Attribution statement **Team name: RubberCorn**

Our team, whose signatures appear below, completed this project as a group effort. By our signatures, we indicate that we agree that each of us has made the following contributions.

Member 1 (enter your contributions here and then both print and sign your name)

Brainstormed preprocessing steps like column dropping, implemented multi hot encoding, formalized the writing for the extra credits, ran experiments on the models, etc.

We worked on the project over the course of multiple calls and in person meetings.

Maliha Tasnim

Member 2 (enter your contributions here and then both print and sign your name)

Brainstormed preprocessing steps like getting rid of certain instances, formalized the writing for the main project, ran experiments on the models, etc.

We worked on the project over the course of multiple calls and in person meetings.

Arshadul Monir

arshadul monir

Table 0. Dated History of Models

Date	Type	Accuracy	F1	MCC	Precision	Recall
May 5	Fulfill on Test	0.5790	0.4738	0.3129	0.5067	0.4567
May 5	Satisfy on Test	0.6109	0.4871	0.3290	0.5049	0.4761
May 11	Fulfill on Test	0.6171	0.5090	0.3865	0.5548	0.4903
May 11	Satisfy on Test	0.6557	0.4670	0.3797	0.5284	0.4420
May 14	Fulfill on Test	0.6277	0.5317	0.4074	0.5706	0.5151
May 14	Satisfy on Test	0.6624	0.5060	0.4050	0.5548	0.4818
May 11	Fulfill on Holdout	0.5427	0.4594	0.3061	0.4606	0.4769
May 11	Satisfy on Holdout	0.6303	0.4062	0.3292	0.4719	0.3917
May 14	Fulfill on Holdout	0.6258	0.5351	0.4073	0.5739	0.5192
May 14	Satisfy on Holdout	0.6581	0.4886	0.4009	0.5345	0.4697

Task 11

a) For satisfy_RubberCorn, the attribute we are learning to predict is q61, which represents how satisfied the employee is with their work. For fulfill_RubberCorn, the attribute we are learning to predict is q62, which represents how fulfilled the employee is with their work. The features for each attribute correspond to different questions about the employees.

Table 1. Feature Types and Descriptions in the Employee Survey Dataset

Features	Information
form_end, form_start (the first two columns)	These features represent the time an employee started the form and the time the employee ended the form. In the dataset, these features are represented by time stamps.
q01-q12, q30, q33, q36, q38-q39, q41-q44, q46-q49, q51, q53-q60	These features have two possible responses and each feature can have only one of the two responses. These questions measure the typical behavior of the employee.
q13-q29, q31-q32, q34-q35, q37, q40, q45, q50, q52	These features also have two possible responses. Each feature can have only one response. These questions make the employee choose between two words. These questions reveal what ideas the employee resonates with.
q61-q62	These two features are the attributes we will be predicting. q61 will be used for predicting q62 and q62 will be used for predicting q61. These features are represented by the numbers -100, -33, 33, and 100.
q63-q67, q74	These features have way more than 2 possible choices. For every feature of this type, there are 2^n - 1 possible answers where n is the number of possible answers to

Table 1. Feature Types and Descriptions in the Employee Survey Dataset	
	pick for on the Google form. It's also possible to select more than one answer choice.
q68-q70	These features have more than 2 possible responses but multiple choices cannot be selected. The questions are directly about the company.
q71-q72	These features have two possible responses and each feature can have only one of the two responses. They represent gender and marital status.
q73	This feature represents the answer to the question "I am dependent," which, given the answer choices, is asking who the employee is responsible for. The question on the form does not allow selection for multiple choices even though that would make sense in this case. The answer choices list the possible number of children that are dependent on the employee as well as other types of dependents.

This table summarizes the features for each attribute we are trying to predict based on response format and purpose.

b) Both q61 and q62 take on one of four possible values: -100, -33, 33, and 100. These values represent levels of satisfaction and fulfillment. Although the features are represented numerically, they represent qualitative levels, not measurable quantities. The differences between the values are not guaranteed to be meaningful. Therefore, this problem is a classification problem. It's specifically a multinomial classification problem as we are trying to predict which of 4 distinct classes an employee falls into.

c) In our preprocessing we approached the features based on the following categories: binary features, the alternate target feature, the multi select features, and the one hot features. We used a custom function transformer on all the binary features. Since the answer choices are 1 and 2, our

transformation decreases the value by 1 to keep within on the scale [0,1]. For the alternate target feature (if q61 is the target, q62 is the alternate target feature and vice versa), we transformed using MinMaxScaler to turn its values of -100, -33, 33, and 100 to values in between [0,1] as we believed there to be significance in the ordinal values. For the multi select features, we used a custom encoder to multi hot encode any feature that could contain a list of answers. It works by creating a new feature for each possible answer seen, similar to one-hot encoding. However, instead of only one of the columns having a value of 1 and the rest being 1**.** For the one-hot features, we used the OneHotEncoder to encode the features with more than 2 categorical values. We made two separate, almost identical pipelines. However, for the satisfy preprocessing, our alternate target feature was q62 and for the fulfill preprocessing, the target feature was q61.

Figure 1. Satisfy Pipeline

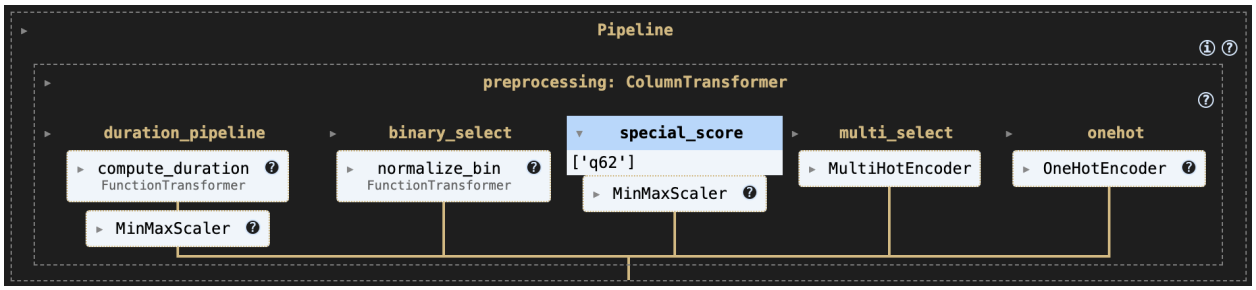
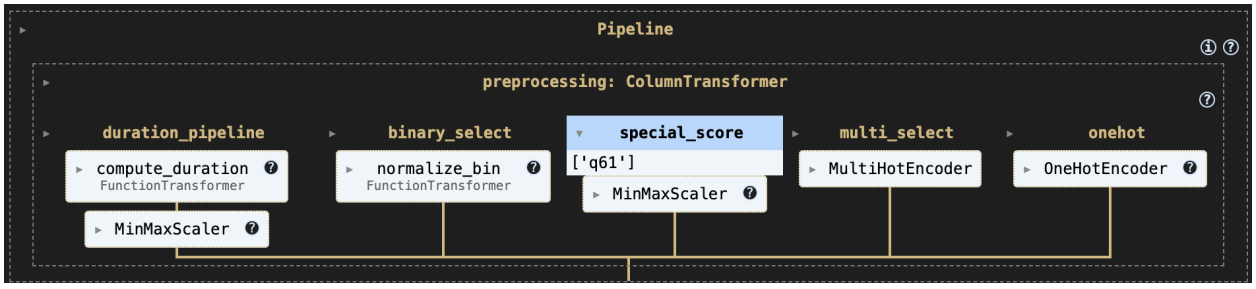


Figure 2. Fulfill Pipeline



This table describes how we preprocessed our features:

Table 2. Preprocessing Steps and Details	
Feature Groups	Modifications
form_end, form_start	The time difference is calculated between the form_end and form_start columns and is stored as the number of seconds in a "duration" column. The number of seconds is then MinMax scaled within the range [0,1]
q01-q60	A custom function is used in a FunctionTransformer to change the inputs from 1 and 2 to 0 and 1 to stay within the range [0,1].
q61/q62	For the satisfy set, q62 is MinMax scaled within the range [0,1]. For the fulfill set, q61 is MinMax scaled in between the range [0,1].
q63-q67, q73-q74	A custom transformer is used to multi-hot encode these features. A new feature is created for each unique value found in the entire column. A 1 is placed in the columns that are associated with any values that existed in the list of values in the original feature. A 0 is placed in the rest of the columns
q68-q70	OneHotEncoder is used to create new features for each unique value found in the entire column. A 1 is placed in only the column that is associated with the value in the original feature. The rest of the features have 0.

d) During the training, the number of instances we dropped was dependent on a threshold set by a custom function, remove_trolls, which filtered out instances from the training set before any preprocessing was done. The function worked by finding the date_time difference between the form_end and the form_start column and keeping only the instances where the difference is within a specified boundary. In order to not delete too many instances from the training set, we got rid of any instance that took less than 480 seconds as when doing the form a few times (after a lengthy pause so that we don't remember our answers), our completion time was around 10

minutes on average so we felt that 8 minutes was a good lower bound estimate. On top of this, we also filtered out any instances that took over 30 minutes (1800 seconds) as taking over 30 minutes most likely meant the person was distracted while filling out the form or took a long break while filling it out which could reflect possible insincerity or a lack of seriousness in the responses. After this, a total of 11916 instances were removed from the training set. While this is a substantial number of instances, it improved the Accuracy, F1 Score, Precision, Recall, and MCC

e) The only two features that were dropped from both the satisfy and fulfill datasets were form_end and form_start. These features were dropped as they, alone, were just unique date-time identifiers associated with each instance. However, they were both used together to create a duration feature which was useful in pruning instances in which a submission took too little or too much time to be seen as valid for learning purposes.

f) The algorithms we chose for both satisfy_RubberCorn and fulfill_RubberCorn were XGBClassifier, RandomForestClassifier, and VotingClassifier using RandomForestClassifier and HistGradientBoosting. XGBClassifier was used as it's great for large datasets, works well with multiclass classification, and handles imbalanced classes by focusing on hard-to-predict cases. VotingClassifier with RandomForestClassifier and HistGradientBoostingClassifier as voting classifiers work well with classifiers with already high accuracies (which RandomForestClassifier and HistGradientBoostingClassifier had) and even better when the learners are different types of classifiers. RandomForestClassifier was used as it works well with on large datasets, overfitting was managed using the right hyperparameters without diminishing its test scores, and it’s also a good standalone algorithm for multinomial classification even outside of VotingClassifier.

   We used the same metrics for both satisfy_RubberCorn and fulfil_RubberCorn which were accuracy (as a simple and intuitive baseline), precision (a measure of the false positive rate), recall (a measure of the false negative rate, f1-score (to see the balance between precision and recall), and mcc (which considers the aspects of the confusion matrix).

g) For both satisfy_RubberCorn and fulfill_RubberCorn, we used RandomizedSearchCV with 4-fold cross-validation to tune hyperparameters.

4-fold RandomizedSearchCV splits the training data into 4 equal parts. For each hyperparameter setting, the model is trained on three folds and validated on the fourth. The process is then repeated with the validation fold rotating until every fold has been used as validation once. The average performance across all four folds is then used to compare hyperparameter combinations. Because we used a VotingClassifier, we had to tune the parameters for two different algorithms, RandomForest and HistGradientBoosting.

```
Best Parameters:
'classifier__hgb__l2_regularization': np.float64(0.3398231196727128)
'classifier__hgb__learning_rate': np.float64(0.02883701168472723)
'classifier__hgb__max_depth': 7
'classifier__hgb__max_iter': 312
'classifier__hgb__max_leaf_nodes': 37
'classifier__hgb__min_samples_leaf': 15
'classifier__rf__bootstrap': False
'classifier__rf__max_depth': 18
'classifier__rf__max_features': 'sqrt'
'classifier__rf__min_samples_leaf': 3
'classifier__rf__min_samples_split': 5
'classifier__rf__n_estimators': 379

Best CV F1 Score:
0.5528974580144655
```

Task 12

We used a single preprocessing method for both satisfy_RubberCorn and fulfill_RubberCorn. To determine the best model statistically, we performed the tukey test with the three models on the five metrics mentioned. The results for satisfy_RubberCorn are below:

Table 3. Tukey HSD Test Results for Satisfy

```
Metric: accuracy
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.011406 0.274144 -0.007856  0.030668          no
     RF  Voting   0.031368 0.003570  0.012107  0.050630         yes
    XGB  Voting   0.019962 0.042676  0.000700  0.039224         yes

Metric: mcc
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.031881 0.109258 -0.007047  0.070809          no
     RF  Voting   0.018034 0.433267 -0.020894  0.056962          no
    XGB  Voting  -0.013847 0.599136 -0.052775  0.025081          no

Metric: f1
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.017940 0.380543 -0.017815  0.053695          no
     RF  Voting  -0.043555 0.019415 -0.079310 -0.007800         yes
    XGB  Voting  -0.061496 0.002503 -0.097251 -0.025741         yes

Metric: precision
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.123727 0.017608  0.024025  0.223429         yes
     RF  Voting   0.144035 0.007483  0.044333  0.243737         yes
    XGB  Voting   0.020308 0.839687 -0.079394  0.120010          no

Metric: recall
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.030517 0.100166 -0.005840  0.066874          no
     RF  Voting  -0.096887 0.000104 -0.133244 -0.060530         yes
    XGB  Voting  -0.127404 0.000011 -0.163761 -0.091047         yes
```

The results of the test reveal the following metrics:

- Accuracy: RandomForest classifier significantly outperforms Voting classifier. XGB classifier significantly outperforms Voting classifier. However there is no significant relationship between RandomForest classifier and XGB classifier.
    - **No single algorithm** learned better to predict q61 according to the accuracy
- MCC: There was no significant relation between any of the algorithms
    - **No single algorithm** learned to predict q61 according to the mcc
- F1: There was no significant relationship between the RandomForest classifier and the XGB classifier. The Voting classifier significantly outperforms both the RandomForest classifier and the XGB classifier
    - **VotingClassifier** learned better to predict q62 according the f1-score

- Precision: RandomForest classifier significantly outperforms both XGB classifier and Voting classifier
    - **RandomForestClassifier** learned better to predict q62 according the precision
- Recall: Voting classifier significantly outperforms both XGB classifier and RandomForest classifier
    - **VotingClassifier** learned better to predict q62 according the precision

According to the results, **VotingClassifier worked the best for satisfy_RubberCorn** as it had the best f1-score, which is a better metric of performance than precision.

The following shows the tukey test results for fulfill_RubberCorn

Figure 4. Tukey HSD Results for Fulfill

```
Metric: accuracy
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.012078 0.319495 -0.009843  0.033999          no
     RF  Voting   0.005032 0.801929 -0.016889  0.026953          no
    XGB  Voting  -0.007046 0.655274 -0.028967  0.014875          no

Metric: mcc
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.029901 0.042927  0.001012  0.058790         yes
     RF  Voting  -0.011682 0.521370 -0.040571  0.017207          no
    XGB  Voting  -0.041583 0.007647 -0.070472 -0.012694         yes

Metric: f1
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB  -0.000048 0.999982 -0.023482  0.023386          no
     RF  Voting  -0.036999 0.004345 -0.060433 -0.013565         yes
    XGB  Voting  -0.036951 0.004381 -0.060385 -0.013517         yes

Metric: precision
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.018633 0.327490 -0.015608  0.052873          no
     RF  Voting   0.014083 0.510625 -0.020158  0.048323          no
    XGB  Voting  -0.004550 0.927525 -0.038790  0.029690          no

Metric: recall
Model 1 Model 2  Statistic  p-value  CI Lower  CI Upper Significant
     RF     XGB   0.004272 0.818331 -0.015279  0.023823          no
     RF  Voting  -0.045894 0.000276 -0.065445 -0.026343         yes
    XGB  Voting  -0.050166 0.000140 -0.069717 -0.030615         yes
```

The results of the test reveal the following metrics:

- Accuracy: There was no significant relation between any of the algorithms
    - **No single algorithm** learned better to predict q62 according to the accuracy
- MCC: RandomForest classifier significantly outperforms XGB classifier. Voting classifier also significantly outperforms XGB classifier. However, there was no significant relation between RandomForest classifier and Voting classifier.
    - **No single algorithm** learned to predict q62 according to the mcc

- F1: Voting classifier significantly outperforms both RandomForest classifier and XGBoost classifier
    - **VotingClassifier** learned better to predict q62 according the f1-score
- Precision: There was no significant relation between any of the algorithms
    - **No single algorithm** learned better to predict q62 according the precision
- Recall: Voting classifier significantly outperforms both RandomForest classifier and XGBoost classifier
    - **VotingClassifier** learned better to predict q62 according the precision

According to the results, **VotingClassifier worked the best for satisfy_RubberCorn** as it had the best f1-score and recall while the other metrics had no algorithm that was significantly better than the rest

Task 13

a) In the end, the model wasn't able to be very accurate, with most of the metrics for even the best algorithm being in the range of 0.5-0.7 and an average accuracy of around 65%. Despite the fact that there were no missing values in the dataset, we believe that the form questions weren't relevant enough to the target questions to create a great predictor. The correlation values for all the features to the target features showed that, apart from the correlations q61 and q62 to each other, the highest correlation value to q61 was 0.230088 and the highest correlation value to q62 to 0.199880. Even when looking at the questions on the google form, it's apparent that most of the questions are irrelevant to question 61 and 62. While questions 61 and 62 are about your personal opinions about your job, almost all of the questions have nothing to do with your job while the rest relate to personal preferences regarding a job but not to the current job of the person themselves. If there were more questions about the person's current job, it would definitely result in many more correlation columns and therefore better predictions. Another thing could be that, even though we did remove the possible invalid instances (troll submissions), our method probably removed many valid instances and missed many other invalid instances such as people who didn't take the form seriously but were slow in submitting or people that were

distracted but still rushed the submission which led to delays but quick answers, leading to a time that appears normal.

b) We don't think we produced a fully trustworthy predictor but it is a fair predictor. The average accuracies of both satisfy_RubberCorn and fulfill_Rubbercorn are between the low end of 60%-70%. Since this is a multinomial classification problem with 4 target classes, the result is far better than the average 25% accuracy that would have been achieved through random guessing. Despite this, the model has substantial inaccuracy, especially when predicting the minority class, and therefore cannot be considered a fully trustworthy or reliable predictor.

c) Some results that surprised us was the fact that PCA or most other dimensionality reduction methods made both the training and test results worse. Since the correlation values were so low, we didn't expect that removing some features with low correlation scores would affect the metrics significantly, much less that they would decrease the metrics. Something that pleased us was that after training on a dataset that had the troll responses removed, the accuracies increased. However, not all minimum and maximum times increased performance and finding the optimal range to where there was a good amount of training examples still existing but also where the metrics increased and the amount of time removed can still be seen as an unreasonable amount of time. Something that disappointed us was that adding an additional column that would calculate the sum of the weighted values for every feature that could contain multiple values negatively affected the performance of the model as that method took a good amount of time to implement and ended up being a scrapped idea.

Extra Credit One Report

To explore whether predictors differ by gender, we created separate models using the satisfy dataset filtered by q71 (the column that represents the gender of the employee). The algorithm we used was Voting Classifier with Random Forest and Histogram Gradient Boosting. We initially started out with the hyperparameters and preprocessing steps from previous successes.

Table 1. Satisfy Male Preprocessing Steps	
Dropped Features	form_start, form_end, q71, q22, q31
Duration Pipeline (with MinMaxScaler)	Include a scaled duration column
Scaled	q62
Binary Features	q01-q60, q72 (not including q22 and q31)
Multihot Features	q63-q67, q73, q74
One-hot Features	q68, q69, q70
Troll Response Removal	Removed instances with durations lower than 300 seconds

Outlines the preprocessing pipeline applied to the male subset of the dataset for predicting satisfaction.

Table 2. Satisfy Female Preprocessing Steps	
Dropped Features	form_start, form_end, q71, q08, q35
Duration Pipeline (with MinMaxScaler)	Include a scaled duration column
Scaled	q62
Binary Features	q01-q60, q72 (not including q08 and q35)
Multihot Features	q63-q67, q73, q74
One-hot Features	q68, q69, q70
Troll Response Removal	Removed instances with durations lower than 300 seconds

Outlines preprocessing steps for modeling satisfaction among females.

Table 3. Fulfill Male Preprocessing Steps	
Dropped Features	form_start, form_end, q71, q08, q06
Duration Pipeline (with MinMaxScaler)	Include a scaled duration column
Scaled	q61
Binary Features	q01-q60, q72 (not including q08 and q06)
Multihot Features	q63-q67, q73, q74
One-hot Features	q68, q69, q70

Outlines preprocessing steps for modeling fulfillment among males.

Table 4. Fulfill Female Preprocessing Steps	
Dropped Features	form_start, form_end, q71, q22, q31
Duration Pipeline (with MinMaxScaler)	Include a scaled duration column
Scaled	q61
Binary Features	q01-q60, q72 (not including q22 and q31)
Multihot Features	q63-q67, q73, q74
One-hot Features	q68, q69, q70

Outlines preprocessing steps for modeling fulfillment among females.

As you can see from Tables 1-4, the major difference between these preprocessing steps is the idea of dropping columns beyond form_start and form_end. For each training set, we calculated the correlation values for each feature against the target. From there, we ran a loop that prints the results of what happens when a certain feature from q01-q60 is dropped. This allowed us to observe the impact of dropping a single column from the dataset.
Let's take the Satisfy Male set for example.

Table 5. Displays the best results from dropping a feature using the Accuracy metric.

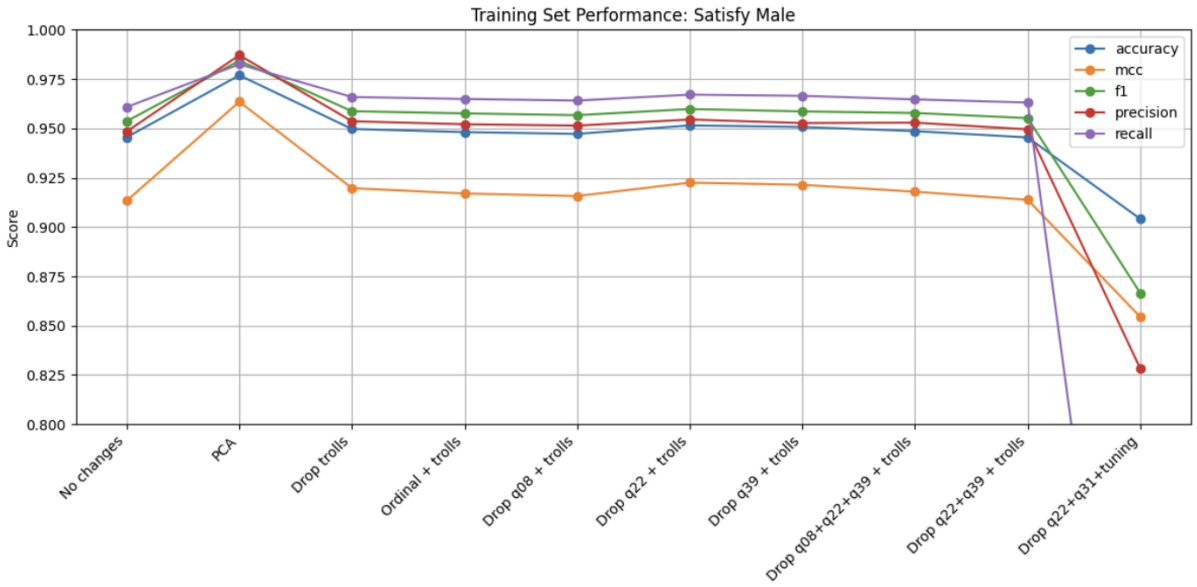
Dropped Feature	CV_Accuracy	CV_F1	CV_Precision	CV_Recall	CV_MCC	Test_Accuracy	Test_F1	Test_Precision	Test_Recall	Test_MCC
q39	0.6394	0.5282	0.5028	0.5830	0.4052	0.6312	0.5729	0.5464	0.6267	0.4329
q28	0.6383	0.5280	0.5021	0.5847	0.4041	0.6459	0.5821	0.5561	0.6331	0.4529
q59	0.6383	0.5260	0.5007	0.5807	0.4044	0.6477	0.5866	0.5604	0.6358	0.4543
q56	0.6380	0.5251	0.4999	0.5809	0.4043	0.6385	0.5797	0.5524	0.6334	0.4395
q37	0.6378	0.5269	0.5016	0.5815	0.4026	0.6330	0.5767	0.5488	0.6362	0.4363
q29	0.6378	0.5256	0.5006	0.5792	0.4026	0.6514	0.5968	0.5688	0.6495	0.4613
q01	0.6376	0.5255	0.5004	0.5798	0.4030	0.6349	0.5760	0.5491	0.6273	0.4373
q06	0.6375	0.5248	0.4998	0.5800	0.4031	0.6440	0.5853	0.5575	0.6420	0.4506
q10	0.6375	0.5269	0.5014	0.5822	0.4027	0.6495	0.5875	0.5611	0.6389	0.4613
q33	0.6374	0.5268	0.5015	0.5810	0.4017	0.6330	0.5706	0.5440	0.6256	0.4358

Displays the “best” features to drop based on how dropping it improves the accuracy of the models predictions.

Based on the values given from running the column dropping loop as well as the general ideas expressed by printing out the correlations, we experimented with several combinations of dropped columns. We noticed that although it would make sense to get rid of all the columns below a certain correlation threshold, getting rid of all of them actually made the performance of our model worse on the metrics of Accuracy, F1 Score, Precision, Recall, and MCC. This suggests that some features, although weakly correlated with the target individually, may contribute value when interacting with other features. Therefore, feature interactions played a significant role in model performance, and dropping too many columns indiscriminately harmed the model’s ability to generalize effectively. Eventually, we settled on the combination of dropping q22 and q39.

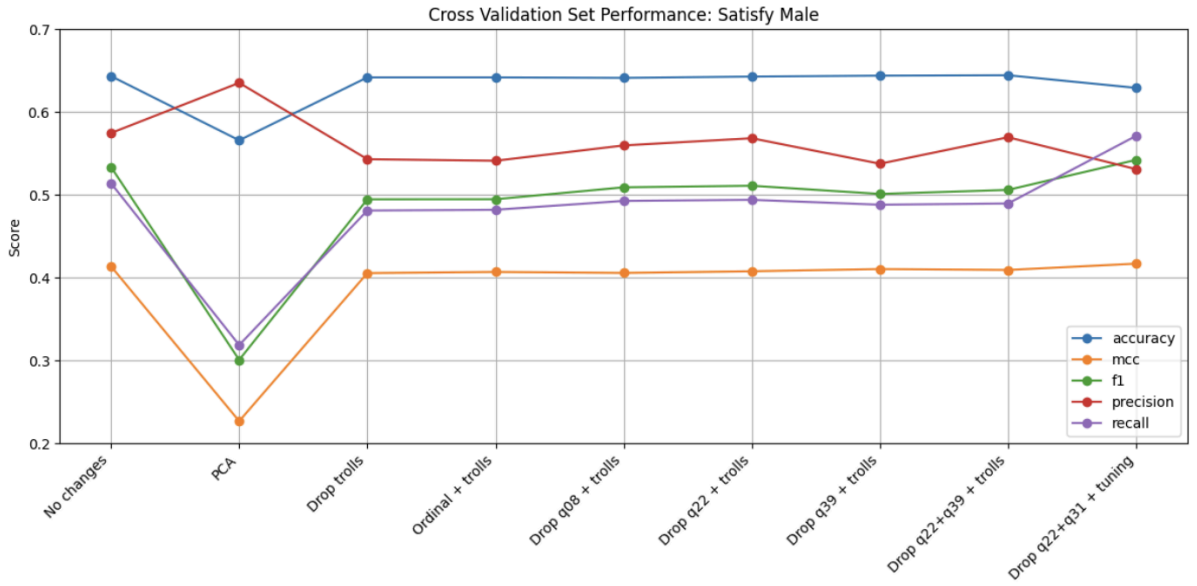
The graphs below show the results of the preprocessing steps we tried on the training set, the training set with cross validation, and our test set. This is to compare model performance between Satisfy Male and Satisfy Female.

Figure 1. Training Set Performance for Satisfy Male



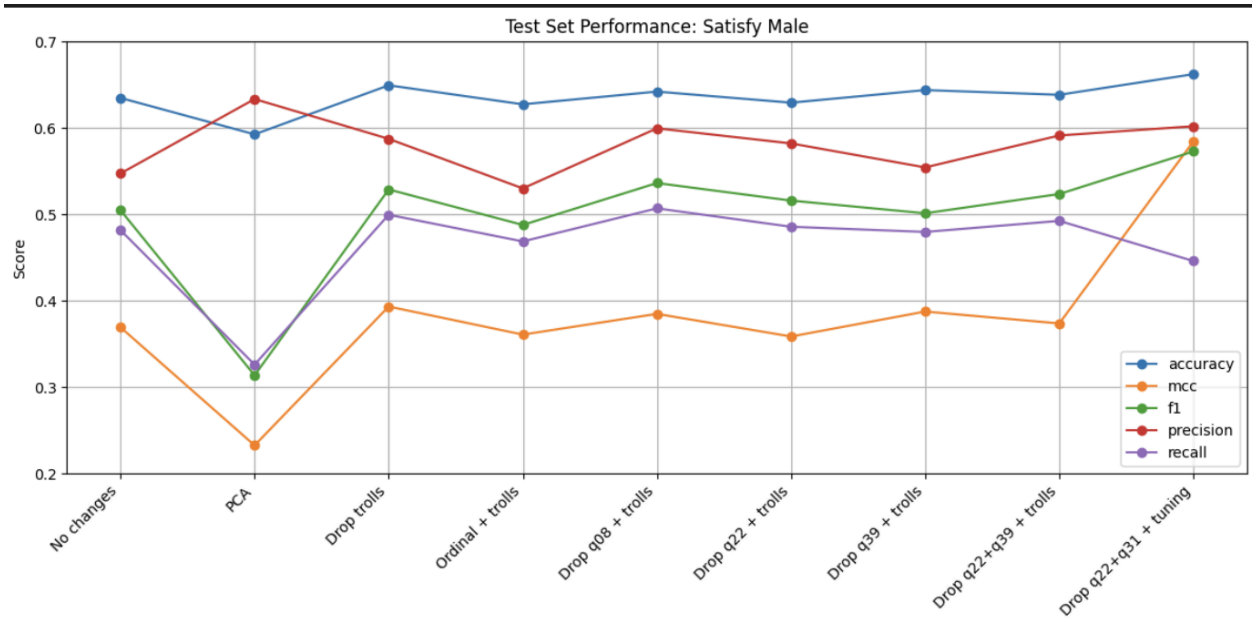
Shows performance of different models that went through different preprocessing steps across different metrics.

Figure 2. Cross Validation Set Performance for Satisfy Male



Shows performance of different models that went through different preprocessing steps across different metrics.

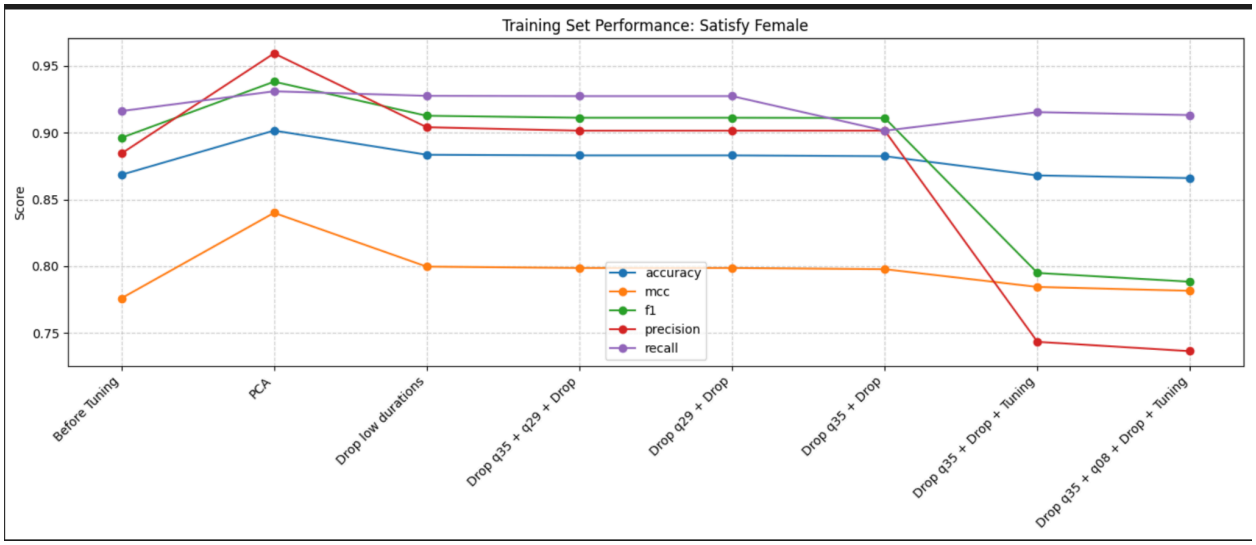
Figure 3. Test Set Performance for Satisfy Male



Shows performance of different models that went through different preprocessing steps across different metrics.

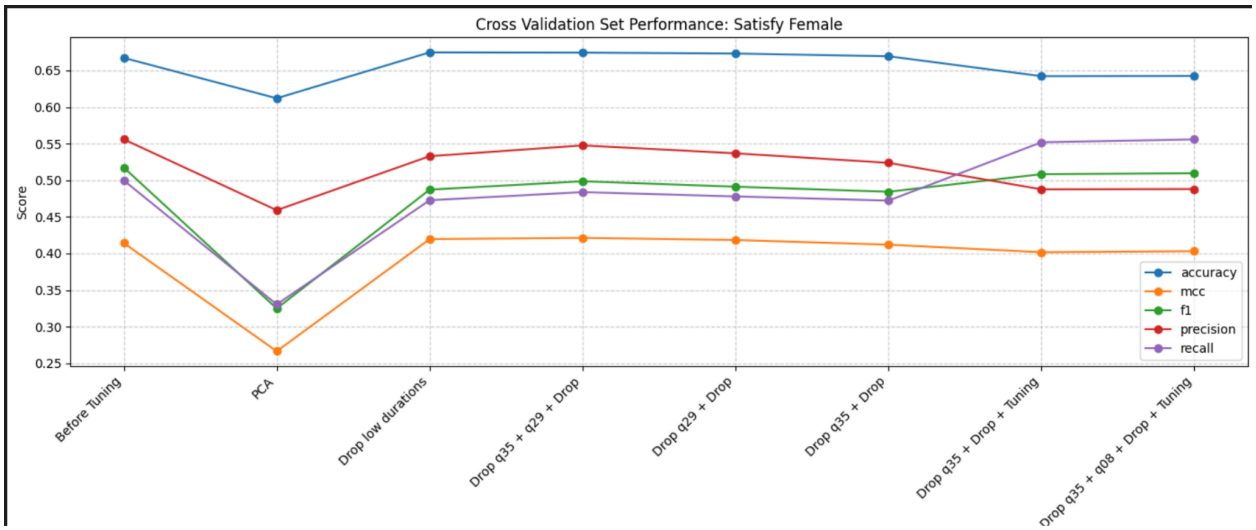
The best-performing model for Satisfy Male was the configuration that dropped q22 and q31 after hyperparameter tuning without dropping any instances. It achieved the highest balanced performance across cross-validation (F1 = 0.5324, MCC = 0.3893) and also performed well on the test set (F1 = 0.5322, MCC = 0.4029). In contrast, PCA showed inflated training scores but failed to generalize to the test set, confirming overfitting.

Figure 4. Training Set Performance for Satisfy Female



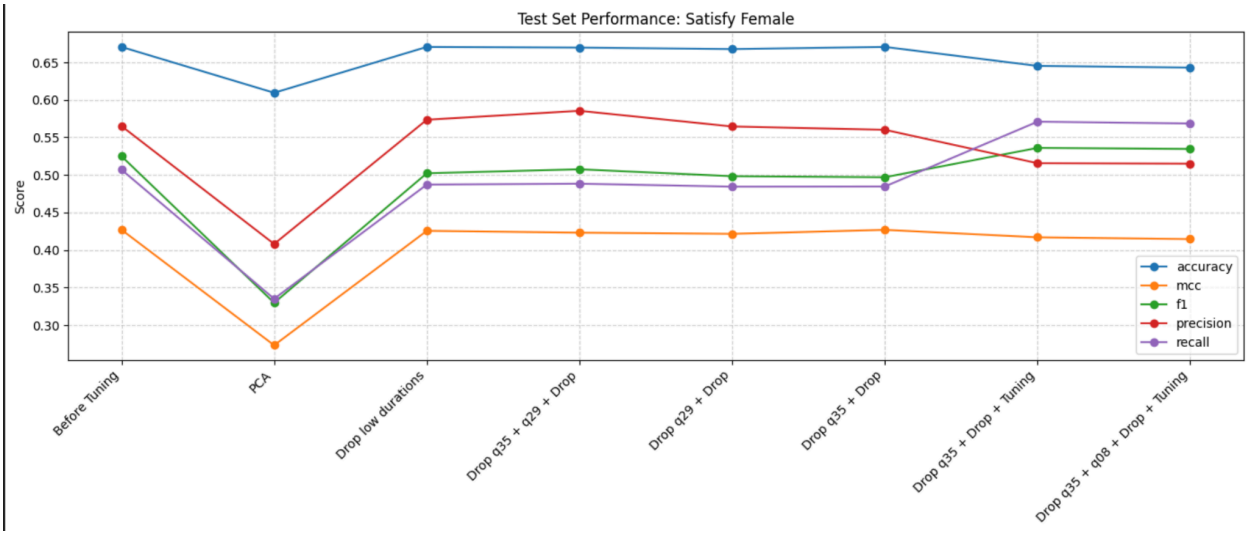
Shows performance of different models that went through different preprocessing steps across different metrics.

Figure 5. Cross Validation Set Performance for Satisfy Female



Shows performance of different models that went through different preprocessing steps across different metrics.

Figure 6. Test Set Performance for Satisfy Female



Shows performance of different models that went through different preprocessing steps across different metrics.

The best performing model for the Satisfy Female set was the one that dropped q35 and q08, filtered out low-duration responses, and included hyperparameter tuning. This configuration achieved the highest cross-validation F1 score of 0.5173, alongside a MCC score of 0.3965. These metrics indicate a well-balanced model in terms of both precision and recall. In contrast, PCA showed inflated training scores but failed to generalize, confirming overfitting.

Figure 7. ANOVA for Satisfy

```
ANOVA Results for Satisfy Models (Male vs Female)

Metric: accuracy
  Voting: F = 3.5458, p = 0.1087 → Not significant

Metric: mcc
  Voting: F = 0.2411, p = 0.6409 → Not significant

Metric: f1
  Voting: F = 0.5623, p = 0.4817 → Not significant

Metric: precision
  Voting: F = 2.7804, p = 0.1465 → Not significant

Metric: recall
  Voting: F = 1.3584, p = 0.2880 → Not significant
```

Shows that the Male and Female models don't have statistically significant differences.

Given the graphs for each model as well as the results from ANOVA, it seems like the effectiveness of the predictors does vary based on the gender of the employee in terms of predicting the answer to q61, but not to a statistically significant extent. In cross-validation, the best models for males and females had different optimal preprocessing steps. For example, the male model performed best after dropping q22 and q31, while the female model performed best after dropping q35, q08, and bad instances. Despite these differences, ANOVA results for Satisfy models showed no statistically significant differences across any metric. The overall behavior and performance of the models were similar.

We did not create graphs for the Fulfill models because that was outside the bounds of the question, but we still wanted to see how the predictors differed based on gender. We included a table that shows the differences in performance across different metrics for our final Female and Male models to predict fulfillment. In addition, you will find results from ANOVA to compare the two models.

We compared the final test performance of the male and female models and found that the male model outperformed the female model across all evaluation metrics. ANOVA shows that the models differ significantly in terms of accuracy and mcc. This indicates that gender meaningfully affects how well the model balances and generalizes fulfillment prediction. The male model achieved these results despite being trained on fewer samples. However, it has a high accuracy compared to its test scores, suggesting it is more prone to overfitting than the female model. This contrasts with the Satisfy models, where ANOVA showed no significant differences across gender.

Table 6. Test Performance for Fulfill Gender Models

Metric	Male (Test)	Female (Test)	Which Performs Better?
Accuracy	0.6220	0.6179	Male
F1 Score	0.5519	0.5257	Male
Precision	0.5763	0.5423	Male
Recall	0.5439	0.5207	Male
MCC	0.4390	0.3937	Male

Displays the test evaluations across metrics for gender based models.

Figure 8. ANOVA for Fulfill

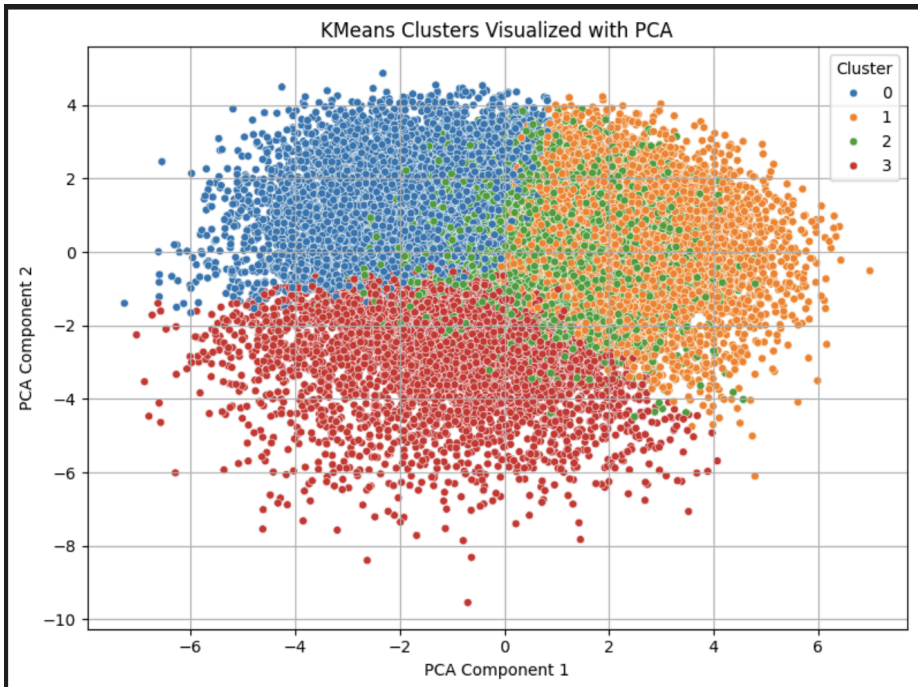
```
ANOVA Results for Fulfill Models (Male vs Female)

Metric: accuracy
  Voting: F = 39.3380, p = 0.0008 → Significant difference

Metric: mcc
  Voting: F = 9.6871, p = 0.0208 → Significant difference

Metric: f1
  Voting: F = 3.8751, p = 0.0965 → Not significant

Metric: precision
  Voting: F = 4.6041, p = 0.0756 → Not significant

Metric: recall
  Voting: F = 3.8458, p = 0.0975 → Not significant
```

Shows that the Male and Female models for Fulfill have statistically significant differences in regards to accuracy and mcc.

Extra Credit Two Report

To explore different subgroups within the Company.csv dataset, we applied KMeans clustering to the dataset based on the survey responses. We did some mild preprocessing. We computed a duration feature from the form_start and form_end timestamps to reflect how long each employee took to complete the form. Initially, we did not remove the "troll" durations. This was because we wanted to see which clusters the instances with wacky duration times belonged to. The columns with multiple selections were encoded with MultiLabelBinarizer so they could be used numerically. All the features were standardized using StandardScaler to ensure equal contribution during clustering. In total, we made 4 clusters directly on the scaled feature set. We did not make clusters using the components kept by PCA because we wanted the clusters to reflect interpretable variables. However, we did apply PCA separately with 2 components for visualization purposes, so we could plot and examine how the clusters were distributed in a two dimensional space.

Figure 1. KMeans Clusters Visualized with PCA



Displays four defined clusters.

As shown in the figure, the PCA graph reveals that cluster 0, 1, and 3 are separated along the principal component axes, with some overlap. Cluster 2 seems to be a little strange as it encompasses space within the bounds of the other clusters. This may suggest that Cluster 2 contains employees with more mixed responses.

After visualizing the clusters, we decided to group our dataset by the clusters we created. From there, we described features we wanted to know more about.

Analyzing Satisfaction and Fulfillment Across Clusters

Table 1. Describing Satisfaction and Fulfillment Across Clusters

	q61								q62							
	count	mean	std	min	25%	50%	75%	max	count	mean	std	min	25%	50%	75%	max
cluster																
0	7246.0	26.106542	44.820775	-100.0	33.0	33.0	33.0	100.0	7246.0	31.398427	59.173555	-100.0	33.0	33.0	100.0	100.0
1	6279.0	64.726071	38.776010	-100.0	33.0	100.0	100.0	100.0	6279.0	70.640707	45.196469	-100.0	33.0	100.0	100.0	100.0
2	5545.0	69.851758	36.885216	-100.0	33.0	100.0	100.0	100.0	5545.0	77.748061	39.367366	-100.0	33.0	100.0	100.0	100.0
3	4769.0	30.946320	49.030163	-100.0	33.0	33.0	33.0	100.0	4769.0	33.640176	60.987385	-100.0	33.0	33.0	100.0	100.0

Summarizes the central tendency and the spread of satisfaction and fulfillment scores across clusters.

Here, we are analyzing the distribution of satisfaction and fulfillment values. Clusters 1 and 2 show median scores of 100 and relatively lower standard deviations, indicating that most employees in these clusters are consistently satisfied and fulfilled. In contrast, the other clusters, despite having positive means, have median values at 33 and higher standard deviations. This means that there is more diversity of opinion for instances in those clusters. More people are strongly dissatisfied and unfulfilled. This is interesting, especially because the graph made using PCA showed Cluster 2 to be embedded within the other clusters. Cluster 2's behavioral distinction was preserved in the full feature space but not fully captured in the 2D space.

Table 2. Counts of Each Satisfaction/Fulfillment Scores per Cluster

```
q61 counts per cluster:
 q61      -100  -33    33    100
cluster
0          282  1224  4720  1020
1           28   190  2845  3216
2           15   112  2228  3190
3          225   701  2852   991

q62 counts per cluster:
 q62      -100  -33    33    100
cluster
0          584  1211  3272  2179
1          114   325  1766  4074
2           60   168  1329  3988
3          404   773  1983  1609
```

Displays frequency of score appearance within each cluster.

Table 2 confirms that Clusters 0 and 3 display more mixed sentiment rather than a fully negative one. While the dataset skewed positive overall, not all clusters are equally engaged or content. It might be difficult to predict for instances in clusters 0 and 3 because of the variability of their responses.

Analyzing Gender in Clusters

Table 3. Describing Gender Across Clusters

cluster	count	mean	std	min	25%	50%	75%	max
0	7246.0	1.934171	0.248001	1.0	2.0	2.0	2.0	2.0
1	6279.0	1.870999	0.335228	1.0	2.0	2.0	2.0	2.0
2	5545.0	1.924076	0.264901	1.0	2.0	2.0	2.0	2.0
3	4769.0	1.895366	0.306113	1.0	2.0	2.0	2.0	2.0

Describes the gender distribution for each cluster, where 1 = male and 2 = female.

According to Table 3, four clusters had average gender values near 2, suggesitng that most participants were female. We also confirmed this when we worked with the first extra credit question. The differences between the clusters in terms of gender were small, meaning that gender did not appear to be a major factor driving cluster separation.

Table 4. Describing Male Responses to q61 and q62 per Cluster

cluster	q61	q62
0	7.660377	11.767296
1	58.783951	60.330864
2	66.627078	69.358670
3	16.779559	18.306613

Shows the average satisfaction and fulfillment responses from male participants in each cluster.

Table 5. Describing Female Responses to q61 and q62 per Cluster

cluster	q61	q62
0	27.406412	32.781799
1	65.606144	72.167672
2	70.116706	78.437354
3	32.601874	35.432084

Shows the average satisfaction and fulfillment responses from female participants in each cluster.

We wanted to see if there were differences among males and females in terms of satisfaction/fulfillment in the same clusters. Table 4 and 5 show that females consistently score higher in both satisfaction and fulfillment across all clusters. The difference was especially pronounced in Cluster 0, where males reported lower scores.

Analyzing Duration in Clusters

Table 6. Describing Duration in Clusters

cluster	count	mean	std	min	25%	50%	75%	max
0	7246.0	6637.222882	114010.287606	48.0	392.0	552.5	883.0	8606730.0
1	6279.0	4918.476987	33752.726511	38.0	389.0	555.0	897.0	1900334.0
2	5545.0	8965.816231	173659.862279	82.0	423.0	624.0	1043.0	11284688.0
3	4769.0	6161.786538	44115.082235	38.0	364.0	523.0	864.0	1650948.0

Summarizes how long respondents in each cluster took to complete the survey in seconds.

Median completion time for all clusters hovered around 9-10 minutes while the mean durations were inflated by extreme outliers. The maximum times exceeded over 130 days. Cluster 2, which had the highest satisfaction and fulfillment scores,

showed the highest average and median duration. This may imply that Cluster 2 included more thoughtful individuals who took their time to think about the survey and the questions.

Analyzing Marital Status in Clusters

Table 7. Describing Marital in Clusters

cluster	count	mean	std	min	25%	50%	75%	max
0	7246.0	1.640767	0.479809	1.0	1.0	2.0	2.0	2.0
1	6279.0	1.646281	0.478161	1.0	1.0	2.0	2.0	2.0
2	5545.0	1.534175	0.498876	1.0	1.0	2.0	2.0	2.0
3	4769.0	1.486895	0.499881	1.0	1.0	1.0	2.0	2.0

Shows the marital status breakdown in each cluster, where 1 = not married and 2 = married.

Clusters 0, 1, and 2 had mean values above 1.5 and medians of 2.0, indicating that the majority of the employees in these groups were married, In contrast, cluster 3 was the only group with a median marital status of 1.0, suggesting that it contains a lot of unmarried employees. However, marital status does not drive cluster formation as the differences between the means aren't that big. All clusters also contain a mix of married and unmarried employees, so marriage is not a defining boundary for separation. Still, because the cluster with the most unmarried employees felt less fulfilled and satisfied overall, it suggests that marital status may influence how employees perceive their work experience.

Table 8. Average q61 and q62 by Marital Status per Cluster

cluster	q72	q61	q62
0	1	23.98	30.16
	2	27.30	32.09
1	1	64.35	70.11
	2	64.93	70.93
2	1	69.05	76.91
	2	70.55	78.48
3	1	30.35	31.08
	2	31.57	36.34

Breaks down satisfaction and fulfillment scores by marital status across clusters, where q72 = 1 represents unmarried and q72 = 2 represents married.

Table 8 indicates that married individuals consistently reported higher levels of satisfaction and fulfillment across all clusters. The difference was most pronounced in lower-performing clusters, where unmarried respondents had notably lower satisfaction and fulfillment. These results suggest that marriage can provide social and emotional support that improves feelings about the workplace.

Analyzing Financial Obligations in Clusters

Table 9. Describing Financial Obligations

cluster	count	unique	top	freq
0	7246	20	4	2560
1	6279	21	4	1950
2	5545	19	5	2248
3	4769	20	4	1631

Shows distribution of multiselect financial obligations reported by respondents in each cluster.

All clusters have high diversity in terms of financial obligations, which makes sense because different people have different circumstances. Clusters 0, 1, and 3 share 4 as the most common response, which represents other loans. However, cluster 2 has 5 as the most common response. 5 is ticking the mark Don't have. Cluster 2 also had higher fulfillment and satisfaction scores than the other clusters, indicating that not having a financial burden may be associated with greater workplace well-being, potentially due to reduced stress.

Analyzing Dependents in Clusters

Table 10. Describing Dependents

cluster	count	unique	top	freq
0	7246	24	2	2123
1	6279	21	2	1864
2	5545	20	6	1774
3	4769	22	6	1742

Displays the variety and frequency of the types of dependents people have in each cluster.

The results from this were interesting because it fits in well with the previous results we found. Cluster 2 and Cluster 3 had their top choice being no dependents, while the other clusters reported 1 child as their dependent. Again, Cluster 2 was associated with the best fulfillment and satisfaction scores. In the financial obligations table, Cluster 2 folks indicated that they had the least financial obligations out of the rest of the clusters. Based on this information, we can assume (and show later) that Cluster 2 instances are made up of younger people who have not yet started families. As they are in the beginning stages of their career, they may be supported by their families. On the other hand, Cluster 3 is made up of very different people despite having the same top choice in dependents (no dependents). Earlier, Cluster 3 was shown to have more instances that were unmarried than others. They also have small loans. In that case, the instances in Cluster 3 are most likely older people with less stability and support.

Analyzing Age in Clusters

Table 11. Describing Age

cluster	count	mean	std	min	25%	50%	75%	max
0	7246.0	5.225642	1.642436	1.0	4.0	5.0	6.0	9.0
1	6279.0	5.043319	1.514937	1.0	4.0	5.0	6.0	9.0
2	5545.0	4.577998	1.846691	1.0	3.0	5.0	6.0	9.0
3	4769.0	4.191445	1.732669	1.0	3.0	4.0	5.0	9.0

Summarizes the age distribution in each cluster, where age was encoded on a 1-9 scale.

Analyzing the age data challenged one of our earlier assumptions. Based on previous trends, we believed that Cluster 3 consisted of older individuals with no family life, but the data shows that Cluster 3 is the youngest group overall. Cluster 3 has more unmarried people as well as smaller loans overall, so people who are unsatisfied or unfulfilled in Cluster 3 may feel that way because they are early in

their career with student loans. On the other hand, Cluster 2 included the second youngest group of people. The people in this Cluster are more likely more established and secure young people. Cluster 0 and 1 make up older people with Cluster 0 including more older people. However, the means are really similar for all the clusters, indidcating that the clusters were not created based on age.

Analyzing Position in Clusters

Table 12. Describing Position

cluster	count	mean	std	min	25%	50%	75%	max
0	7246.0	2.553409	0.506525	1.0	2.0	3.0	3.0	3.0
1	6279.0	2.099060	0.466867	1.0	2.0	2.0	2.0	3.0
2	5545.0	2.739766	0.442894	1.0	2.0	3.0	3.0	3.0
3	4769.0	2.544978	0.517034	1.0	2.0	3.0	3.0	3.0

Shows how job levels are distributed across clusters, where 1 = supervisor, 2 = director, and 3 = ordinary employee.

Clusters 0, 2, and 3 were made up primarily of ordinary employees. while Cluster 1 had more individuals in director roles. Interestingly, Cluster 2 had the highest proportion of ordinary employees, but also the highest satisfaction and fulfillment scores. Meanwhile, Cluster 1, which had more employees in higher level roles, also reported higher satisfaction. Logically, that would make sense, as senior positions come with more stability and wealth. However, it seems that job title does not impact fulfillment or satisfaction as much as other more emotional factors.

Conclusions

Through KMeans clustering, we were able to uncover some meaningful subgroups within the workforce that reflect structural, emotional, and behavioral differences. Cluster 2 consistently emerged as the most positive group, showing the highest levels of satisfaction and fulfillment. Despite being composed of ordinary employees, this group also had fewer financial obligations, less dependents, and a

larger number of unmarried individuals when compared to Clusters 0 and 1. On top of that, they were one of the youngest clusters. This suggests that Cluster 2 may represent young, early career employees who are more stable than a fresh new graduate. Cluster 3 was unique, because although it was similar to Cluster 2 in terms of age and dependents, its members reported much lower fulfillment and satisfaction. This group included the most unmarried individuals, small loans, and ordinary employees. This suggests that younger employees who are just starting to enter the workforce may experience more workplace related stress. Cluster 1 contained more directors and older employees, and also reported high satisfaction and fulfillment. This could be because of stable marriages as well as increased workplace control. Cluster 0, although similarly aged, showed lower fulfillment. Cluster 0 also had a good amount of ordinary employees, so older individuals who hadn't made it in terms of careers could potentially feel worse about the work they do. Across all clusters, we observed: married individuals reported higher satisfaction and fulfillment, those without financial burdens felt more positively about work, gender/age did not drive cluster separation, and job level had a mild correlation with satisfaction, but other factors mattered more. Workplace sentiment is not determined by any one variable, but the intersection of multiple aspects of people's lives. These complex intersections could be picked up by models, which is why reducing the number of columns or degrading the original features often degrades model performance. In addition, using PCA to reduce dimensionality lost important feature interactions.