Predicting Absenteeism At Work

**Introduction**

This project is a study on predicting absenteeism at work. Our objective is - given a set of data points that depicts the frequency of absence due to various health reasons, in a given time period, coupled with absentee’s personal information such as commute distance, age, children etc., we want to be able to predict the number of hours the employee would be absent for using machine learning algorithms. Having the insight and the ability to predict future absenteeism is useful to developing companies, in the sense that they can make necessary adjustments to changing circumstances in advance, which gives it a competitive advantage in the market.

As part of project we used Decision Tree, Naïve Bayes, Support Vector Machine and Random Forest classifiers along with Extreme Gradient Descent, Logistic Regression and K-Nearest Neighbors to try and correctly predict the absenteeism category of a given individual. We cleaned the train and test data, converted Absenteeism in Hours into 7 groups as per project document, pre-processed the resulting dataset as per each model’s requirement and evaluated the accuracy of prediction on both train data and test data.

The goal of our project is to evaluate which machine learning algorithms can best classify and predict Absenteeism category in test data. To do so, we first evaluated each model using the train dataset and ranked the models and reapplied them to the test data.

**Dataset**

The dataset that we are using is a collection of absenteeism records from a courier company for a three-year period between July 2007 and July 2010. There are 631 records with 21 variables. A brief explanation of the variables included in the dataset are as follows (this attribute information is obtained from the project description provided):

1. Individual identification (ID) [Note: ID is not a unique identification column, rather it is reflecting a specific employee]

2. Reason for absence (ICD).

Absences attested by the International Code of Diseases (ICD) stratified into 21 categories (I to XXI) as follows:

I Certain infectious and parasitic diseases

II Neoplasms

III Diseases of the blood and blood-forming organs and certain disorders involving the immune mechanism

IV Endocrine, nutritional and metabolic diseases

V Mental and behavioural disorders

VI Diseases of the nervous system

VII Diseases of the eye and adnexa

VIII Diseases of the ear and mastoid process

IX Diseases of the circulatory system

X Diseases of the respiratory system

XI Diseases of the digestive system

XII Diseases of the skin and subcutaneous tissue

XIII Diseases of the musculoskeletal system and connective tissue

XIV Diseases of the genitourinary system

XV Pregnancy, childbirth and the puerperium

XVI Certain conditions originating in the perinatal period

XVII Congenital malformations, deformations and chromosomal abnormalities

XVIII Symptoms, signs and abnormal clinical and laboratory findings, not elsewhere classified

XIX Injury, poisoning and certain other consequences of external causes

XX External causes of morbidity and mortality

XXI Factors influencing health status and contact with health services.

And 7 categories without (CID) patient follow-up (22), medical consultation (23), blood donation (24), laboratory examination (25), unjustified absence (26), physiotherapy (27), dental consultation (28).

3. Month of absence

4. Day of the week (Monday (2), Tuesday (3), Wednesday (4), Thursday (5), Friday (6))

5. Seasons

6. Transportation expense

7. Distance from Residence to Work (kilometers)

8. Service time

9. Age

10. Work load Average/day

11. Hit target

12. Disciplinary failure (yes=1; no=0)

13. Education (high school (1), graduate (2), postgraduate (3), master and doctor (4))

14. Son (number of children)

15. Social drinker (yes=1; no=0)

16. Social smoker (yes=1; no=0)

17. Pet (number of pet)

18. Weight

19. Height

20. Body mass index

21. Absenteeism time in hours (target)

Our target variable is the last feature – “Absenteeism time in hours”, which we will aim to predict through our analysis, based on the rest of the features available to us.

**Overview of Methodology**

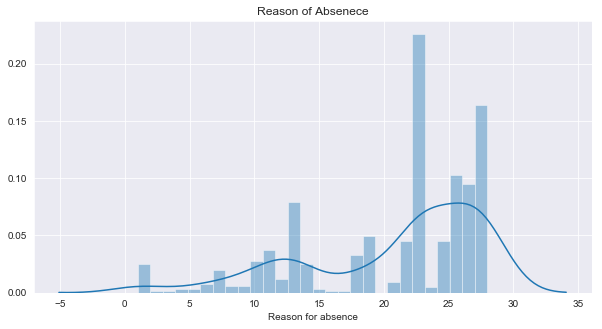
Our analysis follows this 5-step approach:

* Data Cleaning and Preprocessing: Clean and preprocess the dataset to make it analysis ready by removed highly correlated variables for various models
* Data Exploration: Exploring patterns in the data through data visualizations
* Model Building: Build Machine learning models using the training data set
* Model Evaluation: Evaluate the performance of each of the models under consideration on validation data set
* Testing: Test the models evaluated as best ones, on the test data for our final predictions. Each of these steps have been explained in further detail in the following sections of this report.

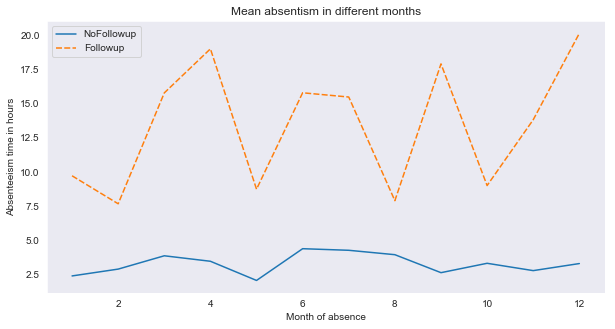
**Exploratory Data Analysis**

Once we got a clean dataset, the next phase was Exploratory Data Analysis (EDA). EDA is the process of figuring out patterns, relationships, or anomalies to help in our subsequent analysis.

Univariate and bi-variate analysis are performed to find relationships between certain set of variables. The absenteeism being the primary target key of this study, we have taken a look at the distribution of reason for this(absenteeism) of employees in the company and found that the highest percentage reason considering the defaulters fall in 22-28 range, which are cases of no follow-up.

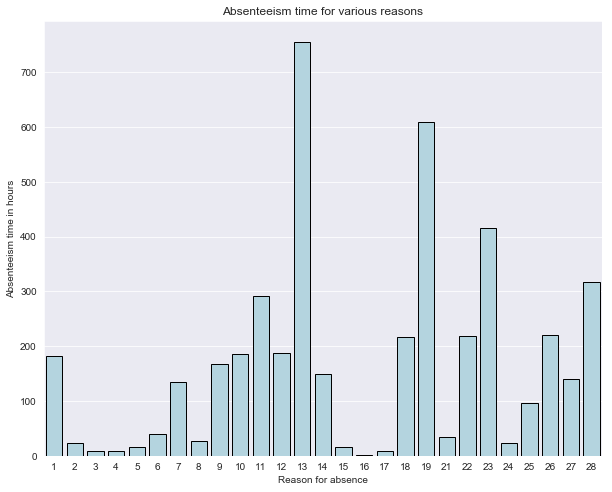
**Figure 1: Distribution of Reason for absence**

Continuing the data exploratory analysis on the hours of absenteeism, across different months, shows high variation for cases where the patients’ needs follow-up compared to no follow-up. The average absenteeism hours for a reason which needs follow-up is 70% greater than, which requires no follow-up.



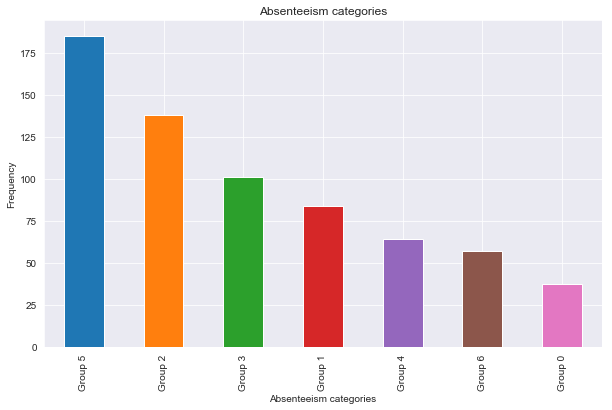
**Figure 2: Distribution of absenteeism w.r.t follow-up needed or not**

Conceptually though there were more records of absence, for no patient follow-up, but the number of absenteeism hours is comparatively less ( i.e 40% approximately) when compared to categories where patients need follow-up. The following bar plot suggests that employees with “Diseases of the musculoskeletal system and connective tissue “ are high percentage(i.e. 17% approximately) of defaulters.



**Figure 3: Absenteeism time w.r.t to reason of absence**

After categorizing the Absenteeism into 7 different groups, based on the hours, is visualized as below. Groups 0-6 indicate less than 1hour, 1-2 hour, 2-3hours, 3-7hours, 7-8 hours, 8-9 hours, more than 9 hours respectively.Group5 i.e the number of 8 hours absence time is the highest group of absenteeism(185 count), is relatively 22%(approximately) more than Group 0 (i.e less than 1hour) which has the least number of absentees, that is equal to 37.All the other groups, fall between these two groups.



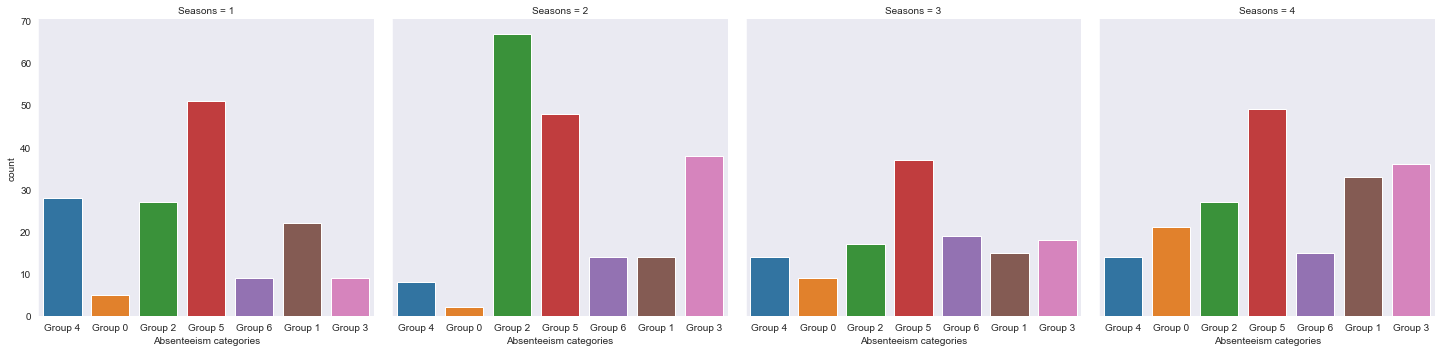
**Figure 4: Frequency of Absenteeism**

The absenteeism categories are faceted on various reasons of absence, is distributed as shown below. Reason of absence 23, which is the absence for medical consultation among the employees is highest in all ranges of timings, followed by reason of absence 28 i.e dental consultation. From this we can infer that employees, took good care of their health by regular health and dental check-up, apparently having less absenteeism due to health issues.



**Figure 5: Absenteeism categories for various reasons**

In response to the question “Whether seasons have any effect on the absenteeism of employees?”, we found that season 1 which happen to be Summer had less absenteeism i.e 21% (approximately) in the company when compared to other seasons. Which apparently a good insight, as people generally fall less sick in summer related to other seasons.

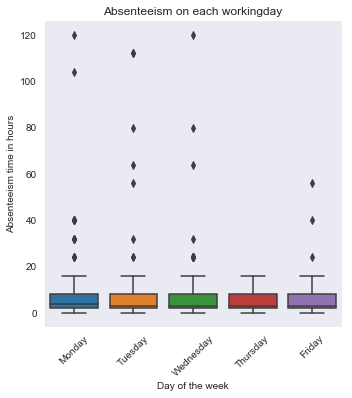


**Figure 6: Absenteeism grouped w.r.t seasons**

Below table shows the absenteeism time in hours, in various seasons for different reasons among employees in the company.

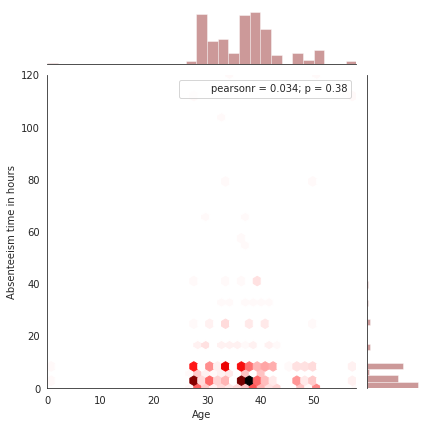
|  |  |
| --- | --- |
| **Seasons** | **Absenteeism time in hours** |
| 1 - Summer | 955 |
| 2 - Winter | 1152 |
| 3 - Spring | 1151 |
| 4 - Fall | 1239 |

When looked into the absenteeism time, on a particular working day in a week, we observed outliers in the first three days of week, and it is obviously making sense after a tiresome weekend activity.



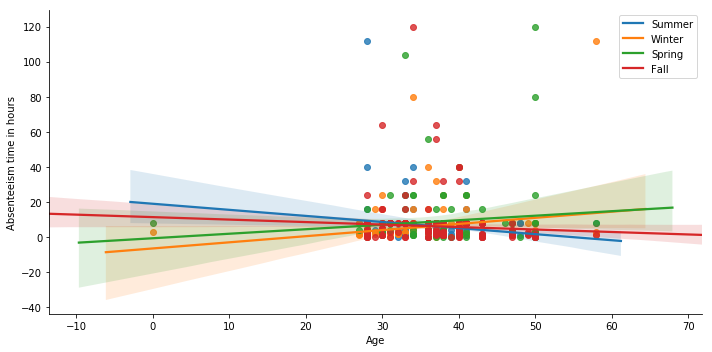
**Figure 7: Absenteeism time in hours w.r.t Weekdays**

Age is another factor, which according to psychology of human being effects the proficiency of work and increase in absenteeism in hours due to health reasons. contradictorily, in the given data set, age has effect or association with absenteeism in hours, with p-value of 0.38, less than level of significance of 0.5.



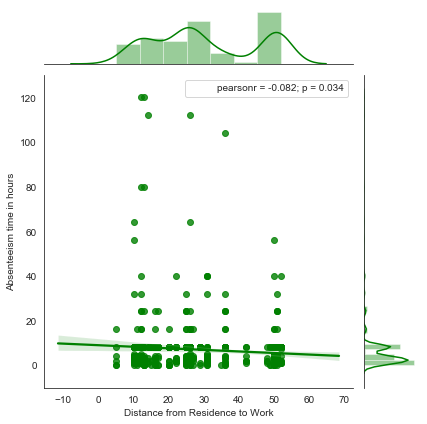
**Figure 8: Effect of Age on Absenteeism time**

Extended the study of analysis on effect of Age on Absenteeism, in various seasons, which figured out to be negatively related in summer. In fall the regression line is almost straight, with no slope specifying age has no effect. And spring and winter have more absenteeism with growing age.

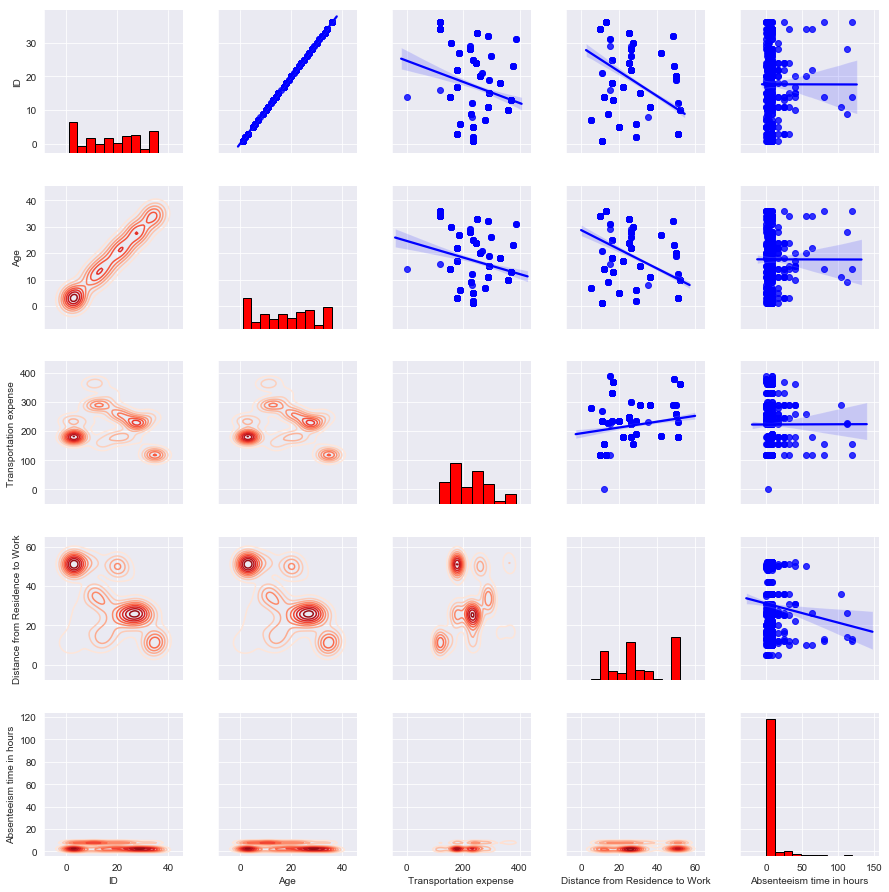


**Figure 9: Effect of Age on Absenteeism time in various seasons**.

With the following ‘reg’ joint plot between Distance from Residence to Work and Absenteeism time in hours, we inferred the presence of some association between variables. we can see individual Pearson residual (-0.082) in order to check association of each variables. By Looking into result, we can say that S the distance from residence to work is more, there seems less absenteeism hours, as shown below.



**Figure 10:Absenteeism time in hours w.r.t Distance from residence to work**



**Figure 11: Pairwise plot between ID, Age, Transport expense, Distance from Residence to work and Absenteeism time in hours.**

A pairwise plot is used to see both distribution of single variables and relationships between two variables and are a great method to identify the relationship between the variables for analysis. The histogram on the diagonal demonstrates the distribution of a single variable, while the scatter plots on the upper and kernel density estimation (KDE) plot on the lower triangles show the relationship between two variables.

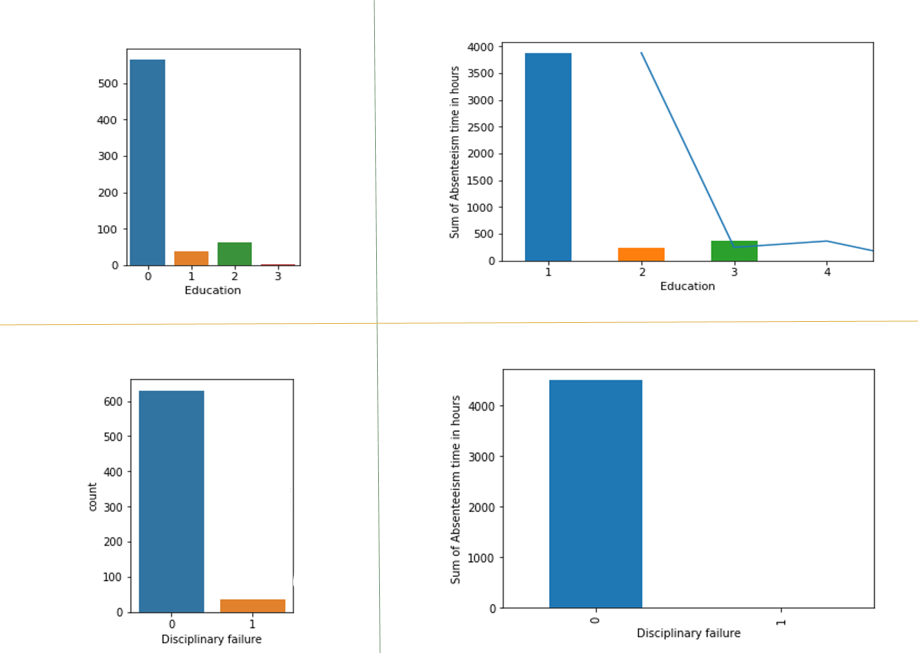
From the plot we can infer that Absenteeism time in hours is independent of ID and Age factor. Whereas the target variable is dependent on slightly on Transport expense and more negatively on Distance from Residence.

A picture containing screen, wall, monitor

Description generated with high confidence

**Figure 12: Probability density plot for Service Time, Work load Average/Day, Hit target and Son**

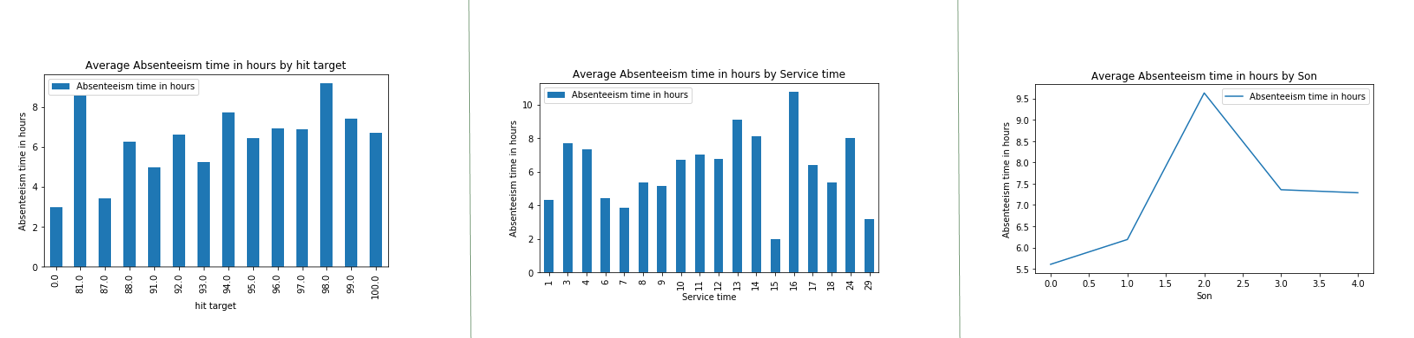
We created this probability density plot to visually get a sense of the distribution of the data. From this PDF plot we can see that Hit Target and Work Load Average/Day are following a closely normal distribution within the range of data that exists for these features. On the other hand, Service Time and Son (i.e. # of children) do not show any obvious distribution pattern.



**Figure 12: Bar plots for Education and Disciplinary failure.**

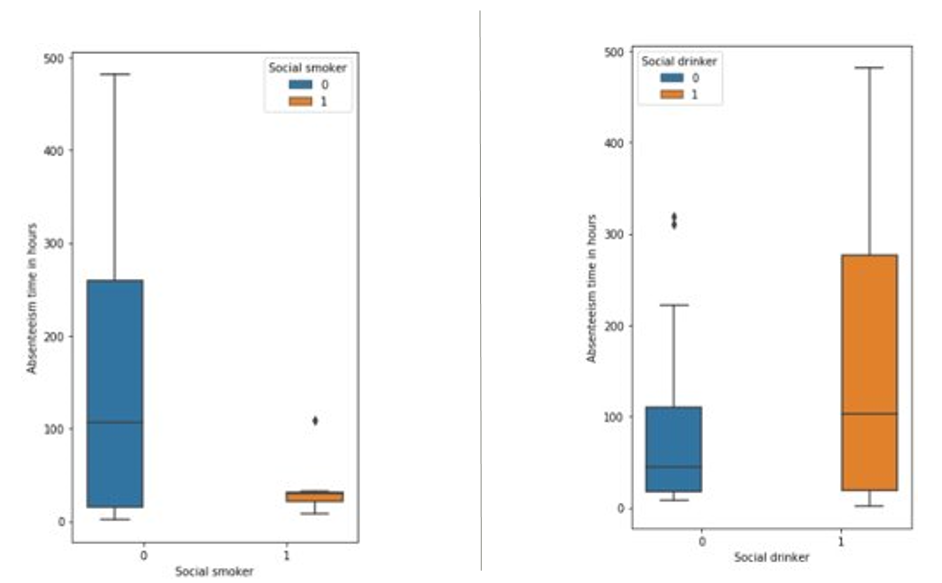
In the left-hand side column of the figure, we're looking at the total count of Education and Disciplinary variables differentiated in color by their categories. The Education count plot shows that most employees in this data set have a high school education only. And the one below shows that very few employees have had disciplinary failure.

On the right-hand side column, we've plotted same variables with respect to total sum of absent hours on y axis. And the first graph of education vs aggregate sum of absent hours is not very surprising. We have more people with high school education, so naturally the sum of absent hours is highest for that group. But the second graph was somehow surprising because, we expected to see employees with disciplinary failure might have more absent hours, but that was not the case. For employees with disciplinary failure, there are number of absent hours is actually zero.



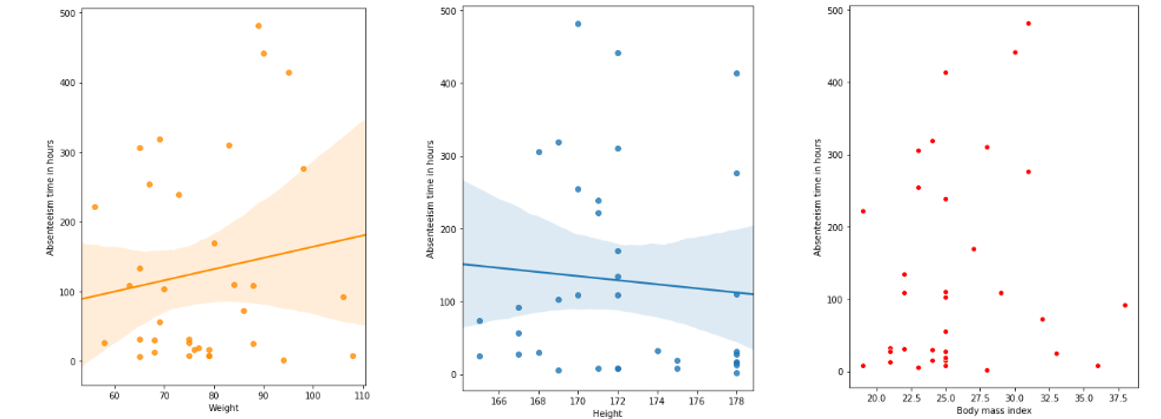
**Figure 13: Bar plots Hit Target, Service Time, # of Children Vs Avg Absenteeism hours**

Through these plots of Hit Target, Service Time, and Son variables plotted against the Average of absenteeism hours of their respective groups, we wanted to observe if there is any relationship between specific values of these variables and the corresponding average value of absenteeism hours for that group. And we see that there is no significant relationship or pattern with these variables either. The only interesting point is, people with “two” children have a higher average of absenteeism hours



**Figure 14: Box plots for Social Drinker/Smoker Vs Sum of total Absenteeism hours**

We took a look at the box plots of total sum absenteeism hours for social smoker on the left-hand side and the social drinkers on the right-hand side plot of this figure. The yellow means that he/she is a smoker or a drinker and the blue mean he is not a smoker or a drinker. This was actually very surprising to see, because we expected to see more absent hours for people who smoke, but that is very less as we can see from the graph on the left side. However, social drinkers seem to be more absent than non-drinkers, which is more or less expected.



**Figure 15: Scatter plots for Height, Weight, BMI vs Sum of total Absenteeism hours**

We created these scatter plots, to see how total number absenteeism hours relate to height, weight and Body mass index. By observing these scatter plots, we can deduce that there is no obvious relationship between these variables and absenteeism hours. Also, body mass index is a function of height and weight anyway, so if height and weight are not related to absenteeism hours, then obviously BMI will also be unrelated.

**Decision Tree Classifier**

**Decision Tree Classifier** preprocessing – The cleaned train and test data are taken, and all columns are converted into categorical data. Below are the results we observed for:

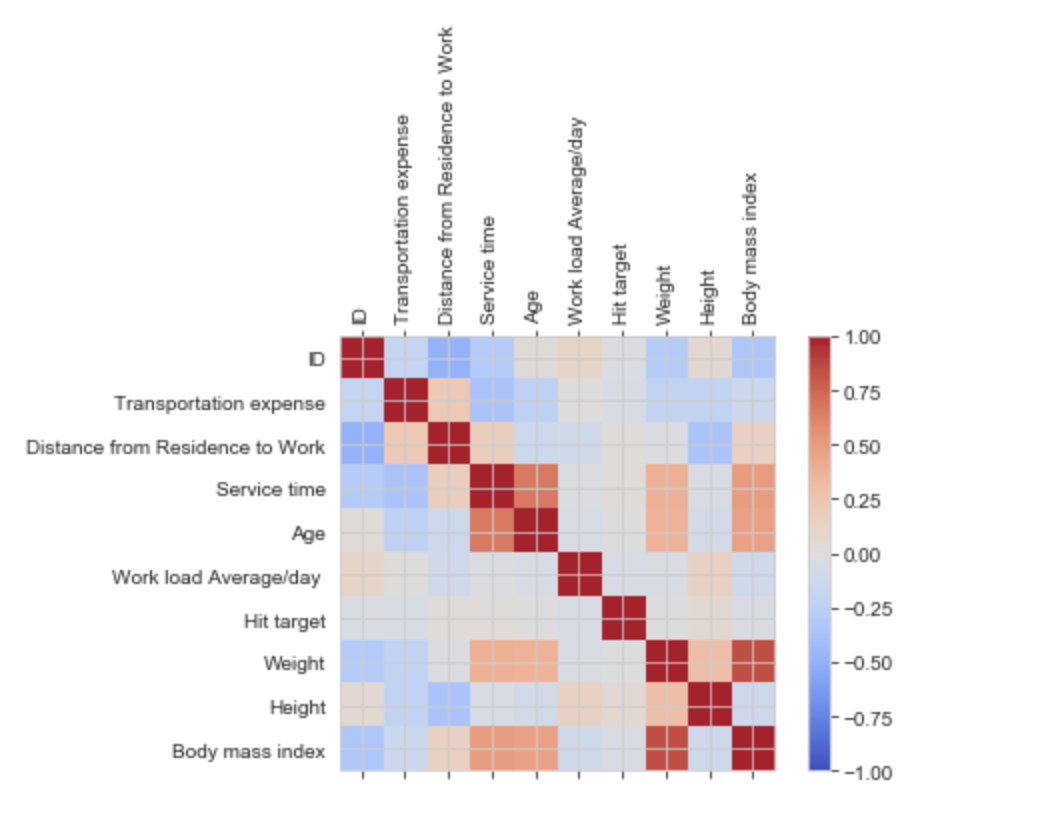
Training Data – (Split: 90:10 for training and testing, Criterion = ‘gini’, max\_depth = 7)

* Accuracy – 47.76%

Validation Test Data

* Accuracy – 43.24%

**Decision Tree Classifier with Feature Selection using correlation** pre-processing – based on the correlation heat plot we eliminated Weight, Age columns from test and train data:



**Figure 16: Correlation plot of feature set**

After removing features that were highly correlated, we scaled the feature column using scale() and below is the accuracy that we got:

Training Data – (Split: 90:10 for training and testing, Criterion = ‘gini’, max\_depth = 7)

* Accuracy – 49.25%

Validation Test Data

* Accuracy – 39.18%

We also applied Decision Tree on continuous data. On the cleaned training data, we applied DecisionTreeRegressor. Below are the results we got:

Training Data – (Split: 90:10 for training and testing, DecisionTreeRegressor)

* Mean Absolute error – 5.14
* Root mean squared error: 14.50

Validation Test Data

* Mean Absolute error – 14.12
* Root mean squared error - 33.36

After applying cross-validation below are the results:

scoring = make\_scorer(mean\_squared\_error)

reg\_cv = GridSearchCV(DecisionTreeRegressor(random\_state=0),

param\_grid={'min\_samples\_split': range(2, 10),'max\_features': ['sqrt', 'log2', **None**]},

scoring=scoring, cv=10, refit=**True**)

Training Data – (Split: 90:10 for training and testing, DecisionTreeRegressor)

* Mean Absolute error – 5.31
* Root mean squared error: 15.366

Validation Test Data

* Mean Absolute error – 12.28
* Root mean squared error – 9.43

The errors for the test data came down drastically after using cross validation on the DecisionTreeRegressor model.

**Random Forest**

For designing the model to predict the absenteeism time category, in spite of single Decision Tree, used ensembled Random Forest Classifier, which creates more than one decision tree algorithm for classifying groups. Results of just one machine learning may not be efficient to rely upon model. Hence, by combining the predictive power of multiple learners into a single model which gives the aggregated output from several models, overcomes the issue of overfitting issue of Decision tree.

The cleaned data set is portioned into 80-20 for training and testing purpose. First using the validation set, random forest is trained as below. 

**Result**:



**Preprocessing – Scaling the features**:

From the accuracy we can infer that model is overfitting, as the prediction rate is high as 95% on trained dataset, but was low on the test dataset, which is a high variance problem. In order to rationalize the bias-variance effect, we later standardized as a of preprocessing before training the model as follows:

**Result**: 

**Dataset size**:

Subsequently our ability to fit the model will depend on the amount of data we have, increased the size of the train to test set to 95 – 5 ratios.

**Result**:



As we see the accuracy improved to 56% (approximately).

**Tuning Parameters**:

The Random forest model was tuned to make predictions of the model better.

* n\_jobs: -1 🡺 to allow engine to run on multiple processors in parallel.
* oob\_score: True 🡺 use out-of-bag samples to estimate the generalization accuracy.
* warm\_start: True 🡺 When set to True, reuse the solution of the previous call to fit and add more estimators to the ensemble.
* min\_samples\_leaf: 2 🡺 The minimum number of samples required to be at a leaf node.

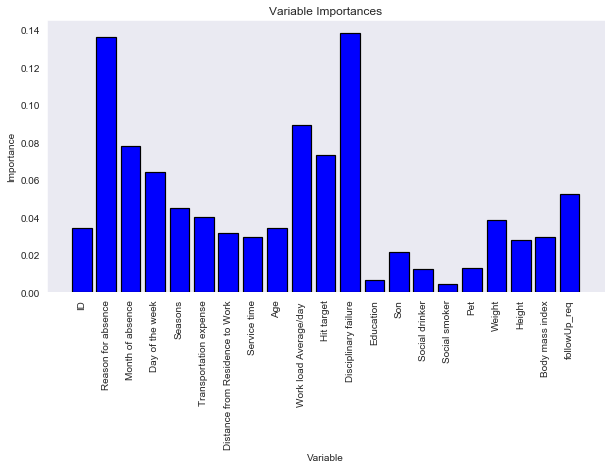


**Result**:



**Feature Selection**:

Until now, we considered all variables or features for modelling, but random forest can identify important features for the model. The importance of all features is plotted as below:



**Figure 17: Feature set vs Importance**

As the plot shows Reason of Absence is the most important feature on the prediction of Absenteeism in the company.



In view of maximum frequency, threshold of 0.033, with 14 features was the optimum condition. Accuracy was 64.71% for the absenteeism dataset.



**Result**:



**PCA before Random Forest**:

PCA is the most popular dimensionality reduction algorithm, which identifies the hyper-plane closest to the data and projects the data onto it. Cumulative sum of the variance is as shown below:

pc.explained\_variance\_ratio\_.cumsum()

array([0.16389609, 0.28065268, 0.39203304, 0.46813478, 0.53531334, 0.59780716, 0.65429825, 0.70665232]) considered 8 components, which explains 70% of variance of all the features and the accuracy of random forest on this principal component was 50%

**LDA before Random Forest**:

LDA which is used to find linear combinations of variables that characterize or separate classes of objects or events. Considering the n\_components of 3, the accuracy was 58%. Hence for random forest the maximum accuracy was with 14 features, i.e. 65%(approximately) on validation set.

**Extreme Gradient Descent**

Extreme gradient boosting as name suggests is an ensembled boosting method. Gradient descent is used for optimizing the loss function. And in boosting, the trees are built sequentially such that each subsequent tree aims to reduce the errors of the previous tree, updating the residual errors. Using XGBOOST library in python, trained the model and studied the accuracy.

Model:



Result:



With default parameters the accuracy is 62% (Approximately). Working with XGBOOST to improve the efficiency, did feature importance. This importance gives a score that specifies how each variable is efficient in the building of the boosted decision trees within the model. The optimized number of features was noticed as 10.

**Result**:

Best was Thresh=0.047, n=10, Accuracy: 67.65%.

The accuracy improved to 67.65% with ten features. Generally deeper trees would result in fewer trees being required in the model, and vice versa to achieve comparable results. Investigated this relationship by evaluating a grid of **n\_estimators** and **max\_depth** configuration values. Created a grid with 4 different n\_estimators values (50, 100, 150, 200) and 4 different max\_depth values (2, 4, 6, 8) and 5 different learning\_rate values or step size [0.0001, 0.001, 0.01, 0.1, 0.2, 0.3]. Fitting 10 folds for each of 96 candidates, totaling 960 fits.

**Result**:

Best: -1.281691 using {'learning\_rate': 0.1, 'max\_depth': 4, 'n\_estimators': 50}

We can see that the best result was achieved with a **n\_estimators=50** and **max\_depth=4**,at learning rate of 0.1. Using these parameters, the accuracy is recalculated.

Result:



**Logistic Regression**

Multinomial logistic regression is used for classification problem that generalizes logistic regression to multiclass variables, in which the log odds of the outcomes are modeled as a linear combination of the independent variables. Using OneVsOneClassifier in sklearn package the model is trained to classify the response variable.

**Model**:



**Result**:



**Naïve Bayes**

**Naïve Bayes** preprocessing - For applying Naïve Bayes model to our dataset, we performed some more additional preprocessing on the cleaned data. From the cleaned data we removed Month, Day, Seasons since time of absenteeism didn’t have much impact on it. Weight and Height were removed since we included BMI column.

Next, we converted all columns with continuous/discreet data to ordered categorical data by binning technique using IQR – Interquartile Range. So, values in each column less than 25th percentile were set to 0, between 25th and 50th percentile – 1, between 50th and 75th percentile – 2, and above 75th percentile – 3. After this step - Workload, Age, Target, Service time, BMI, Distance, Transport Cost columns had values in [0, 1, 2, 3]. ID and Reason were converted to nominal data (i.e. categories without order). Social Drinker (0, 1), Social Smoker (0,1), Disciplinary Action (0, 1), Pets (0 – 4), Education (1 – 4) were already ordinal data to begin with.

Accuracy of the model – 15 features, 1 target:

Training Data: (Split – 90:10 for training and testing, Gaussian Naive Bayes)

* Accuracy – 38.81%
* Cross validated Accuracy (8 splits) – 34%

Validation Test Data:

* Accuracy – 39.19%

**Naïve Bayes with Feature Selection using Correlation** preprocessing - The assumption in Naïve Bayes is that all columns are independent. In our project we achieve this by removing all columns that are correlated. For this we use the corr() on the feature set data-frame (X) i.e. all columns except the target column and remove features correlated with other features by a value > 0.3. We start by taking the pre-processed data for Naïve Bayes above, and then we remove features that are not entirely independent with one another. After this step we end up with 9 features – ID, Reason, Transport Expense, Distance, Workload, Target, Disciplinary, Education, Social Smoker. We pre-process the test data in a similar fashion.



**Figure 18: Correlation plot between feature set**

Accuracy of the model – 9 features, 1 target:

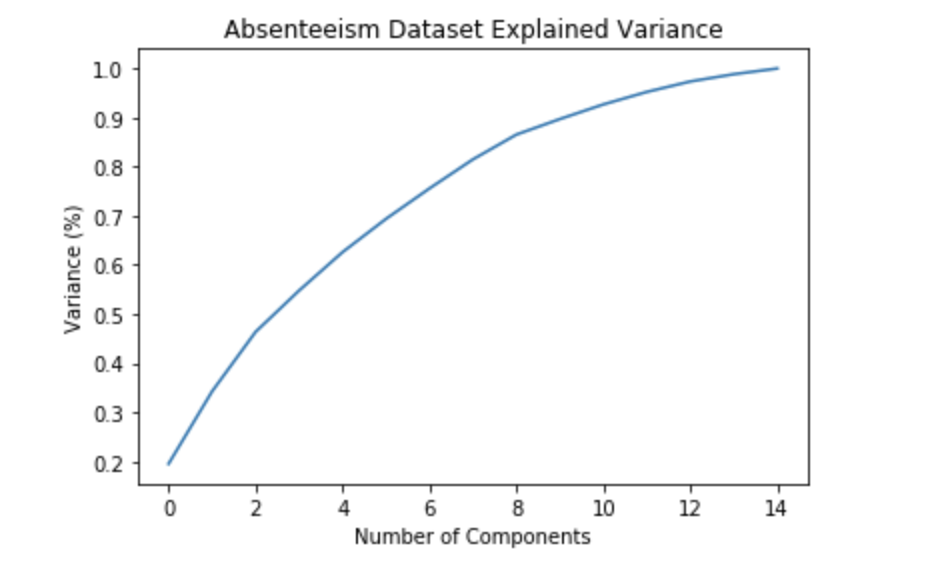
Training Data: (Split – 90:10 for training and testing, Gaussian Naive Bayes)

* Accuracy – 46.27%
* Cross validated Accuracy (8 splits) – 32%

Validation Test Data:

* Accuracy – 36.49%

**Naïve Bayes with PCA** preprocessing - We use PCA to achieve dimensionality reduction by transforming data into a set of uncorrelated/independent variables. In our project, we begin by taking the cleaned data, then we scale the feature set i.e. scale all columns except the target column (Absenteeism Categories) using StandardScaler. The scaling is done for the purposes of running SVM-RBF kernel, we just use the same data-frame for PCA transformation for Naïve Bayes as well. On this scaled feature set we perform data transformation using PCA with n\_components = 6 that captures about 70% variance in training data. Now the number of features in the transformed feature set is 6. We use the same “pca” object to perform a similar transformation on test data as well.



**Figure 19: PCA Plot – Variability vs # of Components**

Accuracy of the model – 6 features, 1 target:

Training Data: (Split – 90:10 for training and testing, Gaussian Naive Bayes)

* Accuracy – 49.25%
* Cross validated Accuracy (8 splits) – 41%

Validation Test Data:

* Accuracy – 39.19%

**Support Vector Machine Classifiers**

**SVM RBF Kernel** preprocessing - We begin by taking the cleaned data then we scale the feature set i.e. scale all columns except the target column (Absenteeism Categories) using StandardScaler since all SVM kernel methods are based on distance it is important to scale data to avoid bias to columns with greater values. We do the same for test data as well.

Accuracy of the model – 15 features, 1 target:

Training Data: (Split – 90:10 for training and testing, Gamma = ‘auto’, RBF Kernel)

* Accuracy – 58.21%
* Cross validated Accuracy (8 splits) – 44%

Validation Test Data:

* Accuracy – 43.24%

**SVM RBF Kernel with Feature selection using Correlatio**n preprocessing - We use the same train and test pre-processed feature set data-frames that we used for Naïve Bayes Classification with Feature Selection using Correlation model. On those data-frames, we run SVM classification using RBF kernel and below are the results:

Accuracy of the model – 9 features, 1 target:

Training Data: (Split – 90:10 for training and testing, Gamma = ‘auto’, RBF Kernel)

* Accuracy – 56.72%
* Cross validated Accuracy (8 splits) – 42%

Validation Test Data:

* Accuracy – 48.65%

**SVM Classifiers RBF kernel with PCA** preprocessing - We use the same train and test transformed feature set data-frames we used for Naïve Bayes with PCA model. On those data-frames, we run SVM classification using RBF kernel and below are the results:

Accuracy of the model – 6 features, 1 target:

Training Data: (Split – 90:10 for training and testing, Gamma = ‘auto’, RBF Kernel)

* Accuracy – 56.72%
* Cross validated Accuracy (8 splits) – 42%

Validation Test Data:

* Accuracy – 50%

**Other SVM Classifier accuracies on Training Data**:

* **Kernel: Linear**
  + On cleaned training data
    - Accuracy – 41.28%
  + On cleaned training data with PCA - 79% variance captured (n\_components = 10)
    - Accuracy –46%
  + On cleaned training data with scaling (using scale())
    - Accuracy – 42.78%
* **Kernel: Poly**
  + On cleaned training data with PCA - 79% variance captured (n\_components = 10)
    - Accuracy – 42.93%
  + On cleaned training data
    - Accuracy – 41.28%

**K-Nearest Neighbors**

As part of our project we tried to use the K-Nearest Neighbors unsupervised classification algorithm. Below are the results we got:

Training Data: Spilt – 90:10 for training and testing, n\_neighbors=23

* Accuracy – 44.77%

Validation Test Data:

* Accuracy – 40.5%

Classification Report for KNN on test data:

Classification report:

precision recall f1-score support

0 1.00 0.57 0.73 7

1 0.00 0.00 0.00 4

2 0.52 0.58 0.55 19

3 0.00 0.00 0.00 11

4 0.00 0.00 0.00 4

5 0.39 0.61 0.47 23

6 0.33 0.17 0.22 6

micro avg 0.41 0.41 0.41 74

macro avg 0.32 0.28 0.28 74

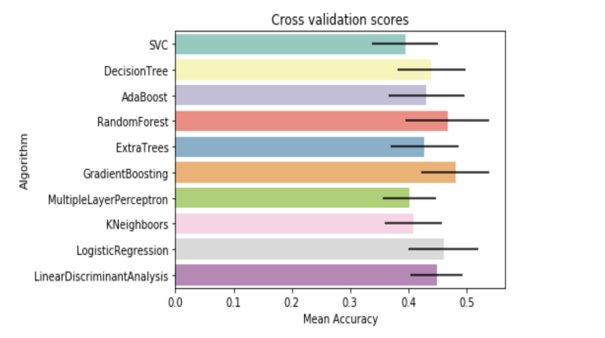
weighted avg 0.38 0.41 0.38 74

Looking at the classification report, the KNN algorithm is best able to identify Group 5 while the precision and recall values for Groups 1, 3, and 4 are 0 which means the algorithm is not able to identify them.

**Model Ensembling**

In statistics and machine learning, ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone.

We analyzed performance of 10 difference algorithm as listed below on our training dataset to understand best individual accuracy (parameters tuned using Grid search)



**Figure 20: Cross Validation Accuracy Bar Plot of Different Models on Training Data**

We got the best accuracy numbers around 45 to 50%, we then picked 3 algorithm that were least correlated - XGB, Random Forest and Linear Regression to create a simple voting ensembler. By doing so we were able to increase the accuracy of the training dataset to 60%.

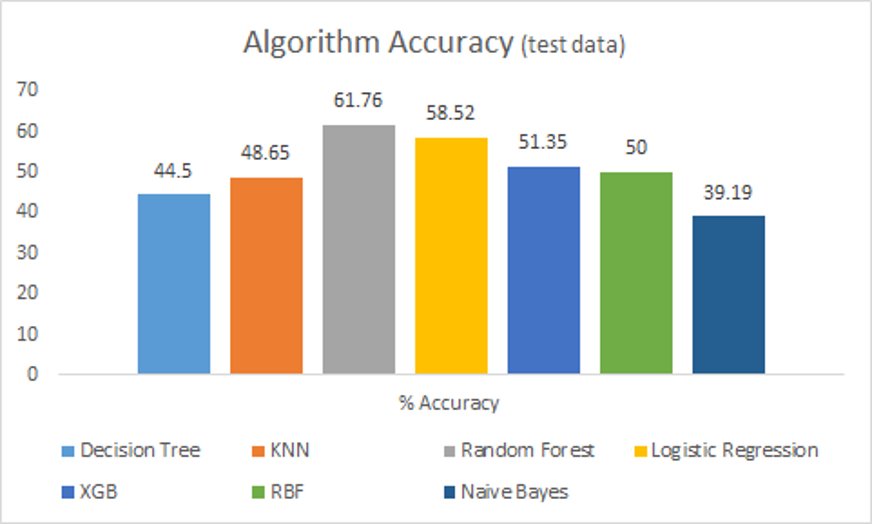
**Conclusion**

We used 7 different methods on our testing data set - Decision Tree, KNN, Random Forest, LR, XGB, RBF and multiple versions of Naive Bayes. Our three best models for training data based on accuracy obtained are:



**Figure 21: Accuracy of algorithms on Train data**

We got the following accuracies on our test data:

****

**Figure 22: Accuracy of various algorithms on Test data**

We were able to achieve test accuracy of 61% using random forest which was close to what we were able to achieve on training dataset thus avoiding overfitting.

**References**

<https://en.wikipedia.org/wiki/Ensemble_learning>

<https://en.wikipedia.org/wiki/Multinomial_logistic_regression>

<https://scikit-learn.org/stable/modules/svm.html>

<https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>