Capstone Project

Machine Learning Engineer Nanodegree

Title: Evaluation of Nursery School Applications

Alphonso Jo Stephen
February 9th, 2019

I. Definition

Domain Background:

Nursery school, pre-primary school, playschool or kindergarten, is an educational establishment or learning space offering early childhood education to children before they begin compulsory education at primary school. It may be publicly or privately operated, and may be subsidized from public funds.

In an age when school was restricted to children who had already learned to read and write at home, there were many attempts to make school accessible to orphans or to the children of women who worked in factories.

With development of science and technology and with improved civilization, the need for basic education has increased which in turn lead to high number of schools and colleges. But here we are only discussing about Nursery school which is the first door towards education

Here I am finding out whether a child is accepted into a nursery school or not by evaluating the accuracy and acceptance

Database reference: https://archive.ics.uci.edu/ml/machine-learning-databases/nursery/nursery.names

Problem Statement:

The main objective of this project is to evaluate all the applications that are enrolled into a nursery school and identify the children who are capable of getting an admission based on some factors. Classification technique like Decision Tree classifier will be implemented and compared upon standard metric like accuracy. Pipeline is also used to avoid any data leakage.

There are several factors which are based upon to accept the application form like parents, social status, health checkup, housing status, financial stability. Nursery School prediction predicts the acceptance of children into the kindergarten.

Metrics:

Accuracy:

It is the number of correct predictions made by the model over all kinds of predictions made

$$Accuracy = (TP+TN) / (TP+FP+FN+TN)$$

Accuracy is a good measure when the target variable classes in the data are nearly balanced. Here I am predicting the accuracy score for selected models and the best score model is selected.

II. Analysis

Data Exploration:

Dataset Link: https://archive.ics.uci.edu/ml/datasets/Nursery'

In this project I have used 9 attributes and around 1200 trained and test data to evaluate the accuracy score. And this accuracy of classifiers is found out by 5-fold cross validation. The attributes used are shown in the below table

	parents	has_nurs	form	children	housing	finance	social	health	target
0	usual	proper	complete	1	convenient	convenient	nonprob	recommended	recommend
1	usual	proper	complete	1	convenient	convenient	nonprob	priority	priority
2	usual	proper	complete	1	convenient	convenient	nonprob	not_recom	not_recom
3	usual	proper	complete	1	convenient	convenient	slightly_prob	recommended	recommend
4	usual	proper	complete	1	convenient	convenient	slightly_prob	priority	priority
5	usual	proper	complete	1	convenient	convenient	slightly_prob	not_recom	not_recom
6	usual	proper	complete	1	convenient	convenient	problematic	recommended	priority
7	usual	proper	complete	1	convenient	convenient	problematic	priority	priority
8	usual	proper	complete	1	convenient	convenient	problematic	not_recom	not_recom
9	usual	proper	complete	1	convenient	inconv	nonprob	recommended	very_recom
10	usual	proper	complete	1	convenient	inconv	nonprob	priority	priority
11	usual	proper	complete	1	convenient	inconv	nonprob	not_recom	not_recom
12	usual	proper	complete	1	convenient	inconv	slightly_prob	recommended	very_recom
13	usual	proper	complete	1	convenient	inconv	slightly_prob	priority	priority
14	usual	proper	complete	1	convenient	inconv	slightly_prob	not_recom	not_recom
15	usual	proper	complete	1	convenient	inconv	problematic	recommended	priority
16	usual	proper	complete	1	convenient	inconv	problematic	priority	priority

Data Set Information:

Nursery Database was derived from a hierarchical decision model originally developed to rank applications for nursery schools. It was used during several years in 1980's when there was excessive enrollment to these schools in Ljubljana, Slovenia, and the rejected applications frequently needed an objective explanation. The final decision depended on three sub problems: occupation of parents and child's nursery, family structure and financial standing, and social and health picture of the family. The model was developed within expert system shell for decision making DEX (M. Bohanec, V. Rajkovic: Expert system for decision making. Sistemica 1(1), pp. 145-157, 1990.).

Features Information:

- parents - usual or pretentious (real or adapted parents)
- has_nurs – has proper nursing or not
- form – application form complete or incomplete
- children – number of children
- housing – status of house (convenient or inconvenient)
- finance – financial stability(convenient or not)
- social - problematic or not
- health – health status
- target – final decision (recommended or not)

Description of data

```
In [5]: display(df.describe())
```

	parents	has_nurs	form	children	housing	finance	social	health	target
count	12960	12960	12960	12960	12960	12960	12960	12960	12960
unique	3	5	4	4	3	2	3	3	5
top	great_pret	very_crit	foster	more	critical	inconv	problematic	not_recom	not_recom
freq	4320	2592	3240	3240	4320	6480	4320	4320	4320

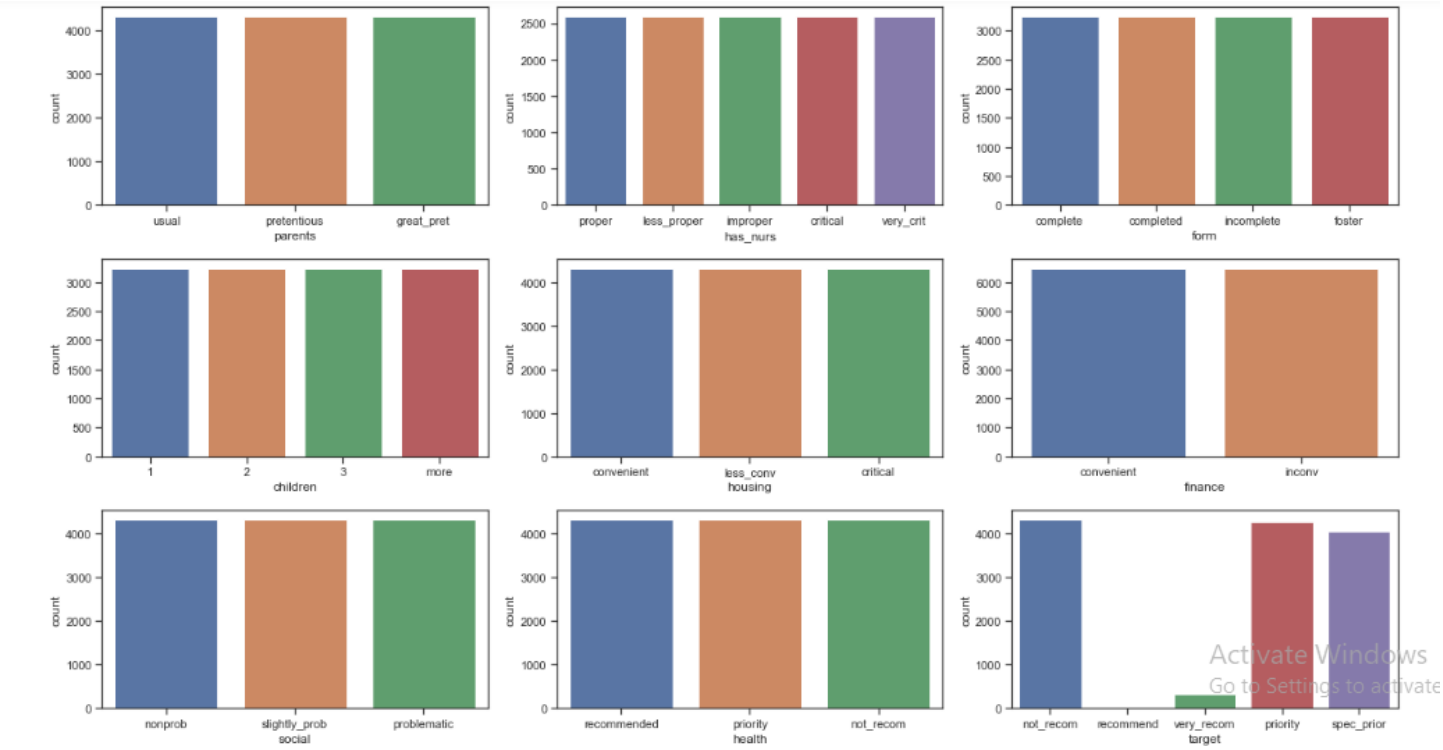
Information of Data set:

```
inputs = pd.read_pickle("inputs.pkl")
inputs.info()
```

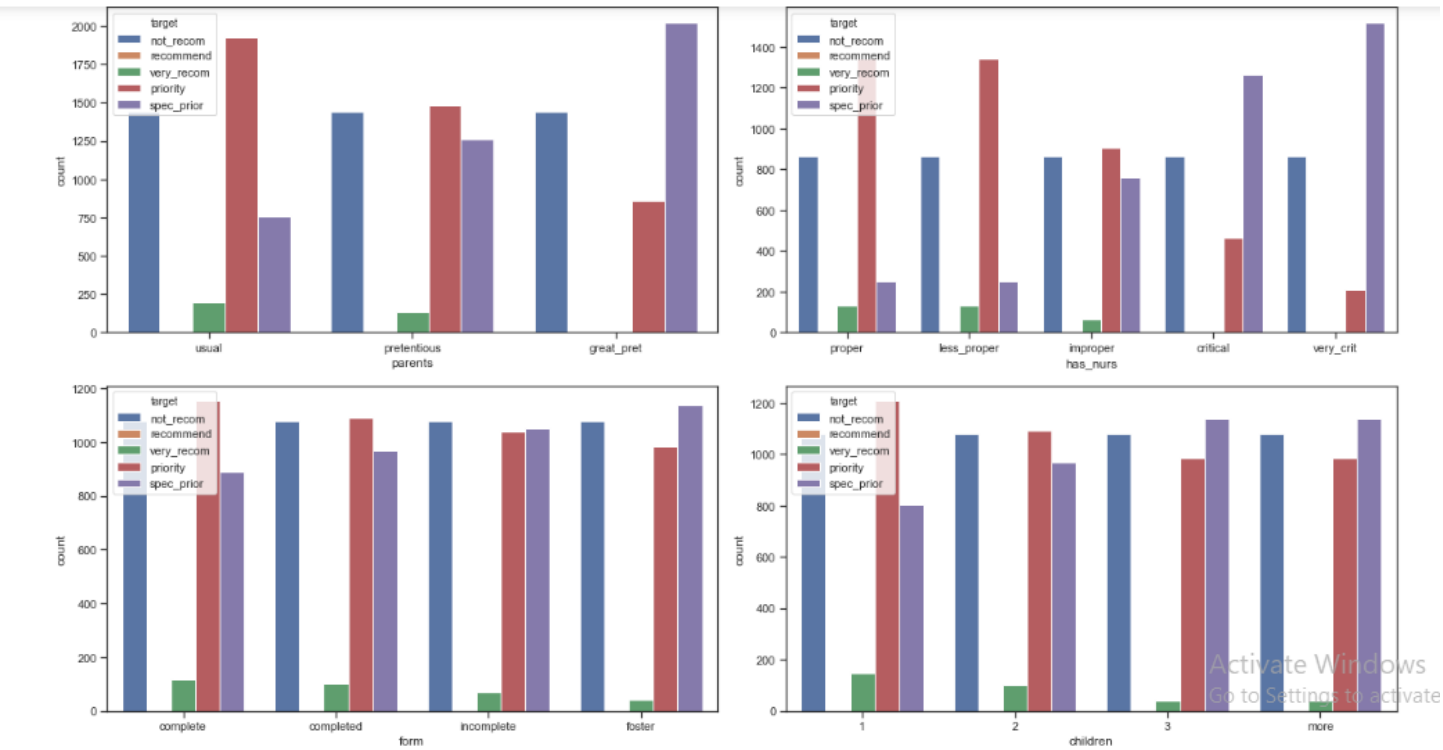
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12960 entries, 0 to 12959
Data columns (total 8 columns):
parents     12960 non-null category
has_nurs    12960 non-null category
form        12960 non-null category
children    12960 non-null category
housing     12960 non-null category
finance     12960 non-null category
social      12960 non-null category
health      12960 non-null category
dtypes: category(8)
memory usage: 101.5 KB
```

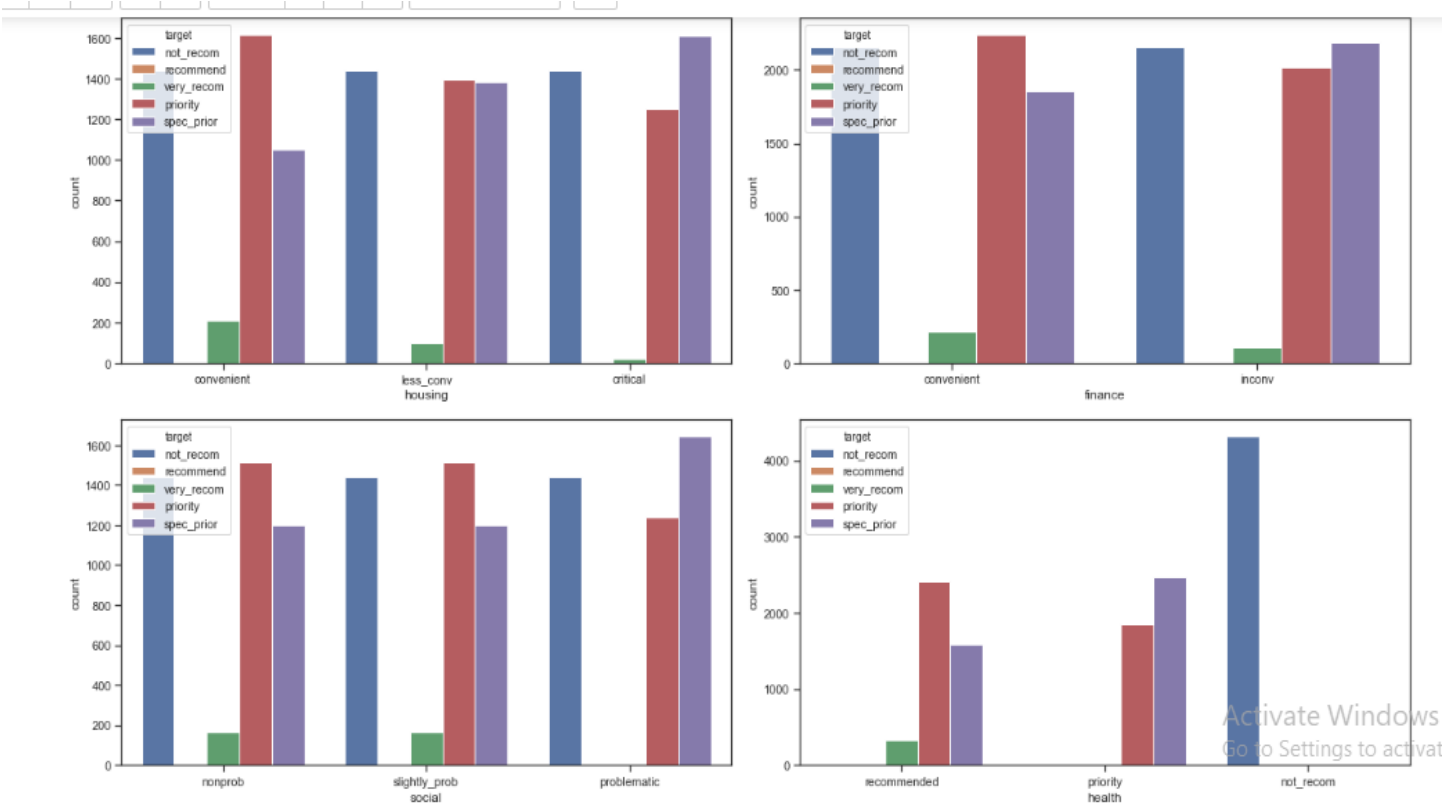
There are no missing values existing in the data set, so we don’t need to modify any data

Data Visualization:



Each of our features is perfectly balanced with respect to the values they can take. That is not what real life datasets would look like, but oh well. We see however that the target class is interesting: almost no observations are labelled as recommended, very few are much recommended; most of them are not recommended, recommended with priority or recommended with special priority.

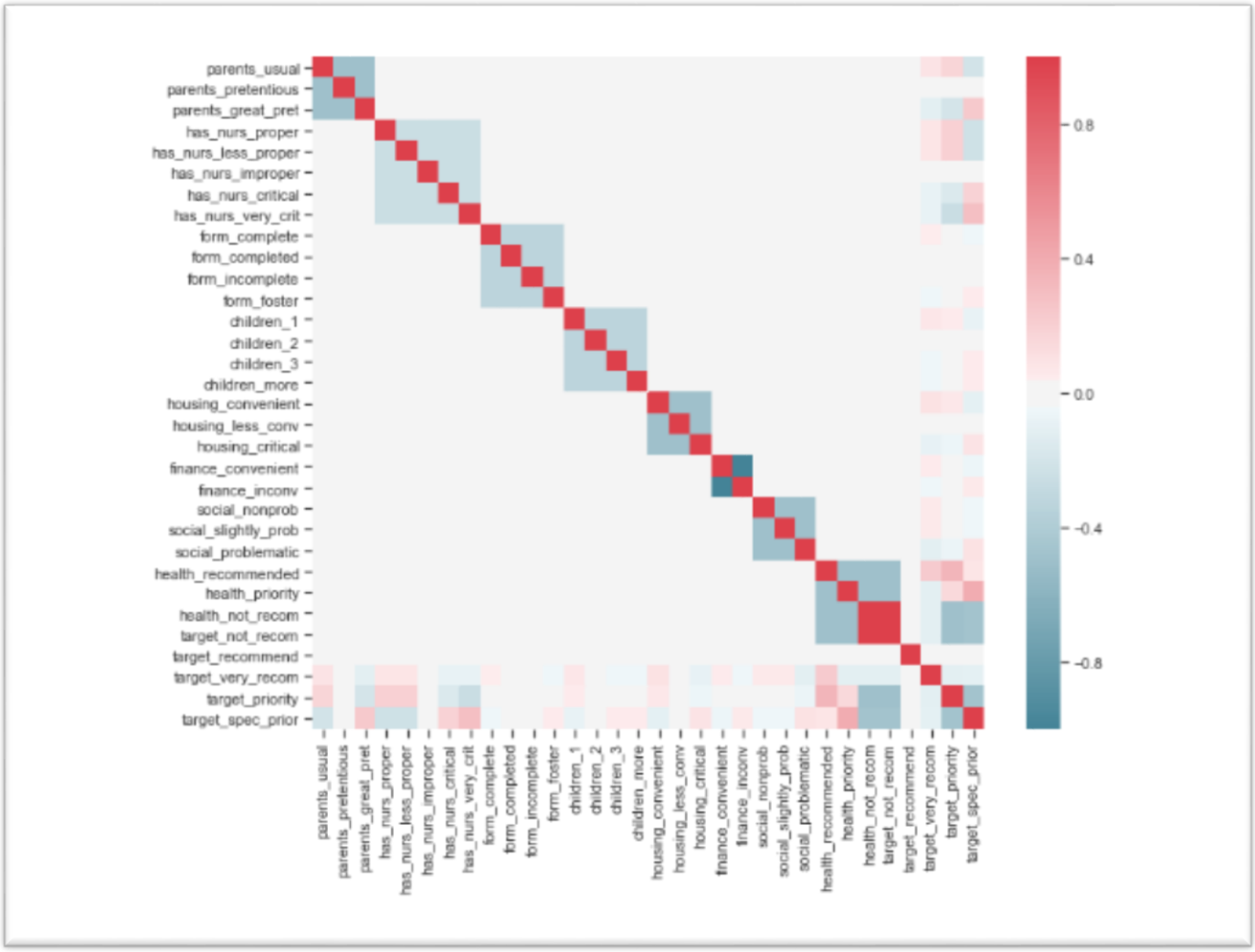




There seems to be some correlation between the family's health status and the acceptance outcome. In particular, in the last plot, we clearly see that if the health status of the family does not recommend final acceptance, the final decision will be to not recommend the child. To visualize this, and some other useful relations in the data, we will resort to use the correlation matrix between each values the features (and the target) can take.

Heat Map:

The heat map is a 2-D representation of data in which values are represented by colors. A simple heat map provides the immediate visual summary of information. More elaborate heat maps allow the user to understand complex data.



Indeed, we found that the correlation between target_not_recom and health_not_recom is 1.0 - They're the same column! At the same time, finance_convenient is perfectly negatively correlated with finance_inconv - which means the two columns are complements.

Modelling and Predicting with Machine Learning:

The main goal of the entire project is to evaluate the nursery school application forms with the highest accuracy. In order to achieve this, we will test several classification algorithms. This section includes all results obtained from the study and introduces the best performer according to accuracy metric. I have chosen several algorithms typical for solving supervised learning problems throughout classification methods.
First of all, let's equip ourselves with a handy tool that benefits from the cohesion of SciKit Learn library and formulate a general function for training our models. The reason for displaying accuracy on both, train and test sets, is to allow us to evaluate whether the model overfits or underfits the data (so-called bias/variance tradeoff).

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.preprocessing import StandardScaler
np.random.seed(1234)

inputs = pd.read_pickle("inputs.pkl")
targets = pd.read_pickle("targets.pkl")
colnames = inputs.columns.tolist()
```

Algorithms and Techniques:

1. K-Nearest Neighbours
2. Decision Trees
3. Logistic Regression
4. Gaussian Naïve Bayes
5. Support Vector Machines
6. Random Forests

1. K-Nearest Neighbours (KNN)

K-Nearest Neighbours algorithm is a non-parametric method used for classification and regression. The principle behind nearest neighbour methods is to find a predefined number of training samples closest in distance to the new point and predict the label from these.

Advantages:

- The K-Nearest Neighbor (KNN) Classifier is a very simple classifier that works well on basic recognition problems.

Disadvantages:

- The main disadvantage of the KNN algorithm is that it is a *lazy learner*, i.e. it does not learn anything from the training data and simply uses the training data itself for classification.
- To predict the label of a new instance the KNN algorithm will find the *K* closest neighbors to the new instance from the training data, the predicted class label will then be set as the most common label among the *K* closest neighboring points.
- The algorithm must compute the distance and sort all the training data at each prediction, which can be slow if there are a large number of training examples.
- Another disadvantage of this approach is that the algorithm does not learn anything from the training data, which can result in the algorithm not generalizing well and also not being robust to noisy data.

2. Decision Tree

Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data. Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.

Advantages: Decision Tree is simple to understand and visualize, requires little data preparation, and can handle both numerical and categorical data.

Disadvantages: Decision tree can create complex trees that do not generalize well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.

3. **Logistic Regression**

Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

Advantages: Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.

Disadvantages: Works only when the predicted variable is binary, assumes all predictors are independent of each other, and assumes data is free of missing values.

4. **Naïve Bayes**

Naive Bayes algorithm based on Bayes' theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.

Advantages: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

Disadvantages: Naive Bayes is known to be a bad estimator.

5**. Support Vector Machine**

Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Advantages: Effective in high dimensional spaces and uses a subset of training points in the decision function so it is also memory efficient.

Disadvantages: The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

6. **Random Forest**

Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Advantages: Reduction in over-fitting and random forest classifier is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.

Benchmark Model:

Here we compare the final model with the remaining models to see if it got better or same or worse. The accuracy scores compared among the models and the optimal one is selected. I choose Logistic Regression model as the benchmark model. Now we will try and achieve better accuracy than this model by using the above mentioned classification models.

III. Methodology:

Pre-Processing:

In this step we will pre-process the data. Data pre-processing is considered to be the first and foremost step that is to be done before starting any process. We will read the data by using read_csv. Then we will know the shape of the data. And by using the info () we will know the information of the attributes. Then we will check whether there are any null values by using isnull ().We will import LabelEncoder from sklearn.preprocessing we will also use fit_transform(y) for Fit label encoder and return encoded labels. After doing that we will use le.transform () for Transform labels to normalized encoding. After that we will divide the whole data into training and testing data. We will assign 70% of the data to the training data and the remaining 30% of the data into testing data. We will do this by using train_test_split from sklearn.model_selection.

Implementation:

Out of the chosen algorithms we will start with KNN classification model. We will take a classifier and fit the training data. After that we will predict that by using predict (X_train). Now we will predict the accuracy of the testing data by using accuracy score (y_test, pred) and F-score by importing fbeta_score from sklearn.metrics. By doing so for, the KNN will give us the accuracy of 91.74%.We will continue the same procedure on Naïve Bayes, SVM, Decision tree, Logistic Regression and Random Forest. By following the same procedure above that is fitting, predicting and finding the accuracy score, we will get the accuracy scores as below.

	Accuracy
KNN	91.74%
Decision Tree	99.54%
Logistic Regression	76.70%
Naïve Bayes	64.24%
Support Vector Machine	96.10%
Random Forests	97.34%

From the above table, we can clearly see that the Decision Tree Classifier performs the best compared to other models

Refinement

I found out 'Decision Tree' as the best classifier out of the chosen classifiers. For refinement, we will perform 5-fold cross validation to avoid over-optimistic. This is as shown below

```
dt = DecisionTreeClassifier()
```

```
# 5-fold cross-validation
print(cross_val_score(dt, X_train, y_train, scoring="accuracy", cv=5).mean())
```

```
0.9917036791360839
```

Now we get a new accuracy of 99.17%

IV. Result

Model evaluation and validation

The final model we have chosen is Decision Tree which gave us more accuracy that is 0.9917. Here we can say that the solution is reasonable because we are getting much less accuracy while using other models. The final model that is tuned random forest has been tested with various inputs to evaluate whether the model generalizes well. This model is also robust enough for the given problem. We can say this by testing it over different random sates. From this we can say that small changes in the training data will not affect the results greatly. So the results found from this model can be trusted.

Justification:

My final model's solution is better than the benchmark model.

	Random Forest	Benchmark Model
Accuracy	0.9917	0.7670

From the above we can conclude that the results for the final model are stronger than the benchmark model. Hence we can say that the decision tree provides the significant to solve the problem of evaluating nursery school application.

V. Conclusion:

The goal of the project was to compare different machine learning algorithms and predict if a child is eligible into nursery school by using different features like 'parents', 'has_nurs', 'form', 'children', 'housing', 'finance', 'social', 'health', . Here are the final results.

	Accuracy
KNN	91.74%
Decision Tree	99.17%(after 5-fold cross validation)
Logistic Regression	76.70%
Naïve Bayes	64.24%
Support Vector Machine	96.10%
Random Forests	97.34%

Form the results we can easily see that the Decision tree classifier model has the highest accuracy score of 0.9917 compared with the other models. By producing decent results, simpler methods proved to be useful as well.

Reflection:

- I have learnt how to visualize and understand the data.
- I have learnt that the data cleaning place a very vital role in data analytics.
- Removing the data features which are not necessary in evaluating model is very important.
- I got to know how to use the best technique for the data using appropriate ways
- I got to know how to tune the parameters in order to achieve the best score.
- On a whole I learnt how to graph a dataset and applying cleaning techniques on it and to fit the best techniques to get best score.

Improvement:

The process which I have followed can be improved to classify not only for Nursery school application but can be extended to all other colleges or University application rankings. As we can say that there has never been an end in the machine learning there will be many more models to learn. By taking more amount of datasets, we can get the model more generalized and optimized to get better accurate results.