UvA Machine Learning Final Group Project - Group 12

Kelvin Gilian Mulder - 12253286, Nitai Nijholt - 12709018, Kevin Veenboer - 12604275

1. Introduction:

It is common knowledge that smoking shortens lifespan and reduces quality of life (Goldenberg, Danovitch IsHak. 2014; Bernhard, Moser, Backovic, Wick, 2007). This is why many governments have anti-smoking programs in place to reduce the strain of smoking related healthcare burden and improve quality of life for their citizens. (Xu, Bishop, Kennedy, Simpson, Pechacek, 2015; De Vries Mudde, Leijs, Charlton, Vartiainen,Buijs, ... & Kremers, S. 2003). A lot of research has been done on damage smoking does and how to rehabilitate from smoking, but as with many other injuries and diseases, prevention remains the best cure. In order to effectively prevent smoking, governments might look at which demographic factors are correlated with an increased chance of smoking incidence and use this information to target their anti smoking campaigns more effectively. This could then help prevent people from taking up smoking which would increase both healthcare system efficiency and improve quality of life in a cost effective way.

However, little research has been done about the underlying demographic factors which influence the uptake of smoking (De Vries et al., 2003). Therefore this paper aims to contribute by identifying demographic risk factors by combining dataset from the CBS and RIVM in order to help better inform anti-smoking campaign policy

2. Theoretical Framework:

Smoking has been proven to have detrimental effects on a person’s health. Examples of these effects are various forms of cancer such as lung cancer, laryngeal cancer, upper digestive cancer, etc. (Gandini et al., 2008). Furthermore, it can lead to COPD and other ailments that decrease a person’s respiratory functioning (Forey, Thornton & Lee, 2011) Finally it can lower a person’s cardiovascular health through several complications (Villablanca, McDonald & Rutledge, 2000).

Once these ailments arise the smokers require extensive medical care. Based on a model created by Coyle et al. (2018), if individuals who are currently smoking were to stop in the next 12 months, then it could lead to a reduction in healthcare costs of €7054 over that individual's lifetime.

Based on the negative effects of smoking there have been efforts from the Dutch government to decrease the rate of smoking under the population. To do this effectively it is important to know which sections of the public have a higher rate of smoking. The largest governmental project we observed which was a collaborative effort of 6 countries concluded more research needed to be done on the social influences which affect smoking rates. (De Vries Mudde, Leijs, Charlton, Vartiainen,Buijs, ... & Kremers, S. 2003). Therefore, we start off with observing the smoking rates per neighborhood basis, in order to take into account social influence of the local environment. We theorize that different demographic characteristics of neighborhoods can reveal the different distributions of capital or other relevant characteristics of different neighborhoods and see if we find an association with increased incidence of smoking. This could show that unhealthy smoking behavior manifests in ‘high-risk’ areas where class, capital, and habitus intersect in a specific way (Bourdieu, 1986; ibid., 1987; ibid., 2007).

Therefore the aim of our research is to identify the demographic risk factors which are linked with an increased incidence of smoking which could help the government predict smoking rates, where the rate of smoking is not known beforehand. We classify neighborhoods based on smoking rate in 4 broad classes, which makes forming policy around these classes more feasible versus observing a smoking percentage per neighborhood and adjusting policy per percentage point. With this knowledge, the government can then deploy their resources to more efficiently run preventive anti-smoking campaigns, for example in newly built neighborhoods with low house prices (if this is indeed found to be a factor linked with smoking). This may have relevant practical implications, as our research may reveal hidden demographic factors which influence smoking rates not present in the current literature.

Research Question: Which demographic and neighbourhood characteristics are linked with increased incidence of smoking?

3. Methods

Data loading

We have used the neighborhood and district dataset from the CBS (CBS, 2020) and the health metrics dataset from the RIVM (RIVM, 2020). Both of the datasets contain data for the year 2020 and have the data on the level of individual neighborhoods. The RIVM dataset had more specific data than we required.We discarded the upper and lower bound of the 95% confidence interval and selected only the data containing all ages instead of separate age groups. This gave us two complete datasets, one from the CBS containing the demographic data and one from the RIVM containing the health metrics.

Pre-merger cleaning

Before we could merge the two datasets we had two clean them a little bit. The datasets contained both nominal and numerical data. However the numerical data was not recognised as numerical data by Pandas due to the formatting used by the CBS and the RIVM. It was easier to fix this for both datasets independently before merging than attempting to fix this after the merge. We inspected the datasets and noted the columns that should stay nominal. These columns were 5 columns related to the region identification for the CBS dataset and 8 columns related to region identification for the RIVM dataset. We also noted 2 columns that could be dropped from the CBS dataset since these related to postal codes and could reasonably be assumed to hold no real predictive power. After this we used the .to_numeric() method from the pandas module to convert the remaining columns in the datasets to floats. The reason why Pandas did not recognise this data as numeric in the first place was due to tab-spaces. Therefore we used the .str.strip() method to remove these beforehand. The missing values were coded in both datasets using the '.' character. Using the keyword parameter 'coerce' for error we set these cells to the appropriate NaN type, recognised by Pandas as missing features.

Merging

After the two datasets were sufficiently cleaned they were merged into one large dataset. To merge the datasets we used the neighborhood identification code that was present in both datasets. The code was present under the column named: ‘gwb_code’. Since both datasets were loaded using Pandas , we used the .merge() method to perform an inner join on the gwb_code.

In the RIVM dataset the column containing the identification code had to be renamed as it was called ‘Codering_3’. All the nominal data used to identify the neighborhoods was removed before merging but a separate labels DataFrame was created, containing a corresponding ID and the removed data, in case the identification of a neighborhood was required later on in the project. This appeared to not be the case however. The total number of rows after merging was 16983 with 144 columns, thus 143 features and 1 target.

Data cleaning

After the merge the dataset required further cleaning before models could be trained on it. The dataset still contained missing values and during manual inspection some rows turned out to have anomalous data. According to the variable description given by the CBS some of the variables had no data present in the CBS dataset. These variables related to energy usage by households, education level, labor participation and wage data. This data would possibly become available in the following years but at the moment could not be supplied. Since imputation on could not reasonably be done, due to all of the data missing for these features, they were dropped.

Then we checked the amount of columns that contained missing features and the amount of missing features per each of those columns. The columns with more than 2000 missing features were deemed excessive, all of these variables originally belonged to the CBS dataset and after consulting the variable explanation given by the CBS they were deemed to be most likely uninformative and thus subsequently dropped.

After this the columns with more than 1000 missing values were selected and inspected. This prompted us to look at the number of missing values per row. Some of the rows containing many missing values were inspected closer which showed that these were anomalous neighborhoods with extremely low population and some even zero. Since these were only few samples in the entire dataset and appeared to have no other reason for their anomalous data other than the extremely low or non-existent population, they were dropped.

Finally, we noticed that some of the rows had missing values for the target. These were not many samples and after consideration we decided to drop these samples since they did not constitute a large part of our data and we feared imputation could clutter the results.

We ended up with a DataFrame containing 16565 rows and 89 columns containing 88 features and 1 target variable.

Data checking

After the merge had been completed we ran a few checks. Firstly, using the descriptive tool provided by the Pandas module we checked the data and found no anomalies. To continue this we checked the number of missing values per column again and deemed them to be within acceptable limits and finally, we checked whether all of the cells in the dataset were numerical(integer or float) and confirmed that there were no errors during the cleaning process. We followed this up with a correlation matrix to check for correlations between the features and the target variable. This was done using the built in .corr() method provided by the Pandas module. This allows one to generate a correlation matrix for all of the columns in a Pandas DataFrame. The results of this was visualised using a heatmap generated with the Seaborn module which provides some tools to generate heatmaps.

Thirdly we plotted the target variable and the features in a histogram to check their distributions and ranges. The target variable was plotted in a separate plot and the feature distributions were plotted in a 11 by 8 grid.

Finally we used a simple linear regression to test the suitability of two versions of imputation, namely: median imputation and KNN imputation. Both seemed to return similar results which can be seen under appendix 1. Since both imputation methods seemed to return similar results, we chose to use the KNN imputer based on the reasoning that neighborhoods which shared similarities on most features were likely to have similar values on the missing features. This seemed slightly more useful from a theoretical perspective and was the main motivation for our choice.

Data splitting

The main goal of our research is the attempt to predict which regions in the Netherlands have a higher incidence of smoking than other regions. Our target variable, provided by the RIVM data set, was the percentage of smokers in a given neighborhood and thus continuous. We are interested in identifying areas with a high incidence of smoking behavior on the basis of demographics and neighbourhood health statistics. We were not interested in predicting the exact percentage of smokers in a region but the relative incidence. Therefore we decided to convert the target variable into a categorical variable before splitting the data into the training and test set.

To convert the target into a categorical variable we used the .qcut() method provided by the Pandas module. This turns an array containing numerical data into a specified amount of categorical labels based on the percentiles. We chose to work with quartiles since we were interested in finding the regions with higher proportions of smokers and believed that quartiles provided a good categorization based on the descriptive exploration.

After this we used the train_test_split function provided by the Sklearn module to split the feature array(X_array) and the target array(Y_array_Class) into two arrays, one for training and validation, the other for testing the final model, so four arrays in total.

We used the default splitting meaning that 75% of the model was used for training and 25% was used for testing. We refrained from using a smaller test set since we desired to have a better image of the generalizability instead of creating a model with the best performance. We hoped to partially achieve this by limiting the training set size so that the models might be less prone to overfitting

Model choice

We decided to train three different types of model, namely a linear model, a non-linear model and an ensemble model. We chose to train these three different types to prevent a potential false conclusion that the data could not be predicted well because it could not be fitted on a linear model, even though this might not be due to poor predictability but due to a non-linear relationship. The data could also perform badly on a non-linear model and to make sure that this was not due to bias or variance, we decided to include an ensemble model as well.

For our linear model we decided to use LinearSVC provided by the Sklearn module. As a non-linear model we decided to use an SVC using the RadialBasisFunction(RBF) kernel. We have considered using the KNN classifier as well based on the same reasoning for the KNNimputer but during the initial testing we saw a large difference in performance and chose to focus on the SVC instead. As for our ensemble method we used the RandomForestClassifier(RFC) also provided by the Sklearn module, since this is a relatively simple but well performing model widely used in the industry.

For each of our models we first created a pipeline with the KNNimputer and the model to ensure that every step happened in order, for the LinearSVC and the SVC model we also used the StandardScaler provided by the Sklearn module, since these models are known to be sensitive to scaling. We chose the StandardScaler because it is less sensitive to outliers compared to the MinMaxScaler and provides good performance. The pipeline was created using the make_pipeline function provided by Sklearn. The pipelines were then tuned and validated using GridsearchCV from Sklearn using 5-fold cross validation. The scoring used by the GridsearchCV was not changed and therefore the default scoring of the aforementioned models was used. All models had their random_state set to 0 during the creation of their pipeline to ensure reproducibility.

RandomForest

The first parameter we decided to tune for the RF pipeline was the K value of the imputer, we used the values 3,5 and 8. The other two parameters we tuned were the max_depth and the max_number of features. For the max_depth, we used 5,10,15 and 20 and for the max_features, we used 5,10,20 and 30. To illustrate the effect we plotted the training and validation scores using the cv_results from the GridsearchCV and pyplot from the Matplotlib module. To keep the illustration clear we separate out the graphs such that only one of the three parameters that were tuned is varying the other RandomForest parameter was averaged over. So, in the graph plotting the effect of the max_depth, one point represents the average score of all the max_feature settings. Finally we plotted the feature importances for the features given by the best model from the Gridsearch.

SVC

For the SVC model we also tuned the KNN imputer with the same values for K, namely 3,5 and 8. The two parameters we decided to tune were the value for C, which controls the amount of regularization of the model, and the setting for gamma, which the model uses during fitting when using the RBF kernel. For C we checked the log space between -2 and 5 with 9 steps and for gamma we tried the ‘scale’ and ‘auto’ setting. We plotted the training and validation scores in graphs that are separated out in such a way that only one of the parameters is varying. The SVC model does not provide feature coefficients or feature importances so to get the required feature importances for the feature importance plot, we used the permutation_importance function from the Sklearn module. This function estimates the importance of a feature by checking the decrease in the model score if that feature is randomly shuffled.

LinearSVC

For the LinearSVC model we also tuned the KNN imputer with the values 3,5 and 8. For this model the same regularization parameter C was tuned as for the SVC model with the RBF kernel. The other parameter we tuned was the type of penalty used during regularization, namely L1 and L2. We plotted the training and validation scores in separate graphs such that two of the tuned parameters were kept constant per graph. The linearSVC model does provide us with feature coefficients, however since we have 4 classes this gives a coefficient vector of shape (4,88). Since this is not easily converted into an appropriate measure of importance and we already used the permutation_importance for the SVC model, we decided to use that for the LinearSVC model as well for consistency.

Evaluation

During the model evaluation we looked at the scores presented by the classification_report function from the Sklearn model. We looked at the precision, recall and F1-score of the class for the highest incidence of smoking in particular, since this is our main group of interest. We looked at the regular score of the model, its parameters and given feature importance as well.

4. Results

Exploratory data analysis

During the exploratory data analysis we saw that some of the features did display big correlations with one another. This can be seen in the correlation matrix under appendix 2. We can see that the most and largest correlations occur in the top left section which corresponds to the part of the dataset that belonged to the CBS dataset. These features display population size and specific subsection size of the population. It is reasonable to assume a relation between these variables because a larger total population leads to a larger subsection in absolute numbers unless the proportions change drastically. Some of the RIVM features are intercorrelated as well but this was also expected. The correlation besides these are small although there appear to be some between the demographic and health measures

The target variable distribution can be seen for both the continuous (figure 1) and the categorical version (figure 2). As we can see the continuous version is relatively normally distributed with slight skewness to the right. The categorical variable is very balanced, which is expected due to our method of conversion using the quartiles. The RIVM features appear to be mostly normally distributed. Some of the CBS features show this as well with some of these having a slight skewness. However the variables relating to population appear to be heavily skewed, likely due to several large neighborhoods in urban areas.

RandomForest	**Precision**	**Recall**	**F1-score**	**Support**
High incidence	0.91	0.90	0.91	968
Medium incidence	0.68	0.70	0.69	991
Low incidence	0.76	0.74	0.75	1051
Very low incidence	0.84	0.85	0.85	1132

The table above shows the scores of the RandomForest model. We can see that the model scores really well for the High incidence class and also performs well for the very low incidence class. The performance for the two middle classes is slightly lower but still shows relatively good results. In the evaluation output under appendix 8.1 we can see that the model had an overall accuracy of 0.777, which is lower due to the worse performance of the middle classes. The parameters chosen by the GridsearchCV were, k = 8,max_features = 20 and max_depth = 20. Under appendix 5 we can see the plots showcasing the effect of varying different parameters. We can see that the plots do not appear to show significant differences between the different values of K. Increasing the values for max_features leads to a small increase in performance, but these performance increases appear to level off around 20. The distance between the test and training scores appears to remain constant and thus does not seem to affect over and under fitting to a large degree. Increasing values for max_depth also leads to performance increases but the distance between the test and training scores rapidly, therefore increasing the potential for overfitting as well. The feature importances was higher on average for the RIVM features but one outlier in the CBS features showed much higher importance than the average of the RIVM features, namely the average value of houses (g_woz). The other important features from the RIVM set were difficulty paying the bills (MoeiteMetRondkomen_38) and a high risk of depression or anxiety (HoogRisiscoOpAngstOfDepressie_25).

SVC	**Precision**	**Recall**	**F1-score**	**Support**
High incidence	0.91	0.90	0.90	968
Medium incidence	0.69	0.72	0.70	991
Low incidence	0.78	0.76	0.77	1051
Very low incidence	0.85	0.85	0.85	1132

The table above shows the results of the evaluation of the SVC model. The full results can be found under appendix 8.2. The model scores well on the upper and lower class similar to the RandomForest model with a slight decrease in performance for the middle classes although this decrease is slightly lower than the RandomForest. The overall accuracy of the model was 0.794, with the parameter settings chosen by the GridsearchCV being: K = 5, C= 31.62 and Gamma = ‘scale’. This is somewhat higher than the accuracy of the RandomForest. The effects of different parameter settings can be seen under appendix 6. We can see the plots for the different values for K and the different Gamma options show almost no differences. The main difference in training and test scores was seen for different values of C, which controls the regularization of the model. With an increase we see that the training scores keep increasing until it reaches around 1. Compare this to the test scores and we see that the performance increases at first and then slowly starts to decrease indicating that at moment the model starts to overfit on the training set. We can see in the feature importance plot that it displays high importance for the same three features discussed in the RandomForest section. However it also displays a high importance for the features related to weekly exercise, a physical disability and heavy drinking, and address density(WekelijkseSporters_6, BeperkingInBewegen_23, ZwareDrinker_14 and ste_oad)

LinearSVC	**Precision**	**Recall**	**F1-score**	**Support**
High incidence	0.83	0.92	0.88	968
Medium incidence	0.59	0.48	0.53	991
Low incidence	0.68	0.59	0.63	1051
Very low incidence	0.76	0.91	0.83	1132

Above are the results of the LinearSVC model. The full evaluation results can be found under appendix 8.3. We can see that the model scores well on the upper and lower class. However its performance is lower than the performance of the SVC and RandomForest models. It shows the same decrease in performance for the middle classes as the SVC and the RandomForest model. The decrease in performance is larger than the decrease displayed by the other two models. The overall accuracy of the model was 0.728, with the following parameters: K = 5, C = 398.11, penalty = ‘L2’. This accuracy is relatively low compared to the other two models. The plots showcasing the performance variance based on different parameter settings can be found under appendix 7. Based on these plots we can see that, similar to the previous two models, the different settings for the parameter K and the penalty do not affect the performance significantly. The main difference can be seen for varying levels of the parameter C, with higher values, and thus lower regularization corresponding to greater performance. The performance increase appears to level off at around 0.725. It is interesting to note that the L2 penalty does perform better at higher regularization as illustrated by the lower drop in scores when the parameter C is decreased. The plot showing the feature importance looks quite different to the previous two models. The importance of the RIVM features is significantly lower compared to the SVC and RandomForest models. It gives more importance on average to the CBS features, which is the opposite for the other two models. The largest importance is given to the features related to housing prices.

Conclusion

Based on the results we can conclude that the non-linear models perform better on this dataset. This could be due to an inability to separate the data linearly. This is also supported by the fact that the training scores did not reach above 0.75 for the linear model even with a very low regularization. The SVC model performs slightly better than the RandomForest model and should thus be prefered. The decreased performance for all of the models on predicting the two middle classes might be due to the method we used to create the different categories. By using the quartiles, the middle classes are quite close together in absolute value, perhaps by using a different categorization method we could increase the performance. However, we were most interested in the upper and lower classes and the models performed well on this. Finally we saw that the best performing models gave the most importance to the health measures, with one interesting outlier in the demographic data. Our data has shown that the incidence of smoking in a neighborhood could accurately be predicted using measures related to difficulty paying bills, stress and average housing prices. Based on our findings the Dutch government could attempt to focus on the poorer regions of the country in its effort to decrease the total number of smokers as the most progress stands to be gained in these regions.

Bibliography:

Bernhard, D., Moser, C., Backovic, A., & Wick, G. (2007). Cigarette smoke–an aging accelerator?. *Experimental gerontology*, *42*(3), 160-165.

Bourdieu, Pierre. (1986). *The Forms of Capital*. In J. G. Richardson (Red.). New York: Greenwood Press.

Bourdieu, Pierre. (1987). What Makes A social Class? On The Theoretical and Practical Existence Of Groups. *Berkeley Journal of Sociology*, *32*, 1-17.

Bourdieu, Pierre. (2007). The Contradictions in Inheritance. In P. Bourdieu (Coord.), Misery in the World (pp. 587-593). Petrópolis: Voices.

Coyle, K., Coyle, D., Lester‐George, A., West, R., Nemeth, B., Hiligsmann, M., ... & EQUIPT StudyGroup. (2018). Development and application of an economic model (EQUIPTMOD) to assess the impact of smoking cessation. *Addiction*, *113*, 7-18.

CBS, Centraal Bureau voor de Statistiek. (2020). Kerncijfers wijken en buurten 2020. Produced and distributed by CBS, 84799NED,
https://opendata.cbs.nl/statline/portal.html?_la=nl&_catalog=CBS&tableId=84799NED&_theme=233

De Vries, H., Mudde, A., Leijs, I., Charlton, A., Vartiainen, E., Buijs, G., ... & Kremers, S. (2003). The European Smoking prevention Framework Approach (EFSA): an example of integral prevention. *Health education research*, *18*(5), 611-626.

Forey, B. A., Thornton, A. J., & Lee, P. N. (2011). Systematic review with meta-analysis of the epidemiological evidence relating smoking to COPD, chronic bronchitis and emphysema. *BMC pulmonary medicine*, *11*(1), 1-61.

Gandini, S., Botteri, E., Iodice, S., Boniol, M., Lowenfels, A. B., Maisonneuve, P., & Boyle, P. (2008). Tobacco smoking and cancer: a meta‐analysis. *International journal of cancer*, *122*(1), 155-164.

Goldenberg, M., Danovitch, I., & IsHak, W. W. (2014). Quality of life and smoking. *The American journal on addictions*, *23*(6), 540-562.

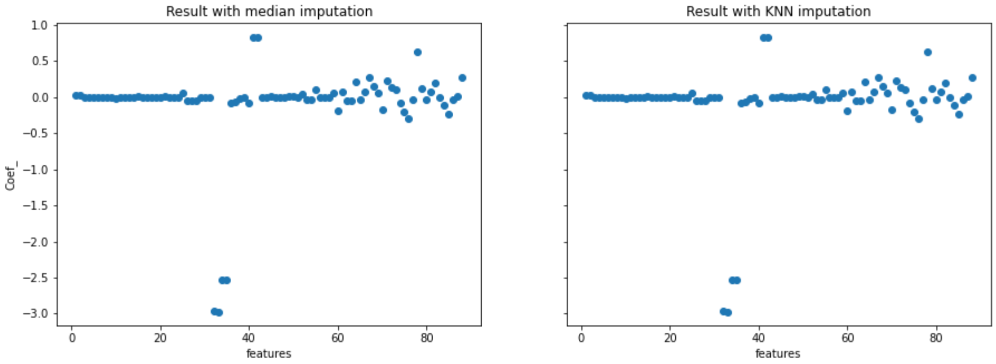
RIVM, Rijksinstituut Voor Volksgezondheid En Milieu. (2020). Gezondheid per wijk en buurt 2020. Produced and distributed by RIVM, 50090NED,
https://statline.rivm.nl/portal.html?_la=nl&_catalog=RIVM&tableId=50090NED&_theme=85

Villablanca, A. C., McDonald, J. M., & Rutledge, J. C. (2000). Smoking and cardiovascular disease. *Clinics in chest medicine*, *21*(1), 159-172.

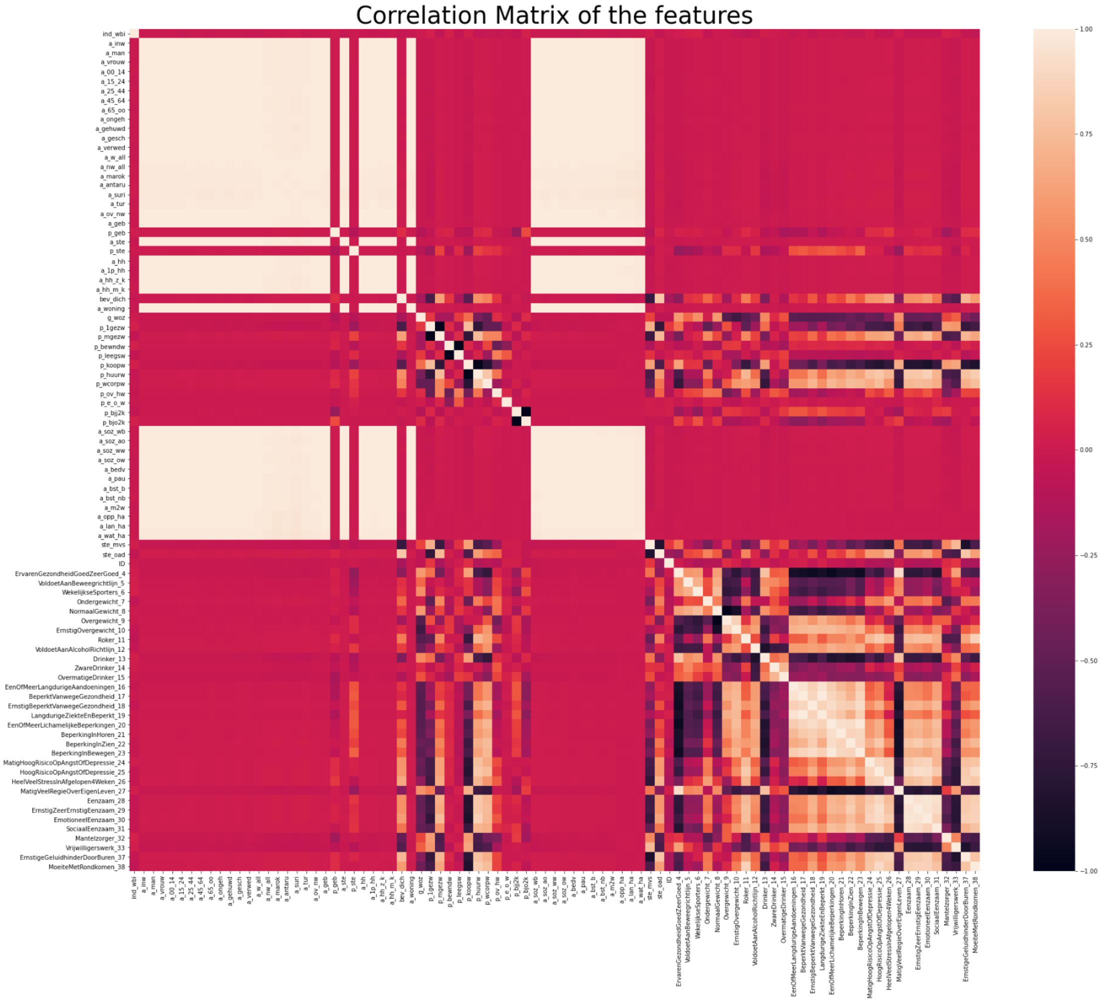
Xu, X., Bishop, E. E., Kennedy, S. M., Simpson, S. A., & Pechacek, T. F. (2015). Annual healthcare spending attributable to cigarette smoking: an update. *American journal of preventive medicine*, *48*(3), 326-333.

APPENDIX

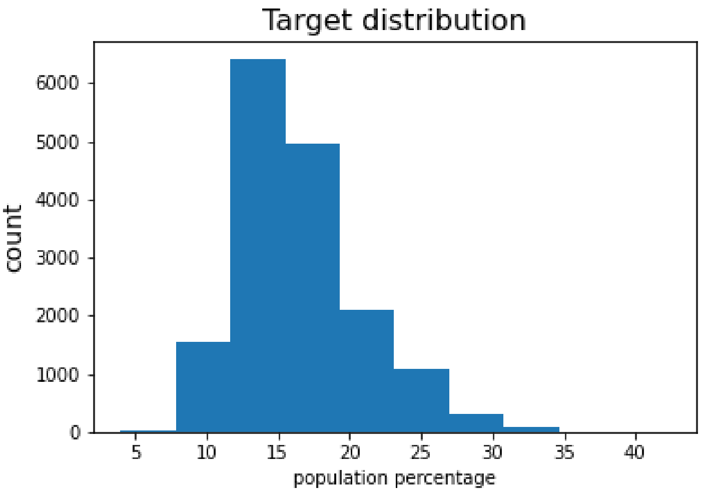
1. KNN imputation test

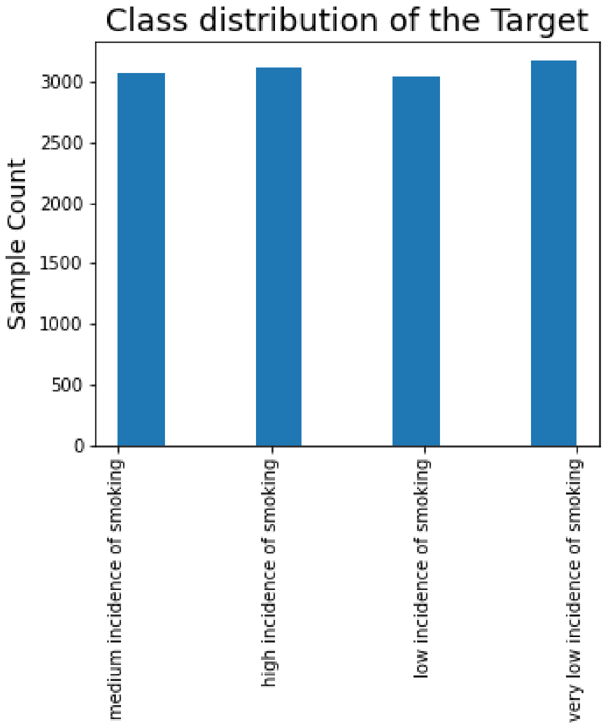


2. Correlation matrix

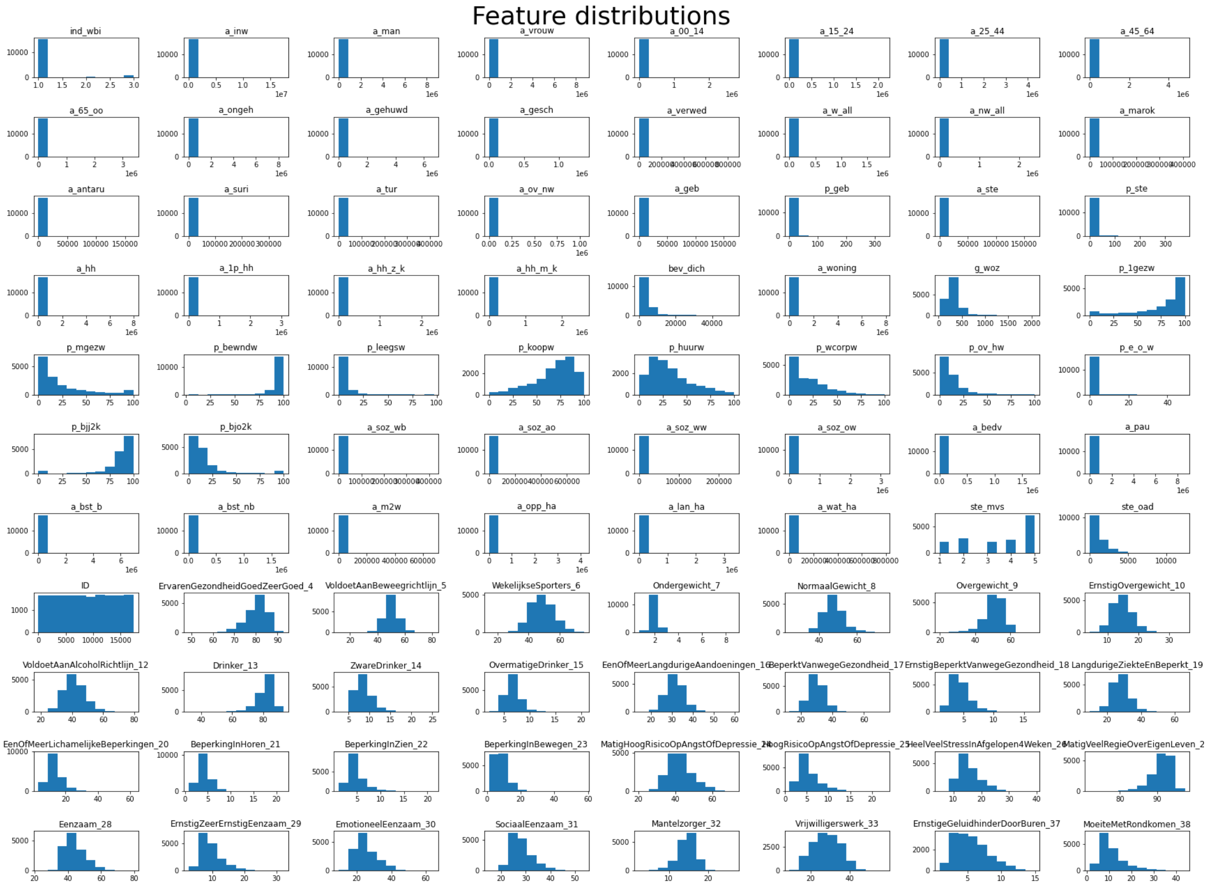


3. Target distributions

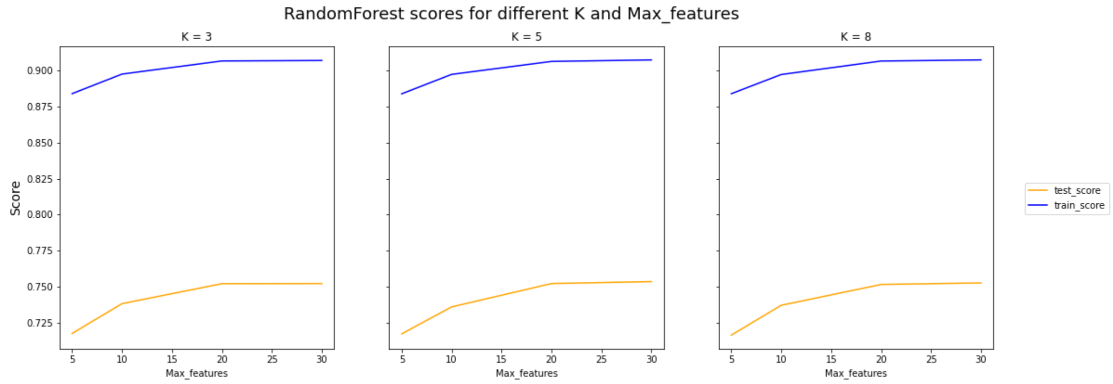


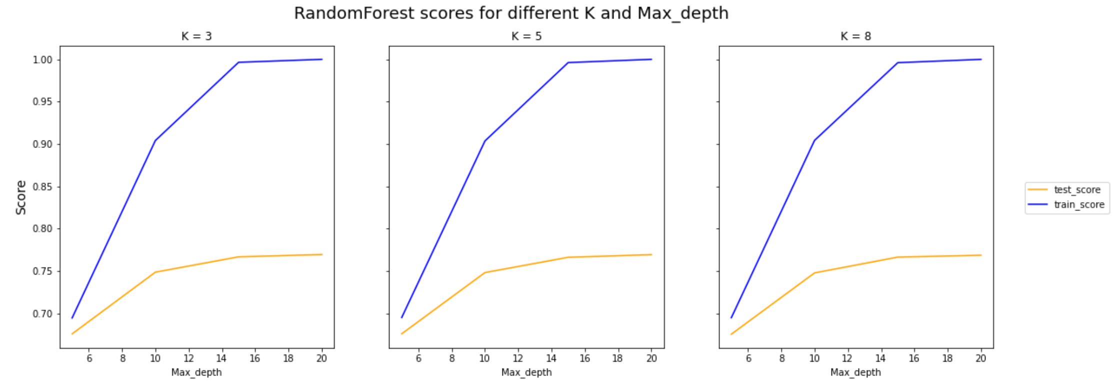


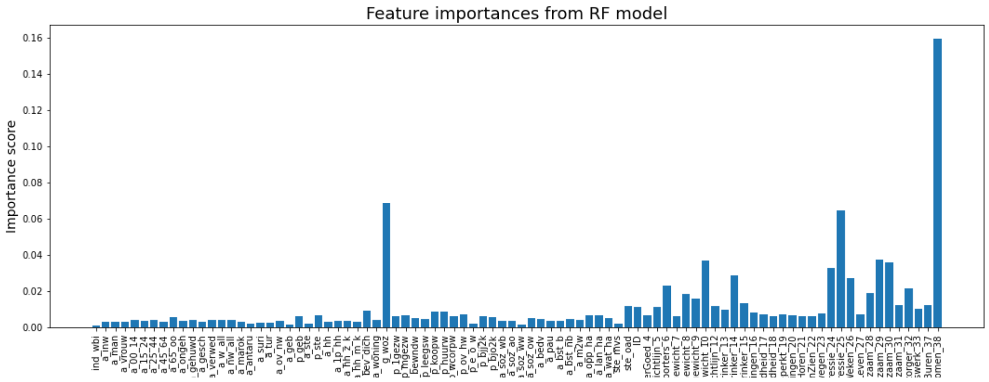
4. Feature distributions



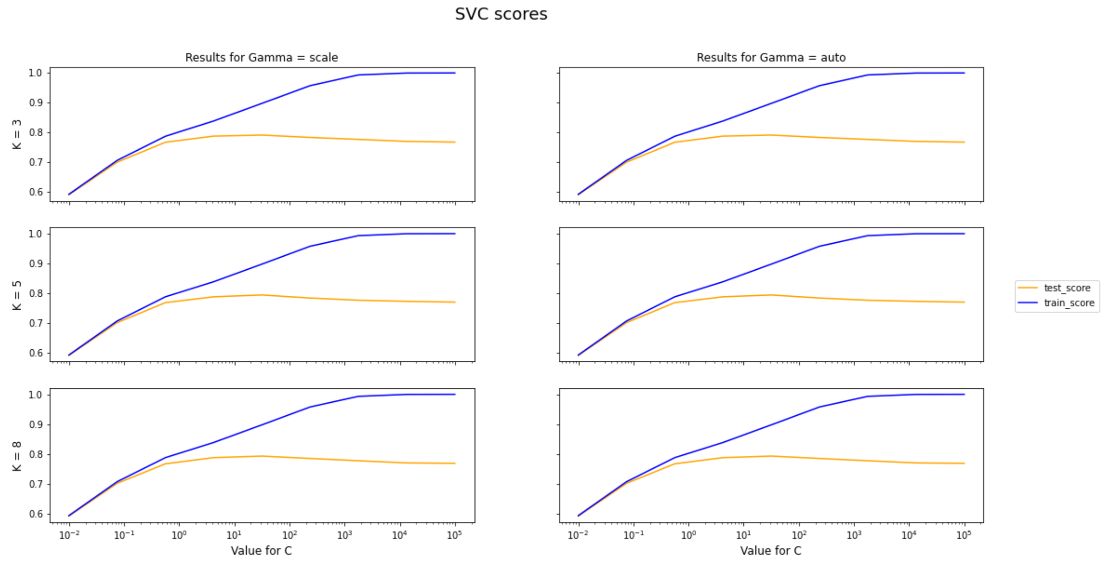
5. RandomForest model parameter plots and feature importances

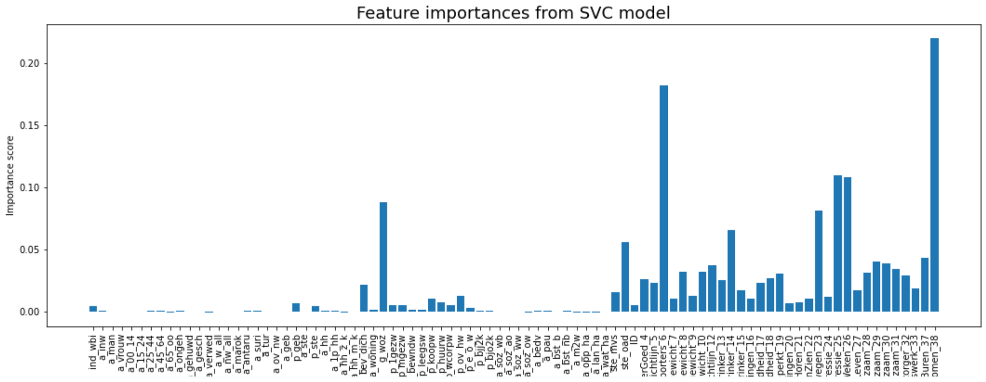




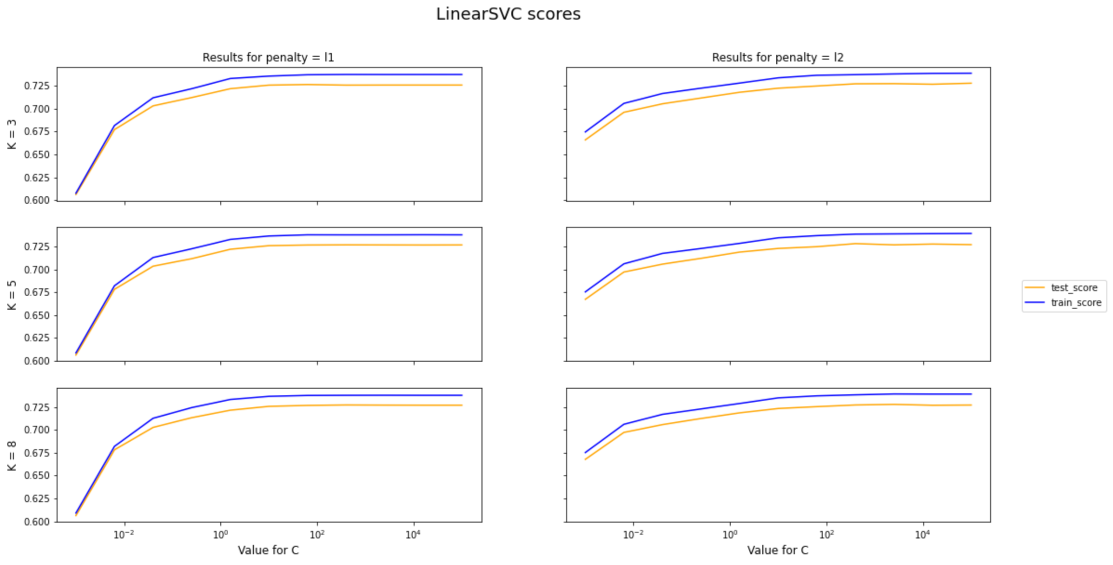


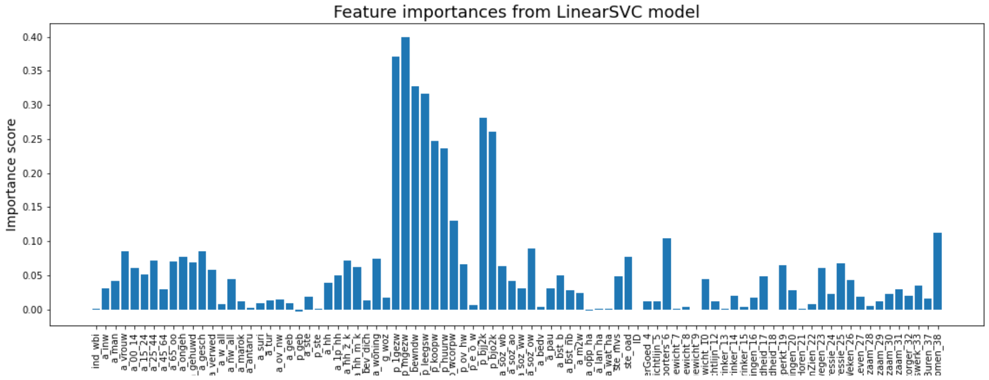
6. SVC model parameter plots and feature importances





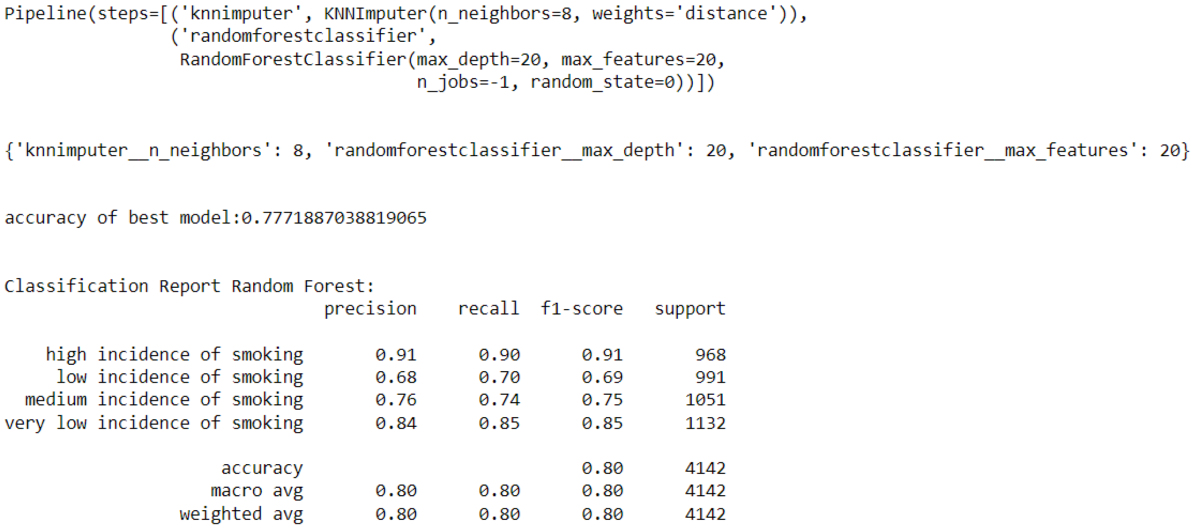
7. LinearSVC model parameter plots and feature importances





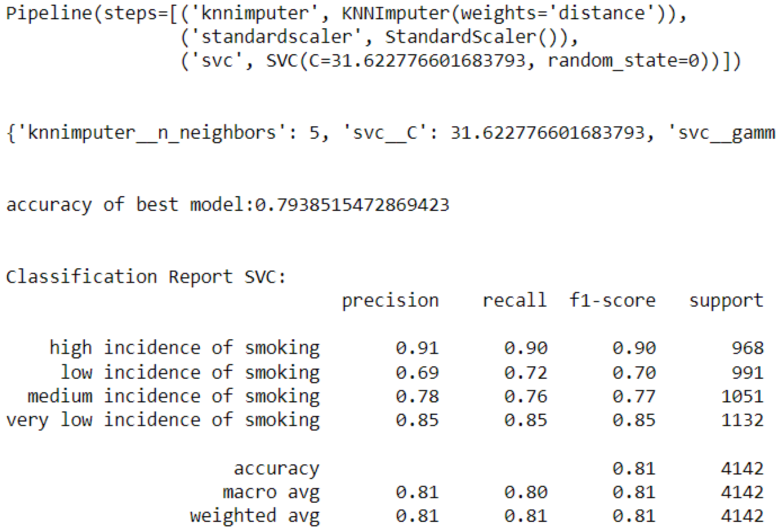
8. Evaluation outputs

8.1 RandomForest



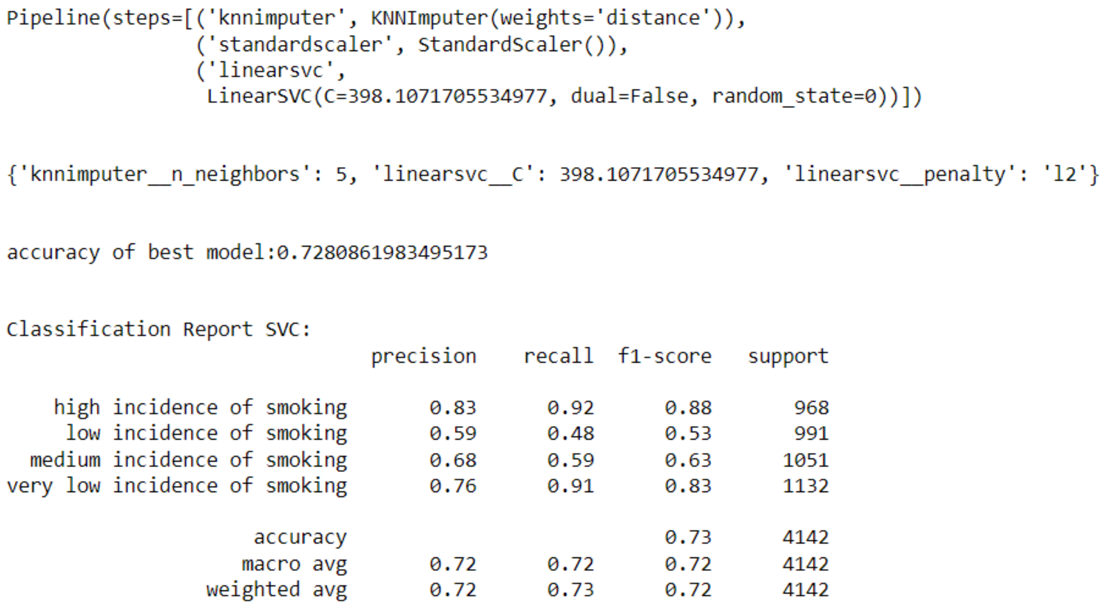
```
Pipeline(steps=[('knnimputer', KNNImputer(n_neighbors=8, weights='distance')),
                ('randomforestclassifier',
                 RandomForestClassifier(max_depth=20, max_features=20,
                                        n_jobs=-1, random_state=0))])


{'knnimputer__n_neighbors': 8, 'randomforestclassifier__max_depth': 20, 'randomforestclassifier__max_features': 20}


accuracy of best model:0.7771887038819065


Classification Report Random Forest:
                              precision    recall  f1-score   support

     high incidence of smoking       0.91      0.90      0.91       968
      low incidence of smoking       0.68      0.70      0.69       991
   medium incidence of smoking       0.76      0.74      0.75      1051
 very low incidence of smoking       0.84      0.85      0.85      1132

                      accuracy                           0.80      4142
                     macro avg       0.80      0.80      0.80      4142
                  weighted avg       0.80      0.80      0.80      4142
```

8.2 SVC



```
Pipeline(steps=[('knnimputer', KNNImputer(weights='distance')),
                ('standardscaler', StandardScaler()),
                ('svc', SVC(C=31.622776601683793, random_state=0))])


{'knnimputer__n_neighbors': 5, 'svc__C': 31.622776601683793, 'svc__gamma': 'scale'}


accuracy of best model:0.7938515472869423


Classification Report SVC:
                              precision    recall  f1-score   support

     high incidence of smoking       0.91      0.90      0.90       968
      low incidence of smoking       0.69      0.72      0.70       991
   medium incidence of smoking       0.78      0.76      0.77      1051
 very low incidence of smoking       0.85      0.85      0.85      1132

                      accuracy                           0.81      4142
                     macro avg       0.81      0.80      0.81      4142
                  weighted avg       0.81      0.81      0.81      4142
```

8.3 LinearSVC



```
Pipeline(steps=[('knnimputer', KNNImputer(weights='distance')),
                ('standardscaler', StandardScaler()),
                ('linearsvc',
                 LinearSVC(C=398.1071705534977, dual=False, random_state=0))])


{'knnimputer__n_neighbors': 5, 'linearsvc__C': 398.1071705534977, 'linearsvc__penalty': 'l2'}


accuracy of best model:0.7280861983495173


Classification Report SVC:
                               precision    recall  f1-score   support

    high incidence of smoking       0.83      0.92      0.88       968
     low incidence of smoking       0.59      0.48      0.53       991
  medium incidence of smoking       0.68      0.59      0.63      1051
very low incidence of smoking       0.76      0.91      0.83      1132

                     accuracy                           0.73      4142
                    macro avg       0.72      0.72      0.72      4142
                 weighted avg       0.72      0.73      0.72      4142
```