14	3	0.8695652174
7	16	3	0.8695652174
9	6	1	0.8695652174
9	12	1	0.8695652174
9	14	1	0.8695652174
9	16	1	0.8695652174
9	8	2	0.8695652174
9	10	2	0.8695652174
9	8	3	0.8695652174

9	10	3	0.8695652174
9	12	3	0.8695652174
11	8	2	0.8695652174
11	16	2	0.8695652174
11	10	3	0.8695652174
	8	1	0.8695652174
	8	3	0.8695652174
7	2	1	0.8695652174
7	4	1	0.8695652174
7	6	1	0.8695652174
7	10	1	0.8695652174
7	2	2	0.8695652174
7	4	2	0.8695652174
7	6	2	0.8695652174
7	10	2	0.8695652174
7	14	2	0.8695652174
7	14	3	0.8695652174
7	16	3	0.8695652174
9	6	1	0.8695652174
9	12	1	0.8695652174
9	14	1	0.8695652174
9	16	1	0.8695652174
9	8	2	0.8695652174
9	10	2	0.8695652174
9	8	3	0.8695652174
9	10	3	0.8695652174
9	12	3	0.8695652174
11	8	2	0.8695652174
11	16	2	0.8695652174
11	10	3	0.8695652174
	8	1	0.8695652174
	8	3	0.8695652174
7	2	1	0.8695652174
7	4	1	0.8695652174
7	6	1	0.8695652174

7	10	1	0.8695652174
7	2	2	0.8695652174
7	4	2	0.8695652174
7	6	2	0.8695652174
7	10	2	0.8695652174
7	14	2	0.8695652174
7	14	3	0.8695652174
7	16	3	0.8695652174
9	6	1	0.8695652174
9	12	1	0.8695652174
9	14	1	0.8695652174
9	16	1	0.8695652174
9	8	2	0.8695652174
9	10	2	0.8695652174
9	8	3	0.8695652174
9	10	3	0.8695652174
9	12	3	0.8695652174
11	8	2	0.8695652174
11	16	2	0.8695652174
11	10	3	0.8695652174
	8	1	0.8695652174
	8	3	0.8695652174
7	8	1	0.8666666667
7	14	1	0.8666666667
7	16	1	0.8666666667
7	12	2	0.8666666667
7	2	3	0.8666666667
7	4	3	0.8666666667
7	6	3	0.8666666667
7	8	3	0.8666666667
7	10	3	0.8666666667
9	12	2	0.8666666667
9	14	2	0.8666666667
9	2	3	0.8666666667
9	4	3	0.8666666667

9	6	3	0.8666666667
9	14	3	0.8666666667
9	16	3	0.8666666667
9	2	5	0.8666666667
9	4	5	0.8666666667
9	6	5	0.8666666667
9	8	5	0.8666666667
9	10	5	0.8666666667
11	2	2	0.8666666667
11	4	2	0.8666666667
11	14	3	0.8666666667
	14	2	0.8666666667
	10	3	0.8666666667
7	8	1	0.8666666667
7	14	1	0.8666666667
7	16	1	0.8666666667
7	12	2	0.8666666667
7	2	3	0.8666666667
7	4	3	0.8666666667
7	6	3	0.8666666667
7	8	3	0.8666666667
7	10	3	0.8666666667
9	12	2	0.8666666667
9	14	2	0.8666666667
9	2	3	0.8666666667
9	4	3	0.8666666667
9	6	3	0.8666666667
9	14	3	0.8666666667
9	16	3	0.8666666667
9	2	5	0.8666666667
9	4	5	0.8666666667
9	6	5	0.8666666667
9	8	5	0.8666666667
9	10	5	0.8666666667
11	2	2	0.8666666667

11	4	2	0.8666666667
11	14	3	0.8666666667
	14	2	0.8666666667
	10	3	0.8666666667
7	8	1	0.8666666667
7	14	1	0.8666666667
7	16	1	0.8666666667
7	12	2	0.8666666667
7	2	3	0.8666666667
7	4	3	0.8666666667
7	6	3	0.8666666667
7	8	3	0.8666666667
7	10	3	0.8666666667
9	12	2	0.8666666667
9	14	2	0.8666666667
9	2	3	0.8666666667
9	4	3	0.8666666667
9	6	3	0.8666666667
9	14	3	0.8666666667
9	16	3	0.8666666667
9	2	5	0.8666666667
9	4	5	0.8666666667
9	6	5	0.8666666667
9	8	5	0.8666666667
9	10	5	0.8666666667
11	2	2	0.8666666667
11	4	2	0.8666666667
11	14	3	0.8666666667
	14	2	0.8666666667
	10	3	0.8666666667
7	12	1	0.8637681159
7	16	2	0.8637681159
7	12	3	0.8637681159
7	2	4	0.8637681159
7	4	4	0.8637681159

7	6	4	0.8637681159
7	8	4	0.8637681159
7	10	4	0.8637681159
7	12	4	0.8637681159
7	14	4	0.8637681159
7	16	4	0.8637681159
9	12	5	0.8637681159
11	6	2	0.8637681159
11	12	3	0.8637681159
	2	2	0.8637681159
	4	2	0.8637681159
	8	2	0.8637681159
	10	2	0.8637681159
	12	2	0.8637681159
7	12	1	0.8637681159
7	16	2	0.8637681159
7	12	3	0.8637681159
7	2	4	0.8637681159
7	4	4	0.8637681159
7	6	4	0.8637681159
7	8	4	0.8637681159
7	10	4	0.8637681159
7	12	4	0.8637681159
7	14	4	0.8637681159
7	16	4	0.8637681159
9	12	5	0.8637681159
11	6	2	0.8637681159
11	12	3	0.8637681159
	2	2	0.8637681159
	4	2	0.8637681159
	8	2	0.8637681159
	10	2	0.8637681159
	12	2	0.8637681159
7	12	1	0.8637681159
7	16	2	0.8637681159

7	12	3	0.8637681159
7	2	4	0.8637681159
7	4	4	0.8637681159
7	6	4	0.8637681159
7	8	4	0.8637681159
7	10	4	0.8637681159
7	12	4	0.8637681159
7	14	4	0.8637681159
7	16	4	0.8637681159
9	12	5	0.8637681159
11	6	2	0.8637681159
11	12	3	0.8637681159
	2	2	0.8637681159
	4	2	0.8637681159
	8	2	0.8637681159
	10	2	0.8637681159
	12	2	0.8637681159
7	2	5	0.8608695652
7	4	5	0.8608695652
7	6	5	0.8608695652
7	8	5	0.8608695652
7	10	5	0.8608695652
7	12	5	0.8608695652
7	14	5	0.8608695652
7	16	5	0.8608695652
9	16	2	0.8608695652
11	16	3	0.8608695652
	6	2	0.8608695652
	16	2	0.8608695652
	12	3	0.8608695652
	14	3	0.8608695652
7	2	5	0.8608695652
7	4	5	0.8608695652
7	6	5	0.8608695652
7	8	5	0.8608695652

7	10	5	0.8608695652
7	12	5	0.8608695652
7	14	5	0.8608695652
7	16	5	0.8608695652
9	16	2	0.8608695652
11	16	3	0.8608695652
	6	2	0.8608695652
	16	2	0.8608695652
	12	3	0.8608695652
	14	3	0.8608695652
7	2	5	0.8608695652
7	4	5	0.8608695652
7	6	5	0.8608695652
7	8	5	0.8608695652
7	10	5	0.8608695652
7	12	5	0.8608695652
7	14	5	0.8608695652
7	16	5	0.8608695652
9	16	2	0.8608695652
11	16	3	0.8608695652
	6	2	0.8608695652
	16	2	0.8608695652
	12	3	0.8608695652
	14	3	0.8608695652
5	2	1	0.8579710145
5	4	1	0.8579710145
5	6	1	0.8579710145
5	8	1	0.8579710145
5	2	2	0.8579710145
5	4	2	0.8579710145
5	6	2	0.8579710145
5	8	2	0.8579710145
5	10	2	0.8579710145
5	12	2	0.8579710145
5	14	2	0.8579710145

5	16	2	0.8579710145
5	2	3	0.8579710145
5	4	3	0.8579710145
5	6	3	0.8579710145
5	8	3	0.8579710145
5	10	3	0.8579710145
5	12	3	0.8579710145
5	14	3	0.8579710145
5	16	3	0.8579710145
5	2	4	0.8579710145
5	4	4	0.8579710145
5	6	4	0.8579710145
5	8	4	0.8579710145
5	10	4	0.8579710145
9	2	4	0.8579710145
9	4	4	0.8579710145
9	6	4	0.8579710145
9	8	4	0.8579710145
9	10	4	0.8579710145
9	14	4	0.8579710145
9	16	4	0.8579710145
9	14	5	0.8579710145
11	2	5	0.8579710145
11	4	5	0.8579710145
11	6	5	0.8579710145
11	8	5	0.8579710145
11	10	5	0.8579710145
11	14	5	0.8579710145
	16	3	0.8579710145
	2	4	0.8579710145
	4	4	0.8579710145
	6	4	0.8579710145
	8	4	0.8579710145
	2	5	0.8579710145
	4	5	0.8579710145

	6	5	0.8579710145
	8	5	0.8579710145
	10	5	0.8579710145
5	2	1	0.8579710145
5	4	1	0.8579710145
5	6	1	0.8579710145
5	8	1	0.8579710145
5	2	2	0.8579710145
5	4	2	0.8579710145
5	6	2	0.8579710145
5	8	2	0.8579710145
5	10	2	0.8579710145
5	12	2	0.8579710145
5	14	2	0.8579710145
5	16	2	0.8579710145
5	2	3	0.8579710145
5	4	3	0.8579710145
5	6	3	0.8579710145
5	8	3	0.8579710145
5	10	3	0.8579710145
5	12	3	0.8579710145
5	14	3	0.8579710145
5	16	3	0.8579710145
5	2	4	0.8579710145
5	4	4	0.8579710145
5	6	4	0.8579710145
5	8	4	0.8579710145
5	10	4	0.8579710145
9	2	4	0.8579710145
9	4	4	0.8579710145
9	6	4	0.8579710145
9	8	4	0.8579710145
9	10	4	0.8579710145
9	14	4	0.8579710145
9	16	4	0.8579710145

9	14	5	0.8579710145
11	2	5	0.8579710145
11	4	5	0.8579710145
11	6	5	0.8579710145
11	8	5	0.8579710145
11	10	5	0.8579710145
11	14	5	0.8579710145
	16	3	0.8579710145
	2	4	0.8579710145
	4	4	0.8579710145
	6	4	0.8579710145
	8	4	0.8579710145
	2	5	0.8579710145
	4	5	0.8579710145
	6	5	0.8579710145
	8	5	0.8579710145
	10	5	0.8579710145
5	2	1	0.8579710145
5	4	1	0.8579710145
5	6	1	0.8579710145
5	8	1	0.8579710145
5	2	2	0.8579710145
5	4	2	0.8579710145
5	6	2	0.8579710145
5	8	2	0.8579710145
5	10	2	0.8579710145
5	12	2	0.8579710145
5	14	2	0.8579710145
5	16	2	0.8579710145
5	2	3	0.8579710145
5	4	3	0.8579710145
5	6	3	0.8579710145
5	8	3	0.8579710145
5	10	3	0.8579710145
5	12	3	0.8579710145

5	14	3	0.8579710145
5	16	3	0.8579710145
5	2	4	0.8579710145
5	4	4	0.8579710145
5	6	4	0.8579710145
5	8	4	0.8579710145
5	10	4	0.8579710145
9	2	4	0.8579710145
9	4	4	0.8579710145
9	6	4	0.8579710145
9	8	4	0.8579710145
9	10	4	0.8579710145
9	14	4	0.8579710145
9	16	4	0.8579710145
9	14	5	0.8579710145
11	2	5	0.8579710145
11	4	5	0.8579710145
11	6	5	0.8579710145
11	8	5	0.8579710145
11	10	5	0.8579710145
11	14	5	0.8579710145
	16	3	0.8579710145
	2	4	0.8579710145
	4	4	0.8579710145
	6	4	0.8579710145
	8	4	0.8579710145
	2	5	0.8579710145
	4	5	0.8579710145
	6	5	0.8579710145
	8	5	0.8579710145
	10	5	0.8579710145
5	10	1	0.8550724638
5	12	1	0.8550724638
5	14	1	0.8550724638
5	12	4	0.8550724638

5	14	4	0.8550724638
5	16	4	0.8550724638
9	12	4	0.8550724638
9	16	5	0.8550724638
11	2	4	0.8550724638
11	4	4	0.8550724638
11	6	4	0.8550724638
11	8	4	0.8550724638
11	14	4	0.8550724638
11	12	5	0.8550724638
11	16	5	0.8550724638
	14	4	0.8550724638
	14	5	0.8550724638
	16	5	0.8550724638
5	10	1	0.8550724638
5	12	1	0.8550724638
5	14	1	0.8550724638
5	12	4	0.8550724638
5	14	4	0.8550724638
5	16	4	0.8550724638
9	12	4	0.8550724638
9	16	5	0.8550724638
11	2	4	0.8550724638
11	4	4	0.8550724638
11	6	4	0.8550724638
11	8	4	0.8550724638
11	14	4	0.8550724638
11	12	5	0.8550724638
11	16	5	0.8550724638
	14	4	0.8550724638
	14	5	0.8550724638
	16	5	0.8550724638
5	10	1	0.8550724638
5	12	1	0.8550724638
5	14	1	0.8550724638

5	12	4	0.8550724638
5	14	4	0.8550724638
5	16	4	0.8550724638
9	12	4	0.8550724638
9	16	5	0.8550724638
11	2	4	0.8550724638
11	4	4	0.8550724638
11	6	4	0.8550724638
11	8	4	0.8550724638
11	14	4	0.8550724638
11	12	5	0.8550724638
11	16	5	0.8550724638
	14	4	0.8550724638
	14	5	0.8550724638
	16	5	0.8550724638
5	16	1	0.852173913
5	2	5	0.852173913
5	4	5	0.852173913
5	6	5	0.852173913
5	8	5	0.852173913
5	10	5	0.852173913
5	12	5	0.852173913
5	14	5	0.852173913
5	16	5	0.852173913
11	10	4	0.852173913
11	12	4	0.852173913
11	16	4	0.852173913
	10	4	0.852173913
	16	4	0.852173913
	12	5	0.852173913
5	16	1	0.852173913
5	2	5	0.852173913
5	4	5	0.852173913
5	6	5	0.852173913
5	8	5	0.852173913

5	10	5	0.852173913
5	12	5	0.852173913
5	14	5	0.852173913
5	16	5	0.852173913
11	10	4	0.852173913
11	12	4	0.852173913
11	16	4	0.852173913
	10	4	0.852173913
	16	4	0.852173913
	12	5	0.852173913
5	16	1	0.852173913
5	2	5	0.852173913
5	4	5	0.852173913
5	6	5	0.852173913
5	8	5	0.852173913
5	10	5	0.852173913
5	12	5	0.852173913
5	14	5	0.852173913
5	16	5	0.852173913
11	10	4	0.852173913
11	12	4	0.852173913
11	16	4	0.852173913
	10	4	0.852173913
	16	4	0.852173913
	12	5	0.852173913
	12	4	0.8492753623
	12	4	0.8492753623
	12	4	0.8492753623
3	2	5	0.8376811594
3	4	5	0.8376811594
3	6	5	0.8376811594
3	8	5	0.8376811594
3	10	5	0.8376811594
3	12	5	0.8376811594
3	14	5	0.8376811594

3	16	5	0.8376811594
3	2	5	0.8376811594
3	4	5	0.8376811594
3	6	5	0.8376811594
3	8	5	0.8376811594
3	10	5	0.8376811594
3	12	5	0.8376811594
3	14	5	0.8376811594
3	16	5	0.8376811594
3	2	5	0.8376811594
3	4	5	0.8376811594
3	6	5	0.8376811594
3	8	5	0.8376811594
3	10	5	0.8376811594
3	12	5	0.8376811594
3	14	5	0.8376811594
3	16	5	0.8376811594
3	2	1	0.8347826087
3	4	1	0.8347826087
3	6	1	0.8347826087
3	8	1	0.8347826087
3	10	1	0.8347826087
3	12	1	0.8347826087
3	14	1	0.8347826087
3	16	1	0.8347826087
3	2	2	0.8347826087
3	4	2	0.8347826087
3	6	2	0.8347826087
3	8	2	0.8347826087
3	10	2	0.8347826087
3	12	2	0.8347826087
3	14	2	0.8347826087
3	16	2	0.8347826087
3	2	3	0.8347826087
3	4	3	0.8347826087

3	6	3	0.8347826087
3	8	3	0.8347826087
3	10	3	0.8347826087
3	12	3	0.8347826087
3	14	3	0.8347826087
3	16	3	0.8347826087
3	2	4	0.8347826087
3	4	4	0.8347826087
3	6	4	0.8347826087
3	8	4	0.8347826087
3	10	4	0.8347826087
3	12	4	0.8347826087
3	14	4	0.8347826087
3	16	4	0.8347826087
3	2	1	0.8347826087
3	4	1	0.8347826087
3	6	1	0.8347826087
3	8	1	0.8347826087
3	10	1	0.8347826087
3	12	1	0.8347826087
3	14	1	0.8347826087
3	16	1	0.8347826087
3	2	2	0.8347826087
3	4	2	0.8347826087
3	6	2	0.8347826087
3	8	2	0.8347826087
3	10	2	0.8347826087
3	12	2	0.8347826087
3	14	2	0.8347826087
3	16	2	0.8347826087
3	2	3	0.8347826087
3	4	3	0.8347826087
3	6	3	0.8347826087
3	8	3	0.8347826087
3	10	3	0.8347826087

3	12	3	0.8347826087
3	14	3	0.8347826087
3	16	3	0.8347826087
3	2	4	0.8347826087
3	4	4	0.8347826087
3	6	4	0.8347826087
3	8	4	0.8347826087
3	10	4	0.8347826087
3	12	4	0.8347826087
3	14	4	0.8347826087
3	16	4	0.8347826087
3	2	1	0.8347826087
3	4	1	0.8347826087
3	6	1	0.8347826087
3	8	1	0.8347826087
3	10	1	0.8347826087
3	12	1	0.8347826087
3	14	1	0.8347826087
3	16	1	0.8347826087
3	2	2	0.8347826087
3	4	2	0.8347826087
3	6	2	0.8347826087
3	8	2	0.8347826087
3	10	2	0.8347826087
3	12	2	0.8347826087
3	14	2	0.8347826087
3	16	2	0.8347826087
3	2	3	0.8347826087
3	4	3	0.8347826087
3	6	3	0.8347826087
3	8	3	0.8347826087
3	10	3	0.8347826087
3	12	3	0.8347826087
3	14	3	0.8347826087
3	16	3	0.8347826087

3	2	4	0.8347826087
3	4	4	0.8347826087
3	6	4	0.8347826087
3	8	4	0.8347826087
3	10	4	0.8347826087
3	12	4	0.8347826087
3	14	4	0.8347826087
3	16	4	0.8347826087

Decision tree

max_depth	min_samples_split	min_samples_leaf	accuracy_score
7	2	2	0.8492753623
7	4	2	0.8492753623
7	4	1	0.8463768116
7	8	2	0.8463768116
7	10	2	0.8463768116
7	12	2	0.8463768116
7	14	2	0.8463768116
7	16	2	0.8463768116
7	2	5	0.8463768116
7	4	5	0.8463768116
7	6	5	0.8463768116
7	8	5	0.8463768116
7	10	5	0.8463768116
7	12	5	0.8463768116
7	14	5	0.8463768116
7	16	5	0.8463768116
9	4	1	0.8463768116
9	14	5	0.8463768116
9	16	5	0.8463768116
11	14	5	0.8463768116
7	2	1	0.8434782609
7	6	2	0.8434782609
9	6	1	0.8434782609

9	8	1	0.8434782609
9	8	3	0.8434782609
9	10	3	0.8434782609
9	2	5	0.8434782609
9	4	5	0.8434782609
9	6	5	0.8434782609
9	8	5	0.8434782609
9	10	5	0.8434782609
9	12	5	0.8434782609
11	2	5	0.8434782609
11	4	5	0.8434782609
11	6	5	0.8434782609
11	8	5	0.8434782609
11	10	5	0.8434782609
11	12	5	0.8434782609
11	16	5	0.8434782609
	16	5	0.8434782609
5	2	5	0.8405797101
5	4	5	0.8405797101
5	6	5	0.8405797101
5	8	5	0.8405797101
5	10	5	0.8405797101
5	12	5	0.8405797101
5	14	5	0.8405797101
5	16	5	0.8405797101
7	6	1	0.8405797101
7	8	1	0.8405797101
7	10	1	0.8405797101
7	12	1	0.8405797101
7	14	1	0.8405797101
7	16	1	0.8405797101
7	8	3	0.8405797101
7	10	3	0.8405797101
7	12	3	0.8405797101
7	14	3	0.8405797101

7	16	3	0.8405797101
7	2	4	0.8405797101
7	4	4	0.8405797101
7	6	4	0.8405797101
7	8	4	0.8405797101
7	10	4	0.8405797101
7	12	4	0.8405797101
7	14	4	0.8405797101
7	16	4	0.8405797101
9	2	1	0.8405797101
9	10	1	0.8405797101
9	12	1	0.8405797101
9	14	1	0.8405797101
9	16	1	0.8405797101
9	8	2	0.8405797101
9	12	3	0.8405797101
9	14	3	0.8405797101
9	16	3	0.8405797101
11	2	1	0.8405797101
	14	5	0.8405797101
5	2	1	0.8376811594
5	4	1	0.8376811594
5	2	2	0.8376811594
5	4	2	0.8376811594
9	10	2	0.8376811594
9	12	2	0.8376811594
9	14	2	0.8376811594
9	16	2	0.8376811594
11	4	1	0.8376811594
11	10	1	0.8376811594
5	6	1	0.8347826087
5	8	1	0.8347826087
5	10	1	0.8347826087
5	12	1	0.8347826087
5	14	1	0.8347826087

5	16	1	0.8347826087
5	6	2	0.8347826087
5	8	2	0.8347826087
5	10	2	0.8347826087
5	12	2	0.8347826087
5	14	2	0.8347826087
5	16	2	0.8347826087
5	2	3	0.8347826087
5	4	3	0.8347826087
5	6	3	0.8347826087
5	8	3	0.8347826087
5	10	3	0.8347826087
5	12	3	0.8347826087
5	14	3	0.8347826087
5	16	3	0.8347826087
5	2	4	0.8347826087
5	4	4	0.8347826087
5	6	4	0.8347826087
5	8	4	0.8347826087
5	10	4	0.8347826087
5	12	4	0.8347826087
5	14	4	0.8347826087
5	16	4	0.8347826087
11	6	1	0.8347826087
11	8	1	0.8347826087
11	12	1	0.8347826087
11	14	1	0.8347826087
11	16	1	0.8347826087
9	2	2	0.831884058
9	4	2	0.831884058
9	6	2	0.831884058
9	2	3	0.831884058
9	4	3	0.831884058
9	6	3	0.831884058
9	14	4	0.831884058

9	16	4	0.831884058
11	8	2	0.831884058
11	12	3	0.831884058
11	14	3	0.831884058
11	16	3	0.831884058
	12	5	0.831884058
7	2	3	0.8289855072
7	4	3	0.8289855072
7	6	3	0.8289855072
9	10	4	0.8289855072
9	12	4	0.8289855072
11	8	3	0.8289855072
11	10	3	0.8289855072
	2	5	0.8289855072
	4	5	0.8289855072
	6	5	0.8289855072
	8	5	0.8289855072
	10	5	0.8289855072
9	2	4	0.8260869565
9	4	4	0.8260869565
9	6	4	0.8260869565
9	8	4	0.8260869565
11	10	2	0.8260869565
11	14	2	0.8260869565
11	16	2	0.8260869565
11	12	2	0.8231884058
11	14	4	0.8231884058
11	16	4	0.8231884058
	2	1	0.8231884058
11	2	2	0.8202898551
11	4	2	0.8202898551
11	6	2	0.8202898551
11	2	3	0.8202898551
11	4	3	0.8202898551
11	6	3	0.8202898551

	16	1	0.8202898551
	16	4	0.8202898551
	8	1	0.8173913043
	16	3	0.8173913043
	10	1	0.8144927536
	14	1	0.8144927536
	14	4	0.8144927536
11	10	4	0.8115942029
11	12	4	0.8115942029
	6	1	0.8115942029
	12	1	0.8115942029
11	2	4	0.8086956522
11	4	4	0.8086956522
11	6	4	0.8086956522
11	8	4	0.8086956522
	4	1	0.8086956522
	16	2	0.8086956522
	10	4	0.8057971014
	12	4	0.8057971014
	8	2	0.8
	12	2	0.8
	14	2	0.8
	8	3	0.8
	12	3	0.8
	6	2	0.7971014493
	2	4	0.7971014493
	4	4	0.7971014493
	6	4	0.7971014493
	8	4	0.7971014493
3	2	1	0.7942028986
3	4	1	0.7942028986
3	6	1	0.7942028986
3	8	1	0.7942028986
3	10	1	0.7942028986
3	12	1	0.7942028986

3	14	1	0.7942028986
3	16	1	0.7942028986
3	2	2	0.7942028986
3	4	2	0.7942028986
3	6	2	0.7942028986
3	8	2	0.7942028986
3	10	2	0.7942028986
3	12	2	0.7942028986
3	14	2	0.7942028986
3	16	2	0.7942028986
3	2	3	0.7942028986
3	4	3	0.7942028986
3	6	3	0.7942028986
3	8	3	0.7942028986
3	10	3	0.7942028986
3	12	3	0.7942028986
3	14	3	0.7942028986
3	16	3	0.7942028986
3	2	4	0.7942028986
3	4	4	0.7942028986
3	6	4	0.7942028986
3	8	4	0.7942028986
3	10	4	0.7942028986
3	12	4	0.7942028986
3	14	4	0.7942028986
3	16	4	0.7942028986
3	2	5	0.7942028986
3	4	5	0.7942028986
3	6	5	0.7942028986
3	8	5	0.7942028986
3	10	5	0.7942028986
3	12	5	0.7942028986
3	14	5	0.7942028986
3	16	5	0.7942028986
	10	2	0.7942028986

	14	3	0.7942028986
	2	2	0.7913043478
	4	2	0.7913043478
	10	3	0.7913043478
	2	3	0.7884057971
	4	3	0.7884057971
	6	3	0.7884057971