DA 514 - Machine Learning I Class Project

Ali Enver Arslan

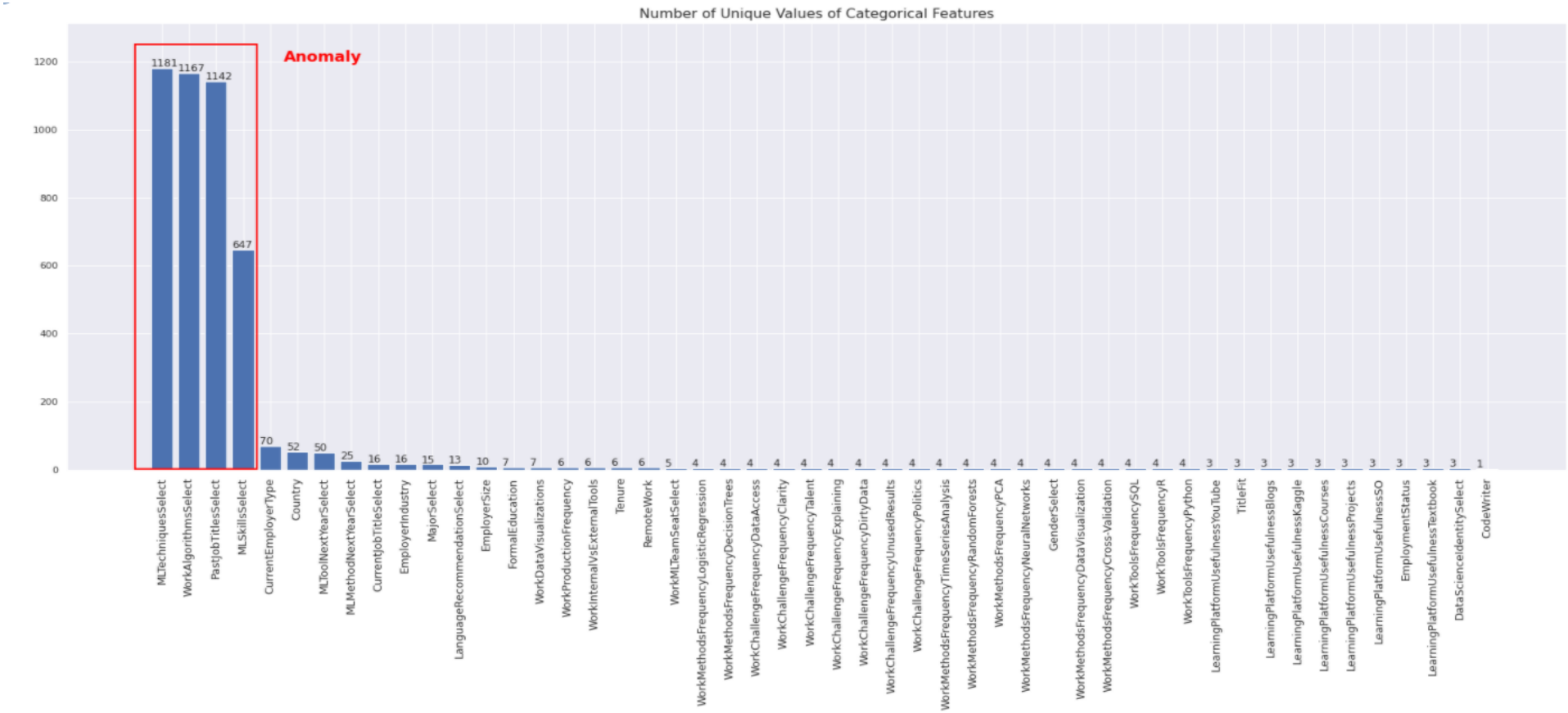
Cem İshakoğlu

Tufan Keser

Yücel Oğuz

TABLE OF CONTENTS

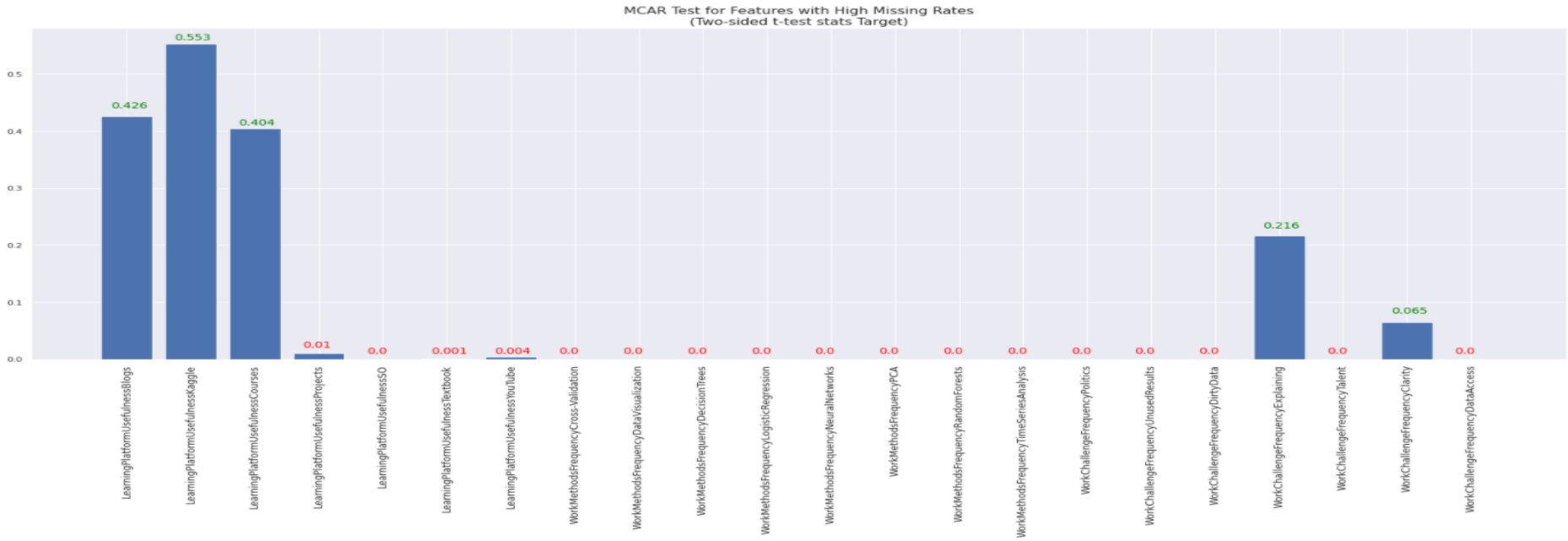
FEATURES



As it is seen above the graph; MLTechniquesSelect, WorkAlgorithmSelect, PastJobTitleSelect and MLSkillsSelect features have much more unique values than the other categorical variables. The number of unique values for MLTechniquesSelect is 14, 15 for workalgorithmselect, 13 for MLskillsselect, 8 for currentemployertype and 16 for pastjobtitlesselect.

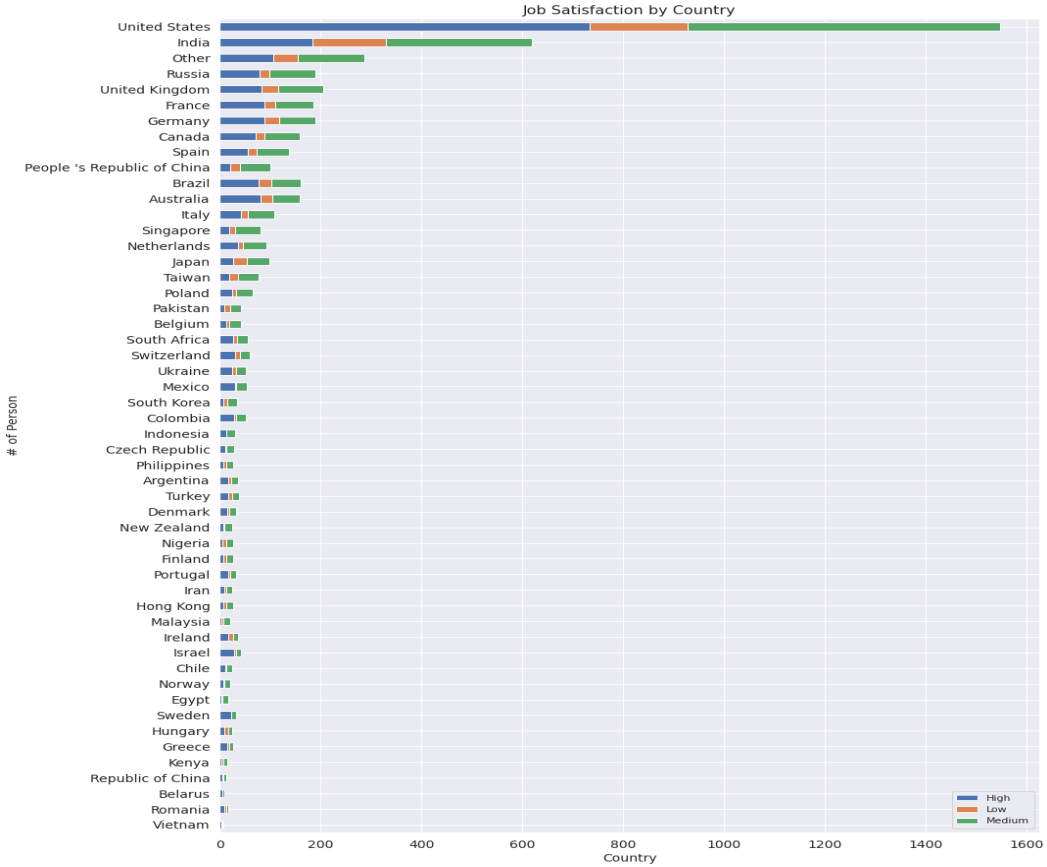
MISSING DATA ANALYSIS

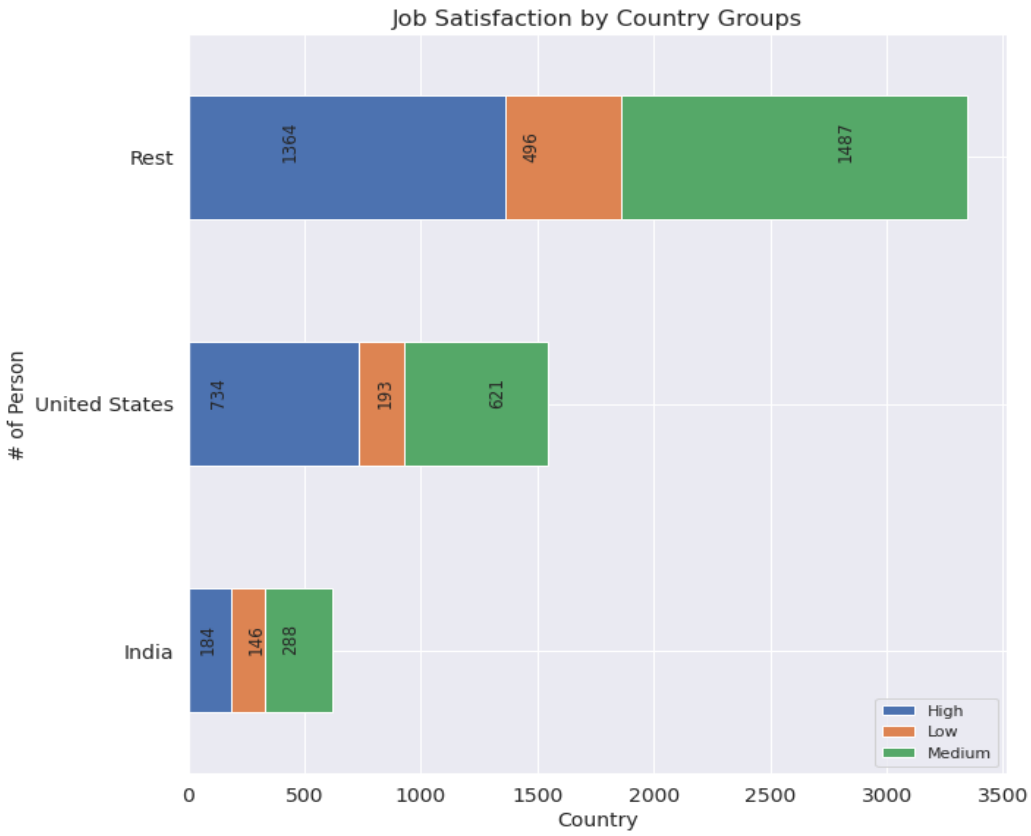
LearningPlatformUsefulness, WorkMethodsFrequency and WorkChallengeFrequency are features that have much more missing values than the other variables. The missingness is analyzed for understanding whether it is MCAR or not by using two sample t-test. The results of the t-test is shown in the following table.



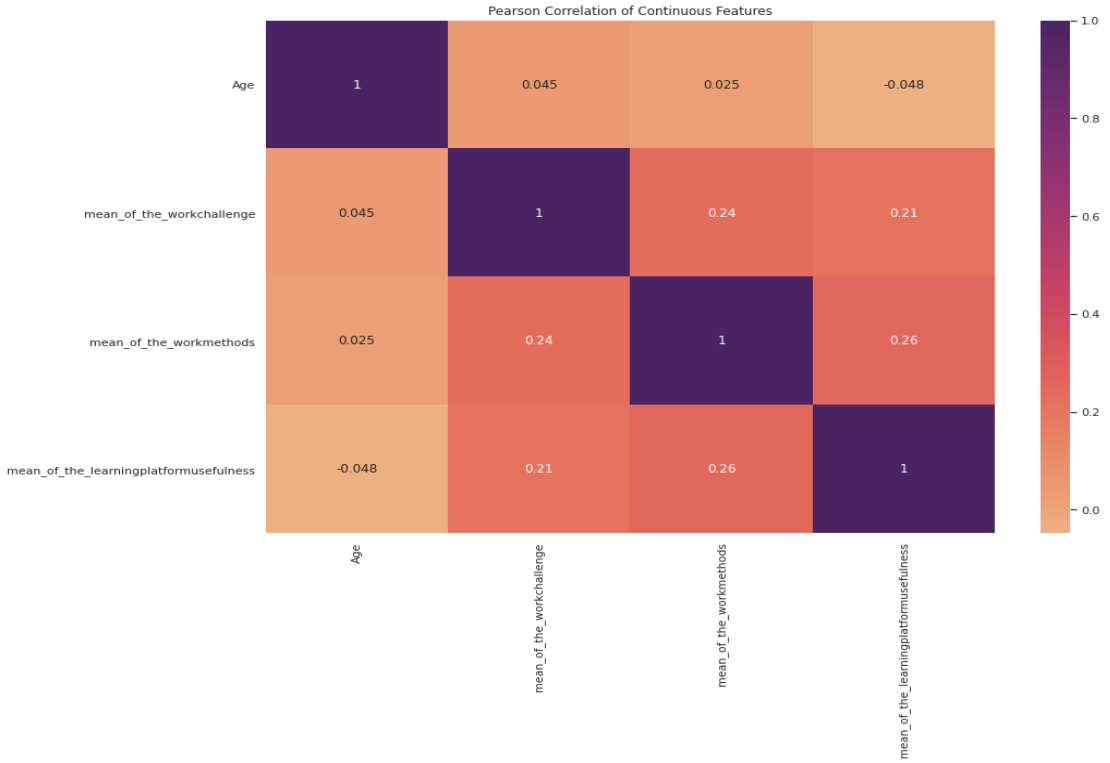
From the table, it can be concluded that the missingness of LearningPlatformUsefulnessBlogs, LearningPlatformUsefulnessKaggle, LearningPlatformUsefulnessCourses, WorkChallengeFrequencyExplaining and WorkChallengeFrequencyClarity is completely at random. There is a possibility of being non MCAR for the features that shown above graph with red.

EXPLORATORY DATA ANALYSIS





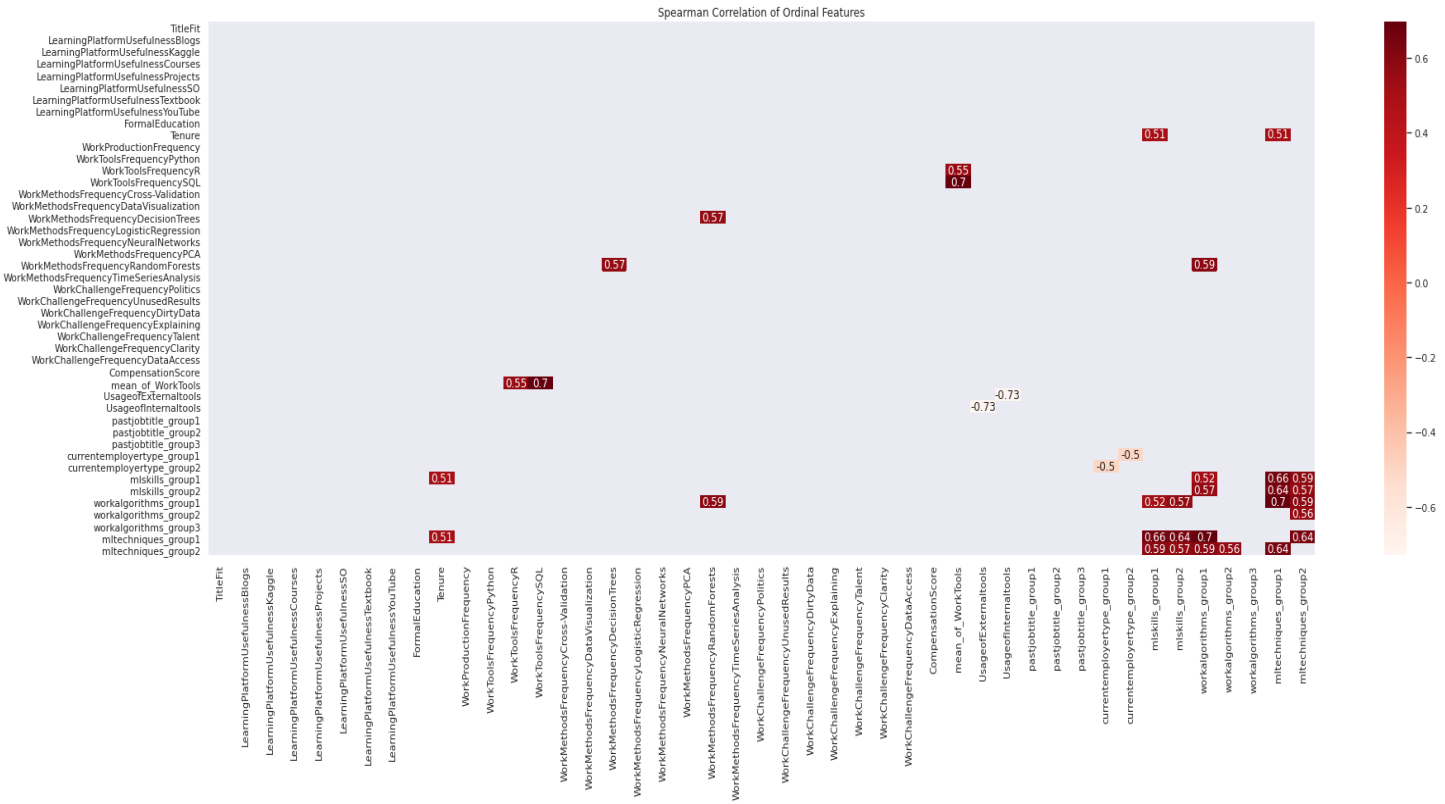
The country, a employee currently lives in, and the title that describes what he/she does has a strong relationship with target variable (Jobsatisfaction). The relationship between job satisfaction level and the country of an employee is summarized in the graphs above. (Further details are placed in the notebook.) Instead of using one hot encoder for making more columns, we choose to show the continents of the countries in order to have less dimensional model.



Spearman Correlation of Continuous Features with the Target

Features	JobSatisfaction
JobSatisfaction	1.000000
mean_of_the_workmethods	0.169169
Age	0.061270
mean_of_the_learningplatformusefulness	0.047625
mean_of_the_workchallenge	-0.119841

Pearson correlation is used between continuos variables. Since the target is ordinal (categorical), we used Spearman correlation to analyze between the target and continuous variables. The highest correlation with target belongs to mean of the work methods which is a generated feature. Also, it can be said that there is no highly correlated continuous feature among themselves.



Correlation of Ordinal Features with the Target

	JobSatisfaction
JobSatisfaction	1.000000
TitleFit	0.179047
currentemployertype_group1	0.177325
workalgorithms_group1	0.155274
WorkMethodsFrequencyCross-Validation	0.148395
WorkProductionFrequency	0.129368
mltechniques_group1	0.126208
pastjobtitle_group1	0.119961
WorkMethodsFrequencyNeuralNetworks	0.113319
WorkMethodsFrequencyRandomForests	0.105850
WorkMethodsFrequencyPCA	0.105013

Cramer's V Correlation of Binary Variables with the Target

```
continent_North America : 0.0828122769635807
continent_Asia : 0.14118065680114458
WorkMLTeamSeatSelect_Other : 0.05342566841120866
WorkMLTeamSeatSelect_IT Department : 0.07828152047913041
RemoteWork_Sometimes : 0.05189571440367688
RemoteWork_Never : 0.12326298483634304
RemoteWork_Most of the time : 0.057933414840916964
EmployerSize_Missing : 0.05323498527611985
EmployerIndustry_Government : 0.05612200897461242
DataScienceIdentitySelect_Unanswered : 0.1382646979847856
DataScienceIdentitySelect_Sort of (Explain more) : 0.057651014485669616
```

Spearman correlation is used between ordinal variables. Spearman correlation among ordinal features (greater than 0.5 and lower than -0.5) is shown in the chart above. TitleFit, currentemployertype_group1 (generated feature) have the highest correlation with the target. Phi's coefficient is used between binary variables. Cramer's V correlation is used between binary variables and the target.

FEATURE ENGINEERING

In order to complete EDA and feature extraction process we create a class in which all of the features in the model is either encoded or mapped with the labels, there is no step in the class which can cause data leakage.

```
1 pipeline_feature_generation = Pipeline([('feature generator', imputer_and_generator.__main__())])
```

FEATURE SELECTION

- The columns that contain more than 30% null values are dropped.
- The rows that contain more than 30% null values are dropped.
- The features that has the highest and the lowest correlation with the dependent variable are chosen then added polynomial and trigonomial combinations of these variables for feature selection.

MODEL BUILDING

- **Initial model** : After completing the EDA process we want to analyze the result of the all of the supervised machine learning algorithms for classification problem.

Model	Accuracy	Balanced Accuracy	ROC AUC	F1 Score	Time Taken
NearestCentroid	0.41	0.43	None	0.40	0.05
BernoulliNB	0.47	0.42	None	0.47	0.05
AdaBoostClassifier	0.50	0.42	None	0.48	0.55
GaussianNB	0.44	0.41	None	0.44	0.05
LogisticRegression	0.51	0.41	None	0.48	0.31
LinearDiscriminantAnalysis	0.50	0.41	None	0.48	0.14
LGBMClassifier	0.49	0.41	None	0.47	0.74
CalibratedClassifierCV	0.51	0.40	None	0.47	15.17
RidgeClassifierCV	0.51	0.40	None	0.47	0.12
RandomForestClassifier	0.50	0.40	None	0.47	1.04
XGBClassifier	0.48	0.40	None	0.46	2.13
LinearSVC	0.51	0.40	None	0.47	4.08
RidgeClassifier	0.51	0.40	None	0.47	0.07
SVC	0.49	0.39	None	0.45	4.11
ExtraTreesClassifier	0.49	0.39	None	0.45	1.09
BaggingClassifier	0.44	0.37	None	0.43	0.60
NuSVC	0.45	0.37	None	0.43	4.68
KNeighborsClassifier	0.43	0.36	None	0.42	1.30
ExtraTreeClassifier	0.40	0.36	None	0.40	0.05
DecisionTreeClassifier	0.41	0.36	None	0.42	0.12
PassiveAggressiveClassifier	0.38	0.36	None	0.38	0.12
SGDClassifier	0.42	0.36	None	0.42	0.44
Perceptron	0.40	0.34	None	0.39	0.07
LabelSpreading	0.16	0.33	None	0.07	1.80
LabelPropagation	0.16	0.33	None	0.07	1.58
QuadraticDiscriminantAnalysis	0.41	0.33	None	0.24	0.17
DummyClassifier	0.38	0.33	None	0.38	0.05

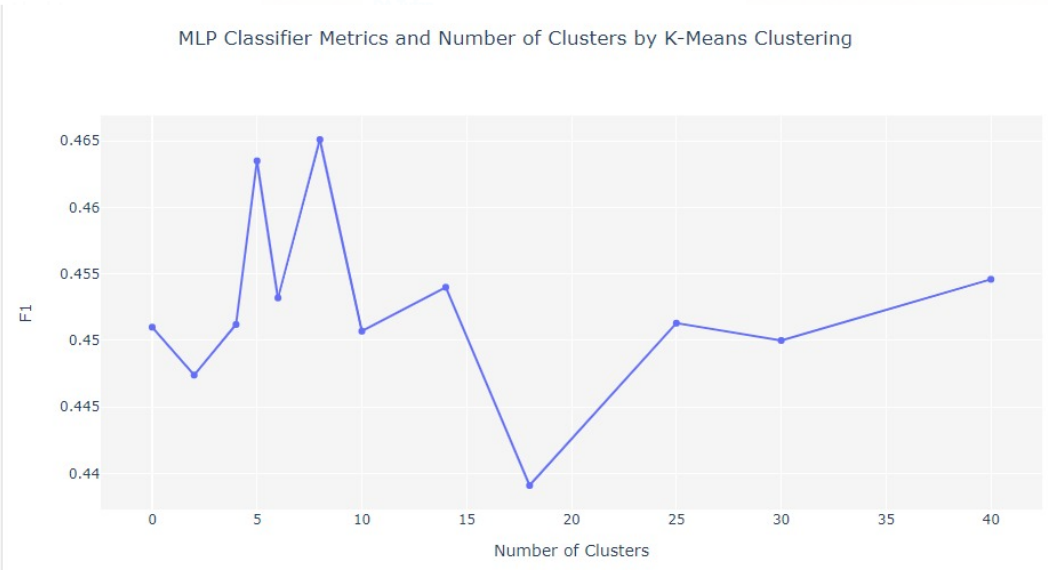
- **Second model** : After analyzing the initial model, we want to tunning for gradient boosting classifier with the parameters given below.

```
params = {'max_depth': [2, 3, 4, 5, None],
          "learning_rate" : [0.05,0.30],
         "n_estimators": [10,90,150]}

model = GradientBoostingClassifier(random_state=seed)

gridSearch2 = GridSearchCV(estimator = model, param_grid = params, scoring='f1_micro', n_jobs=-1)
gridSearch2.fit(X_train,y_train)
print('Best parameters:', gridSearch2.best_params_)
print('Best CV score  :', gridSearch2.best_score_)
```

```
Best parameters: {'learning_rate': 0.05, 'max_depth': 2, 'n_estimators': 90}
Best CV score  : 0.5236411992965833
```

- **Third model** : According to f1 score of the supervised learning algorithms tried, there is no more than 0.55 f1 score. So k-means, a unsupervised clustering method, is used to classify job satisfaction of the Kagglers with the supervised estimator multilayer perceptron.

	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.4552	0.5817	0.4071	0.4513	0.4510	0.1098	0.1103
2	0.4496	0.5758	0.4057	0.4509	0.4474	0.1048	0.1056
4	0.4560	0.5803	0.4036	0.4523	0.4512	0.1062	0.1071
5	0.4666	0.5901	0.4159	0.4656	0.4635	0.1283	0.1290
6	0.4536	0.5895	0.4124	0.4574	0.4532	0.1175	0.1183
8	0.4701	0.5910	0.4195	0.4643	0.4651	0.1289	0.1296
10	0.4551	0.5803	0.4111	0.4521	0.4507	0.1123	0.1130
14	0.4574	0.5831	0.4092	0.4541	0.4540	0.1120	0.1125
18	0.4428	0.5727	0.3972	0.4421	0.4391	0.0919	0.0927
25	0.4547	0.5793	0.4054	0.4525	0.4513	0.1088	0.1095
30	0.4523	0.5819	0.4052	0.4532	0.4500	0.1093	0.1102
40	0.4554	0.5831	0.4117	0.4586	0.4546	0.1174	0.1180



- **Final model** : The final model is Naive Bayes with the 0.5319 f1 score.

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
lightgbm	Light Gradient Boosting Machine	0.5419	0.6732	0.4640	0.5299	0.5272	0.2282	0.2308	13.3720
gbc	Gradient Boosting Classifier	0.5300	0.6495	0.4446	0.5140	0.5108	0.2034	0.2066	18.2320
nb	Naive Bayes	0.5251	0.6917	0.5258	0.5599	0.5319	0.2689	0.2748	1.1760
lda	Linear Discriminant Analysis	0.5183	0.6704	0.4948	0.5331	0.5234	0.2381	0.2392	2.5220
ridge	Ridge Classifier	0.5181	0.0000	0.4963	0.5342	0.5235	0.2390	0.2403	1.2340
rf	Random Forest Classifier	0.5031	0.6212	0.4165	0.4863	0.4818	0.1543	0.1572	2.0560
et	Extra Trees Classifier	0.5026	0.6220	0.4165	0.4864	0.4811	0.1528	0.1557	2.3140
lr	Logistic Regression	0.4894	0.6443	0.4715	0.5085	0.4956	0.1978	0.1993	8.9100
ada	Ada Boost Classifier	0.4661	0.5833	0.4232	0.4633	0.4628	0.1337	0.1344	2.5000
qda	Quadratic Discriminant Analysis	0.4318	0.5054	0.3399	0.4127	0.3912	0.0115	0.0123	3.7240
dt	Decision Tree Classifier	0.4207	0.5324	0.3800	0.4229	0.4215	0.0660	0.0660	1.5420
svm	SVM - Linear Kernel	0.3930	0.0000	0.3927	0.4508	0.3723	0.0812	0.0905	1.9540
knn	K Neighbors Classifier	0.3207	0.5354	0.3678	0.4146	0.3275	0.0422	0.0484	3.9520