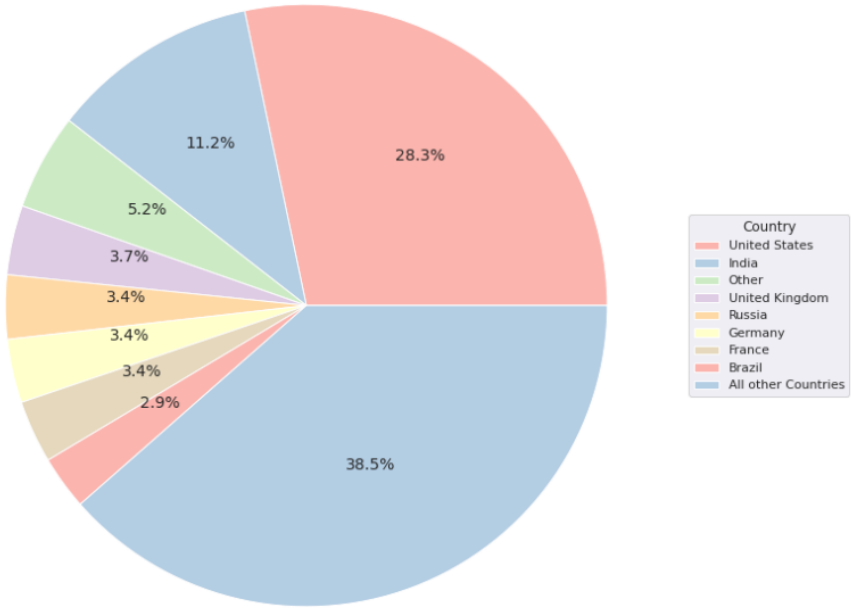
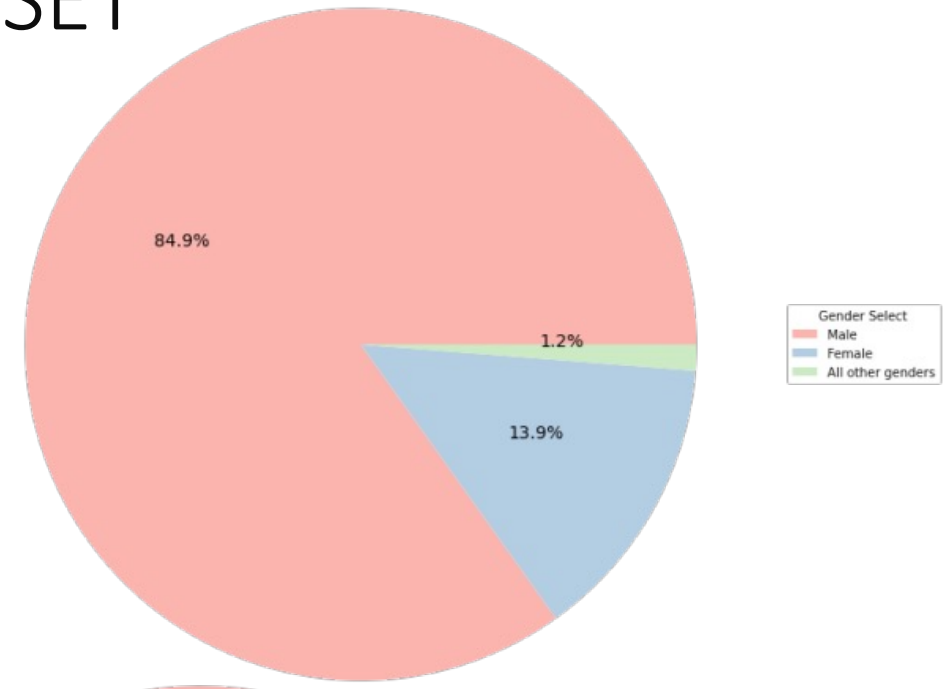
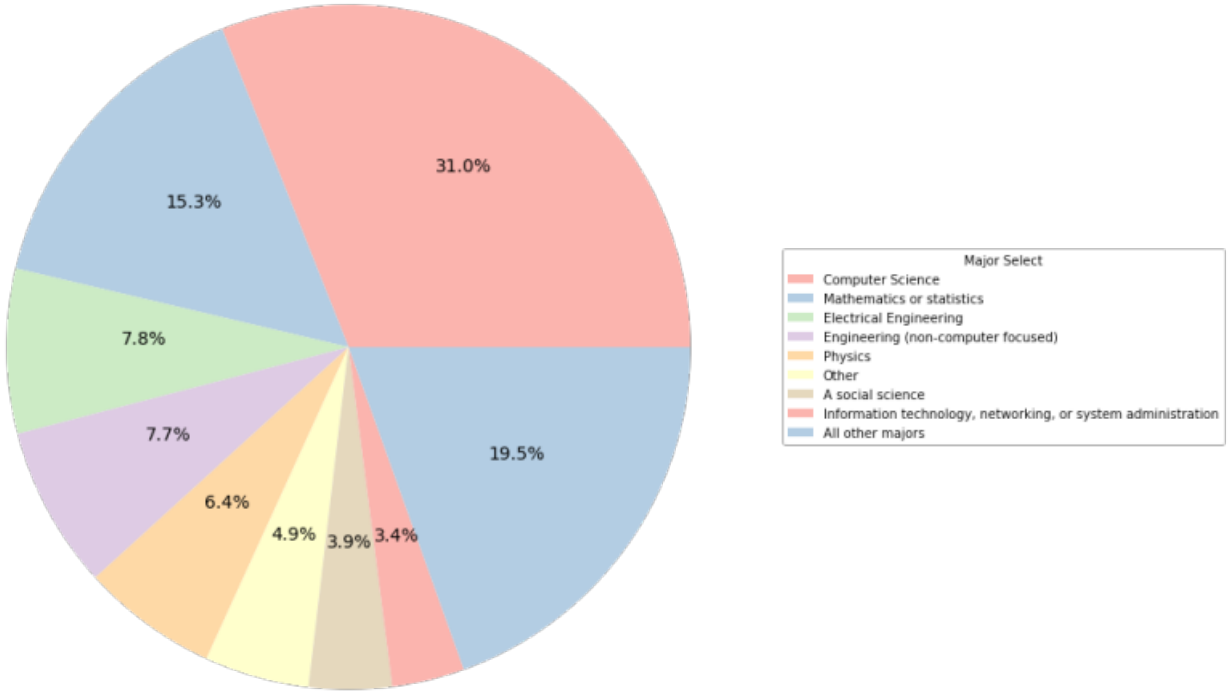
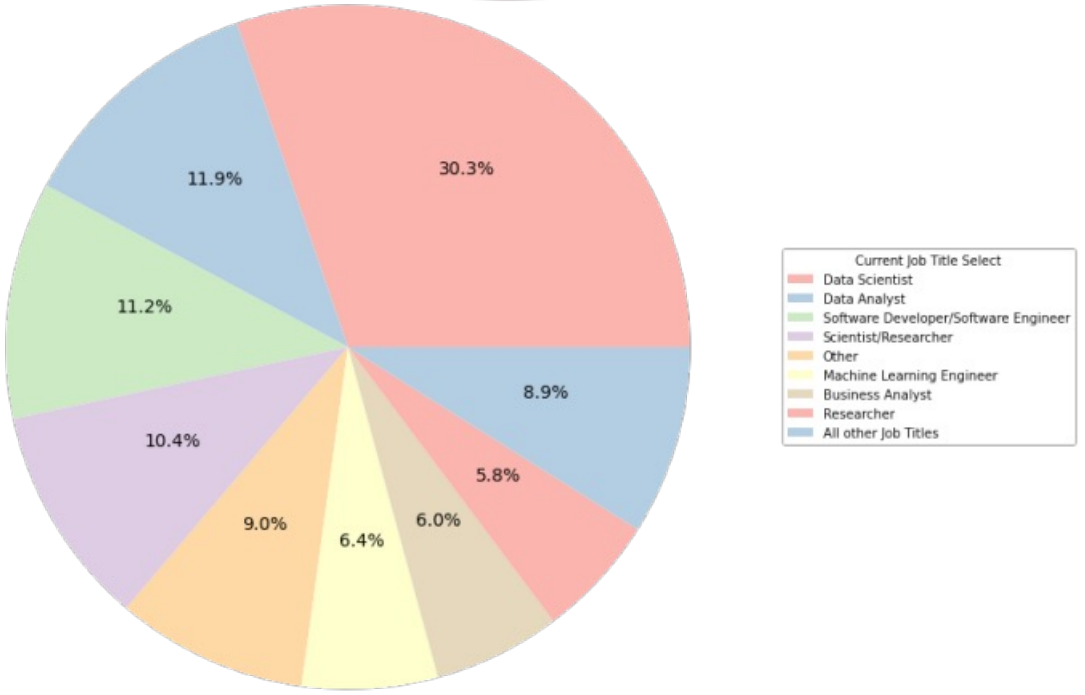
EXPLORING THE DATASET

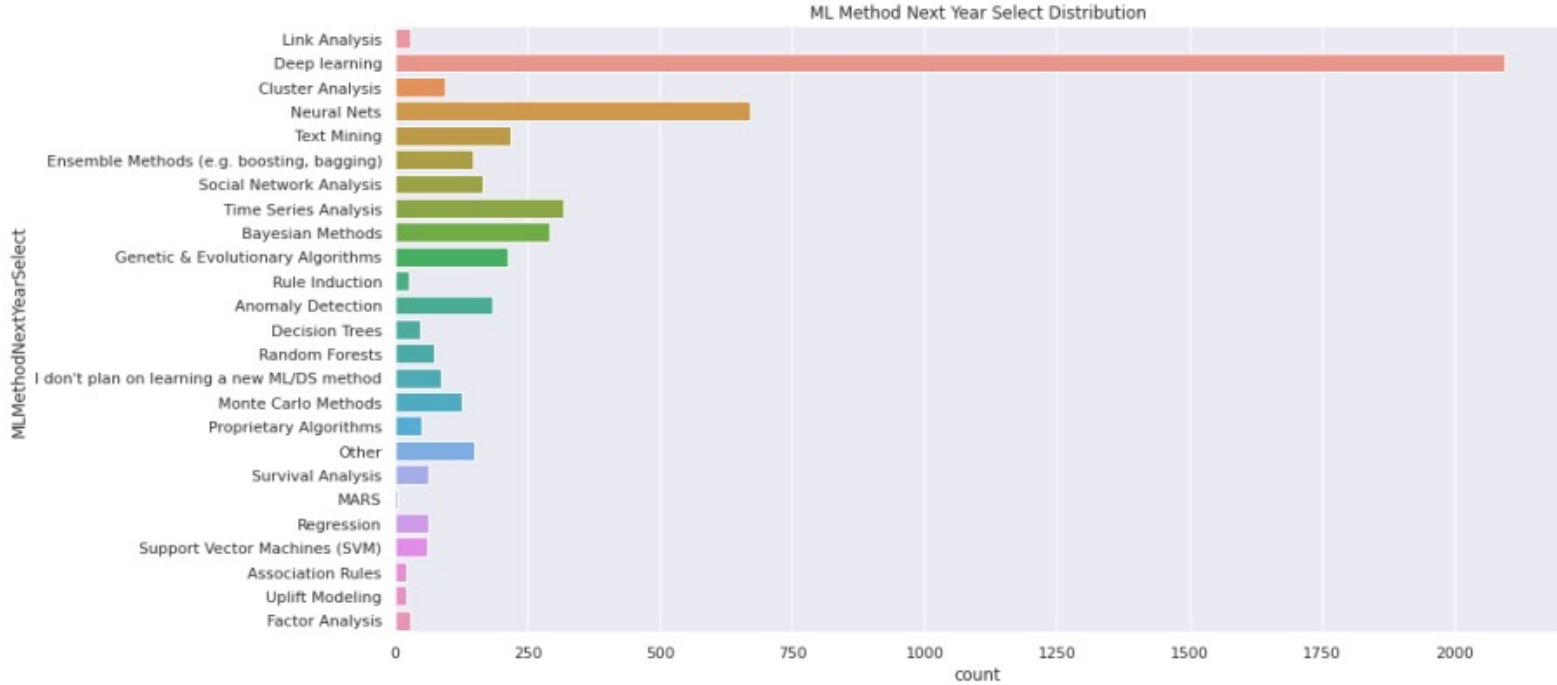
- The dataset contains 54 columns and 5530 observations
- There are 42 categorical and 4 numerical variables including the target variable which is job satisfaction

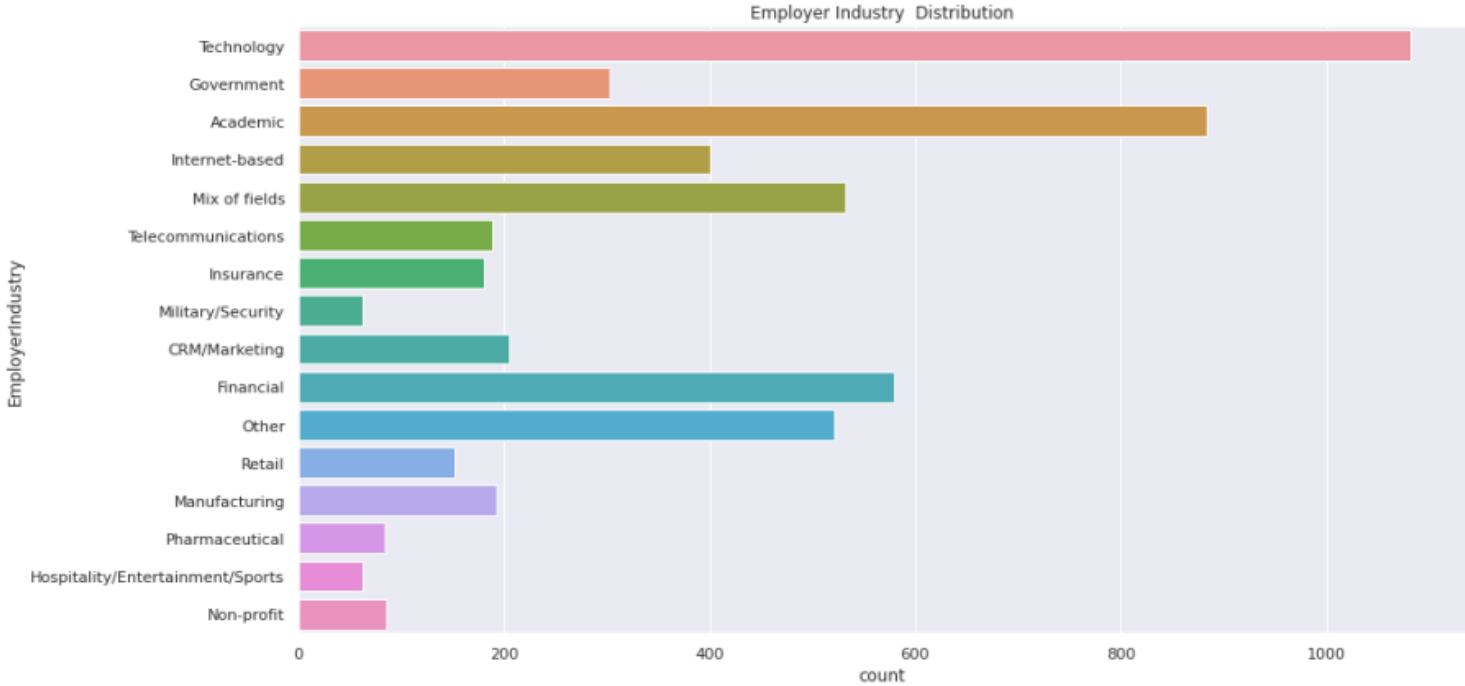




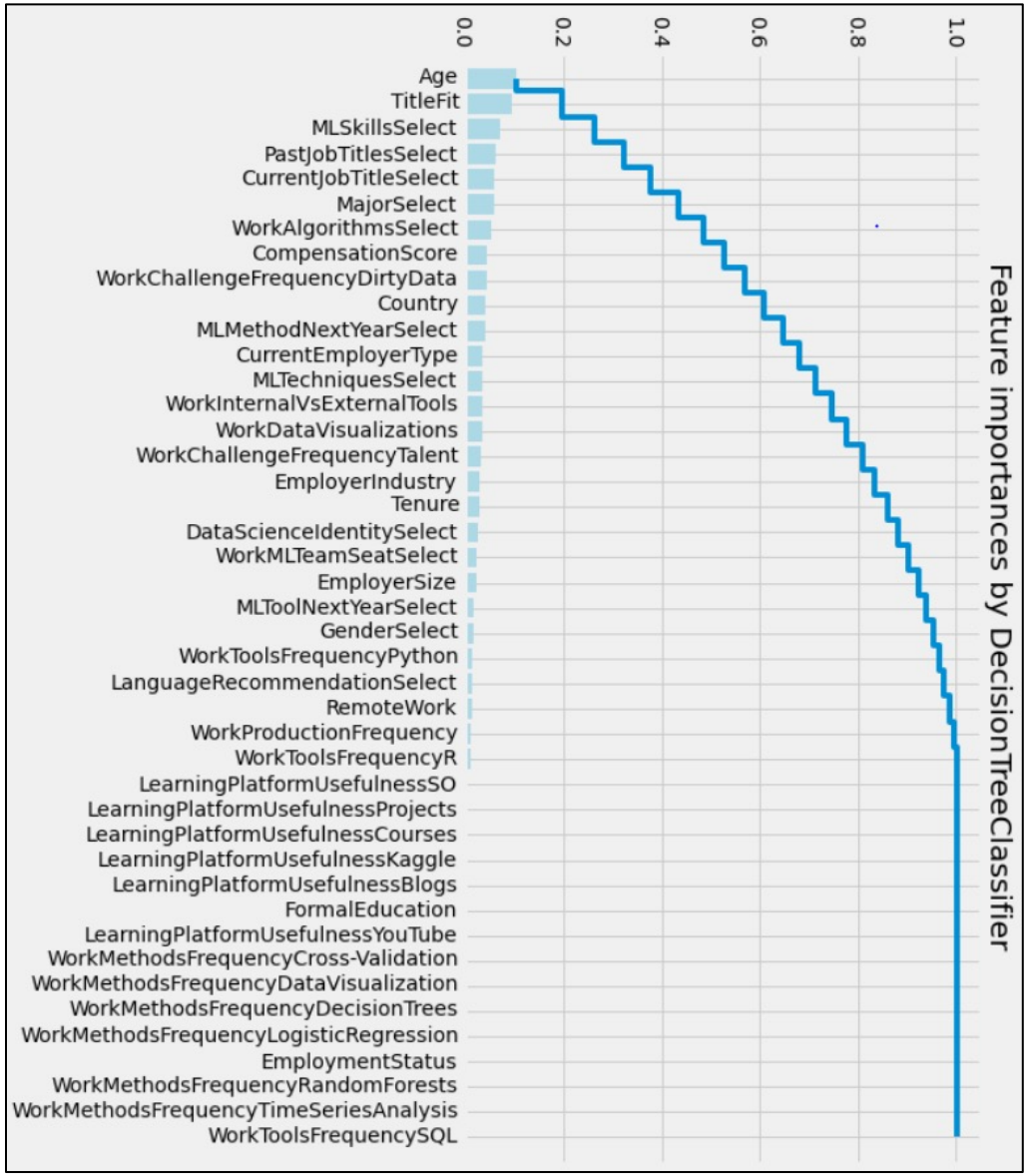




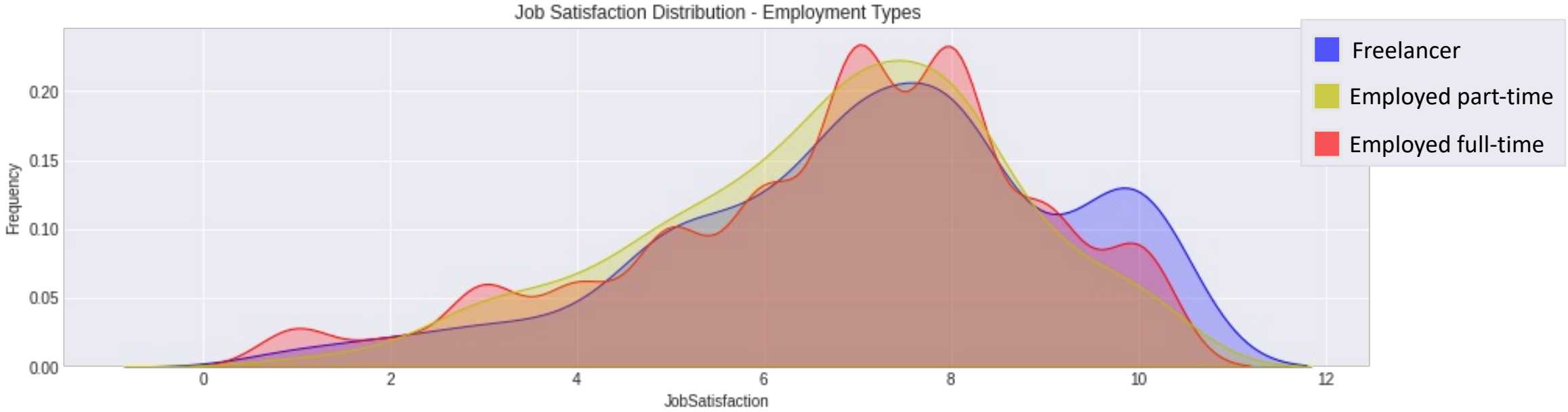

ML Method Next Year Select Distribution
MLMethodNextYearSelect
Link Analysis
Deep learning
Cluster Analysis
Neural Nets
Text Mining
Ensemble Methods (e.g. boosting, bagging)
Social Network Analysis
Time Series Analysis
Bayesian Methods
Genetic & Evolutionary Algorithms
Rule Induction
Anomaly Detection
Decision Trees
Random Forests
I don't plan on learning a new ML/DS method
Monte Carlo Methods
Proprietary Algorithms
Other
Survival Analysis
MARS
Regression
Support Vector Machines (SVM)
Association Rules
Uplift Modeling
Factor Analysis
0
250
500
750
1000
1250
1500
1750
2000
count

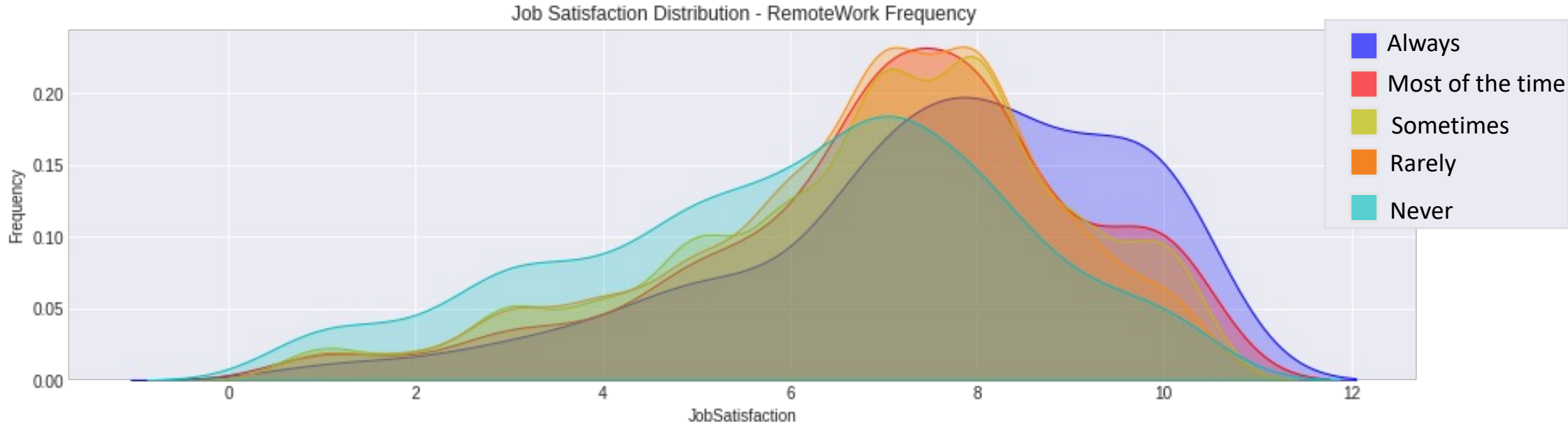

Employer Industry Distribution
EmployerIndustry
Technology
Government
Academic
Internet-based
Mix of fields
Telecommunications
Insurance
Military/Security
CRM/Marketing
Financial
Other
Retail
Manufacturing
Pharmaceutical
Hospitality/Entertainment/Sports
Non-profit
0
200
400
600
800
1000
count

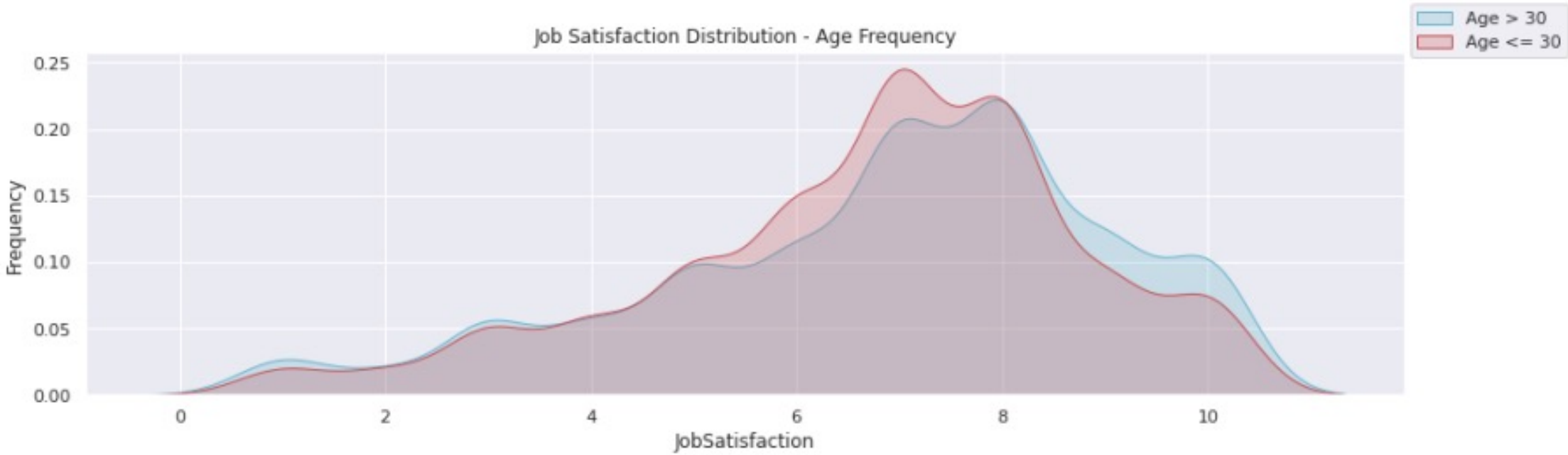
FEATURE IMPORTANCES

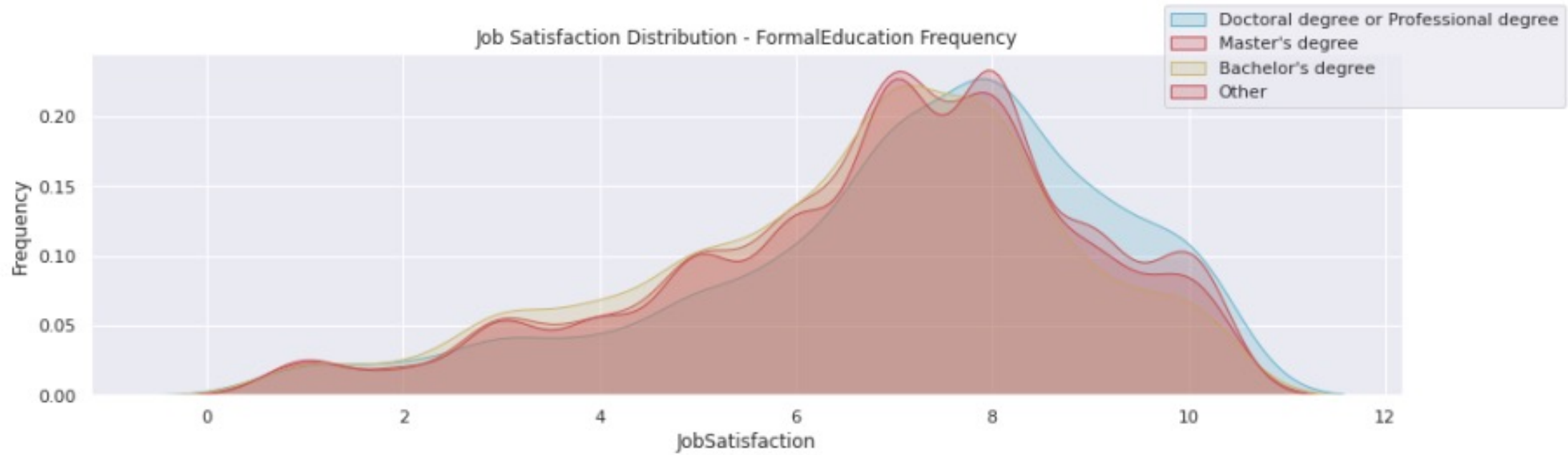


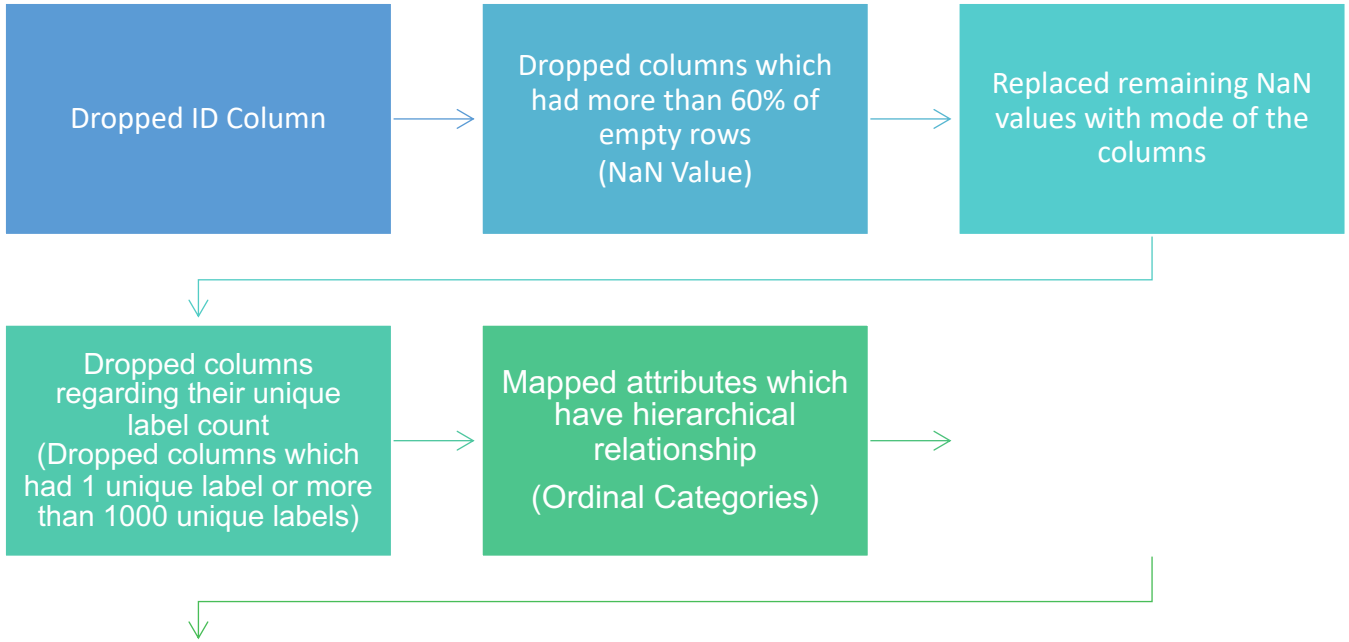
- **Age**
- **Title Fit**
  - ✓ How well do you feel your title suits you
- **ML Skills Select**
- **Past Job Title Select**
  - ✓ Last 10 years past job titles
- **Current Job Title Select**
- **Major Select**
  - ✓ University Major
- **Work Algorithm Select**
  - ✓ What algorithms do you use in your job?
- **Compensation Score**
- **Work Challenge Frequency Dirty Data**
- **Country**


Job Satisfaction Distribution - Employment Types
Freelancer
Employed part-time
Employed full-time
Frequency
0.00
0.05
0.10
0.15
0.20
0
2
4
6
8
10
12
JobSatisfaction


Job Satisfaction Distribution - RemoteWork Frequency
Always
Most of the time
Sometimes
Rarely
Never
Frequency
0.00
0.05
0.10
0.15
0.20
0
2
4
6
8
10
12
JobSatisfaction


Job Satisfaction Distribution - Age Frequency
Age > 30
Age <= 30
Frequency
0.25
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
JobSatisfaction


Job Satisfaction Distribution - FormalEducation Frequency
Doctoral degree or Professional degree
Master's degree
Bachelor's degree
Other
Frequency
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
12
JobSatisfaction


Dropped ID Column
Dropped columns which had more than 60% of empty rows (NaN Value)
Replaced remaining NaN values with mode of the columns
Dropped columns regarding their unique label count (Dropped columns which had 1 unique label or more than 1000 unique labels)
Mapped attributes which have hierarchical relationship (Ordinal Categories)

ML Models

Machine Learning Models Utilized

Gradient Boost Regressor

RMSE is 2.0131283795394395

Random Forest Regression

RMSE is 2.029846419720492

K-Neighbors Regressor

RMSE is **2.0644345761356817**

SVC (Support Vector Classifier)

RMSE is **2.2453504943475955**

Even though the problem we are facing is a regression problem we wanted to try SVC to test its performance

ML Models

Gradient Boosting Regressor

Hyperparameter Tuning

- Hyperparameter tuning is used to set the value of parameters that the algorithm cannot learn on its own.

  There are multiple hyperparameters we need to arrange

  -Number of Estimators; Number of estimators is the number of boosting stages to be performed by the model.

  -Maximum Depth; The maximum depth is the depth of the decision tree estimator in the gradient boosting regressor.

  -Learning Rate; The learning rate is a hyper-parameter in gradient boosting regressor algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.

  -Subsample; Subsample is fraction of samples used for fitting the individual tree learners.

- Made an instance of the **GradientBoostingRegressor** and created our grid with the various values for the hyperparameters

```
GBR=GradientBoostingRegressor()
search_grid={'n_estimators':[500,1000,2000],'learning_rate':[.001,0.01,.1],'max_depth':[1,2,4],'subsample':[.5,.75,1],'random_state':[1]}
search=GridSearchCV(estimator=GBR,param_grid=search_grid,scoring='neg_mean_squared_error',n_jobs=1,cv=cross_validation)
search.fit(X_train,y_train)
```

- Determined the **best combination of hyperparameters.**

```
[138] search.best_params_

{'learning_rate': 0.01,
 'max_depth': 1,
 'n_estimators': 2000,
 'random_state': 1,
 'subsample': 0.75}
```

ML Models

Gradient Boosting Regressor

Training the GBR model

- Made an instance of the GradientBoostingRegressor with using the best hyperparameters we found
- Created an instance, GBR, of the class GradientBoostingRegressor, by passing the best hyperparameters, to the constructor.

```
[21] GBR.fit(X_train,y_train.values.ravel())

    GradientBoostingRegressor(alpha=0.9, ccp_alpha=0.0, criterion='friedman_mse',
                              init=None, learning_rate=0.01, loss='ls', max_depth=1,
                              max_features=None, max_leaf_nodes=None,
                              min_impurity_decrease=0.0, min_impurity_split=None,
                              min_samples_leaf=1, min_samples_split=2,
                              min_weight_fraction_leaf=0.0, n_estimators=2000,
                              n_iter_no_change=None, presort='deprecated',
                              random_state=1, subsample=0.75, tol=0.0001,
                              validation_fraction=0.1, verbose=0, warm_start=False)
```

- Fit method was called on the model instance and it predicted y values

ML Models

Gradient Boosting Regressor

Model Evaluation

- Root Mean Square Error(RMSE) on test data is **1.96966**
- Root Mean Square Error (RMSE) of mean of the cross-validation score is **2.0060102251564653**
- Gradient boosting is used to improve a regression tree by creating multiple models.
- Gradient Boost Regressor will have a **better performance** than other methods which we have used in our project.

Advantages of Gradient Boosting

Higher flexibility: Gradient Boosting Regression can be used with many hyper-parameter and loss functions. This made the model highly flexible and it can be used to solve a wide variety of problems.

Better accuracy: Gradient Boosting Regression provided us better accuracy. When we compare the accuracy of GBR with other models we tried