skimpy summary

Data Summary

dataframe	Values
Number of rows	73525
Number of columns	50

Data Types

Column Type	Count
string	37
int64	13

number

column_name	NA	NA %	mean	sd	p0	p25	p50	p75	p100	hist
encounter_id	0	0	170000000	100000000	13000	85000000	150000000	230000000	440000000	
patient_nbr	0	0	54000000	39000000	140	23000000	46000000	87000000	190000000	
admission_type_id	0	0	2	1.4	1	1	1	3	8	
discharge_disposition_id	0	0	3.7	5.3	1	1	1	4	28	
admission_source_id	0	0	5.7	4.1	1	1	7	7	25	
time_in_hospital	0	0	4.4	3	1	2	4	6	14	
num_lab_procedures	0	0	43	20	1	31	44	57	130	
num_procedures	0	0	1.3	1.7	0	0	1	2	6	
num_medications	0	0	16	8.1	1	10	15	20	81	
number_outpatient	0	0	0.37	1.3	0	0	0	0	40	
number_emergency	0	0	0.2	0.93	0	0	0	0	76	
number_inpatient	0	0	0.64	1.3	0	0	0	1	19	
number_diagnoses	0	0	7.4	1.9	1	6	8	9	16	

string

column_name	NA	NA %	words per row	total words
race	1626	2.21	0.98	71899
gender	2	0	1	73523
age	0	0	1	73525
weight	71247	96.9	0.031	2278
payer_code	29062	39.53	0.6	44463
medical_specialty	36076	49.07	0.51	37449
diag_1	14	0.02	1	73511
diag_2	256	0.35	1	73269
diag_3	1046	1.42	0.99	72479
max_glu_serum	69727	94.83	0.052	3798
A1Cresult	61134	83.15	0.17	12391
metformin	0	0	1	73525
repaglinide	0	0	1	73525
nateglinide	0	0	1	73525
chlorpropamide	0	0	1	73525
glimepiride	0	0	1	73525
acetohexamide	0	0	1	73525
glipizide	0	0	1	73525
glyburide	0	0	1	73525
tolbutamide	0	0	1	73525
pioglitazone	0	0	1	73525
rosiglitazone	0	0	1	73525
acarbose	0	0	1	73525
miglitol	0	0	1	73525
troglitazone	0	0	1	73525
tolazamide	0	0	1	73525
examide	0	0	1	73525
citoglipton	0	0	1	73525
insulin	0	0	1	73525
glyburide-metformin	0	0	1	73525
glipizide-metformin	0	0	1	73525
glimepiride-pioglitazone	0	0	1	73525
metformin-rosiglitazone	0	0	1	73525
metformin-pioglitazone	0	0	1	73525
change	0	0	1	73525
diabetesMed	0	0	1	73525
readmitted	0	0	1	73525

End

Table 1: Descriptive statistics of the training set before any preprocessing. Output is obtained using skimpy in code cell 8. We observe high missing rate for variables like weight. Most numerical variables follow a non-Gaussian, and either left/right skewed distribution.

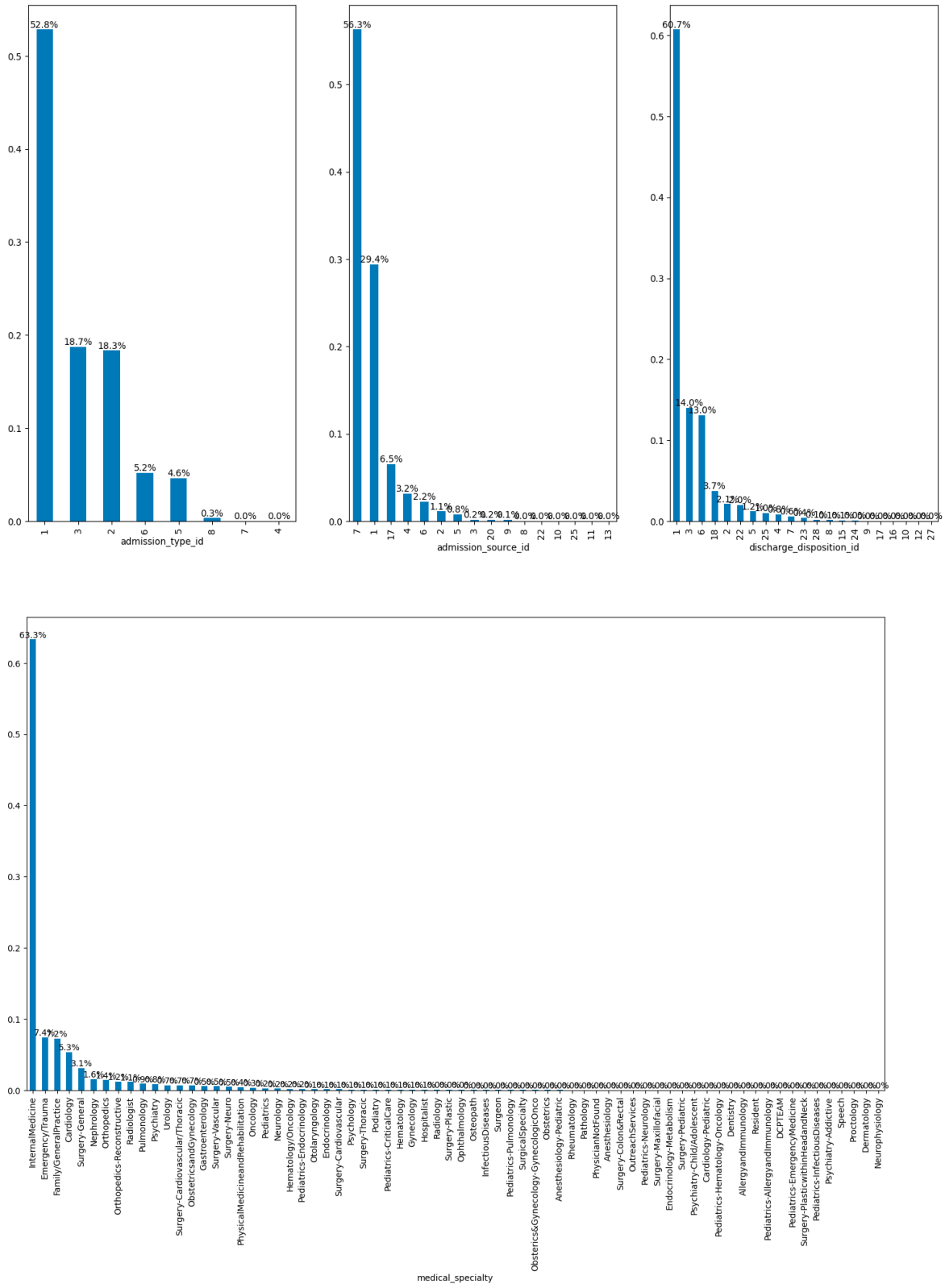
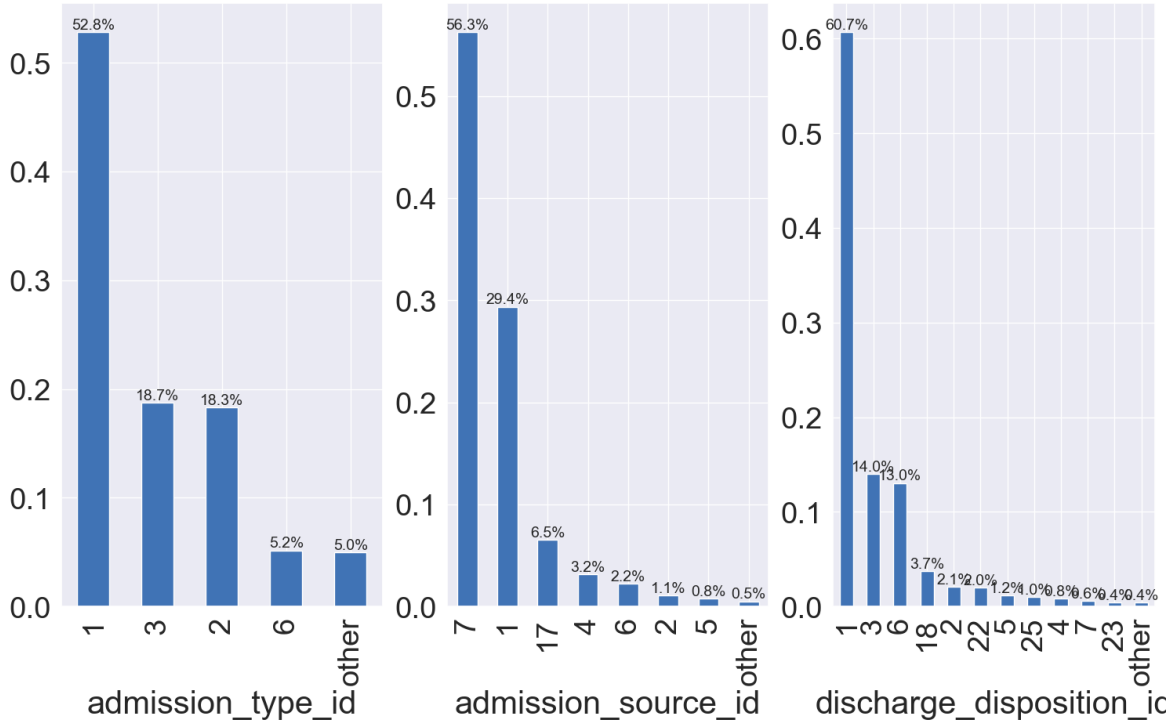


Figure 1: The relative proportion of each category for each of the 4 variables ‘admission type’, ‘admission source’, ‘disposition ID’ and ‘medical specialty’. For ‘disposition ID’, we note that categories 11, 13, 14, 19, 20, and 21 are removed since they’re related to patients who can’t be readmitted due to death or hospice. We note that in the training set, not all categories for each variable are present, e.g., ‘admission source’ has 21 unique values, but only 16 appear by

chance in the training set. We notice a left skew in each case, denoting a severe category imbalance. The top 3 panels are ‘admission type’, ‘admission source’, and ‘disposition ID’; to check what category each number encodes in the x-axis, please see (DeShazo et al., 2014). The lower panel corresponds to medical specialty.



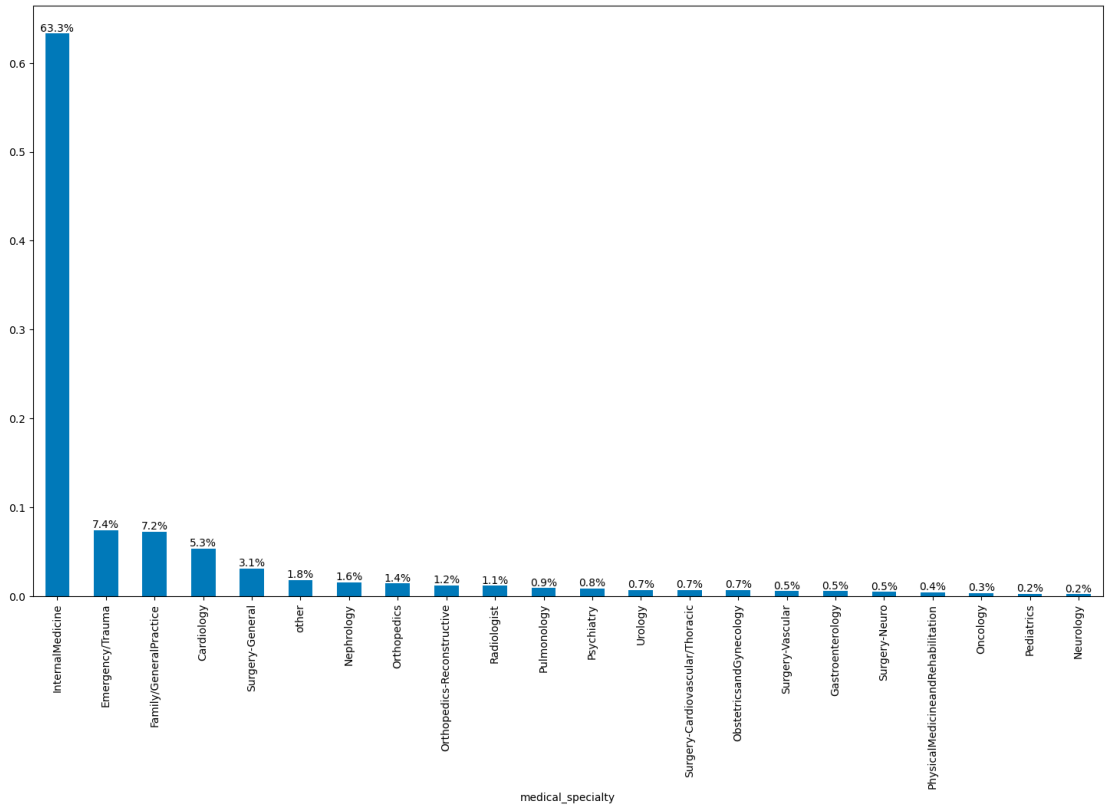


Figure 2: the relative proportion of each category for each of the 4 variables 'admission type', 'admission source', 'disposition ID' and 'medical specialty' after regrouping low appearing categories into "other" to simplify the complexity of this feature. 'Admission type' originally has 8 categories, and we keep only 5 where each has a proportion greater than a threshold of 5%. For 'admission source', there were 16 categories, where 9 were grouped into other based on a threshold of 0.1%. 'Disposition ID' had 21 categories, where 10 were grouped if each had a proportion below a threshold of 0.004%. 'Medical specialty' has 70 unique values, where 49 are grouped into 'Other' as each has a proportion below a threshold of 0.000085%. For 'medical specialty', the cumulative proportion of the 49 grouped variables makes 'Other' the 6th highest appearing category. We again emphasize that in the training set, not all categories for each variable are present, e.g., 'admission source' has 21 unique values, but only 16 appear by chance in the training set. What this implies is that in the val/test set, some of the categories that are grouped may have higher proportion than in the train set, and thus some information is lost due to grouping. This is a trade-off we accept for the benefits of reduced model complexity and sparsity when one-hot encoding the variables. The top 3 panels are 'admission type', 'admission source', and 'disposition ID'; to check what category each number encodes in the x-axis, please see (DeShazo et al., 2014). The lower panel corresponds to medical specialty. See code cell 21 for more implementation details.

ICD-9 Codes	Descriptions
390-459,785	Diseases of the circulatory system
460-519,786	Diseases of the respiratory system
520-579,787	Diseases of the digestive system
250.xx	Diabetes mellitus
800-999	Injury and poisoning
710-739	Diseases of the musculoskeletal system and connective tissue
580-629,788	Diseases of the genitourinary system
140-239	Neoplasms
780,781,784,790-799	Other symptoms, signs, and ill-defined conditions
240-279, excluding 250	Endocrine, nutritional, and metabolic diseases and immunity disorders, without diabetes
680-709,782	Diseases of the skin and subcutaneous tissue
001-139	Infectious and parasitic diseases
290-319	Mental disorders
E-V	External causes of injury an supplemental classification
280-289	Diseases of the blood and blood-forming organs
320-359	Disease of the nervous system

Table 2: ICD coding scheme extracted from Kaggle. Any ICD-9 codes falling outside the above table is categorized as "Unclassified". See code cell 27.

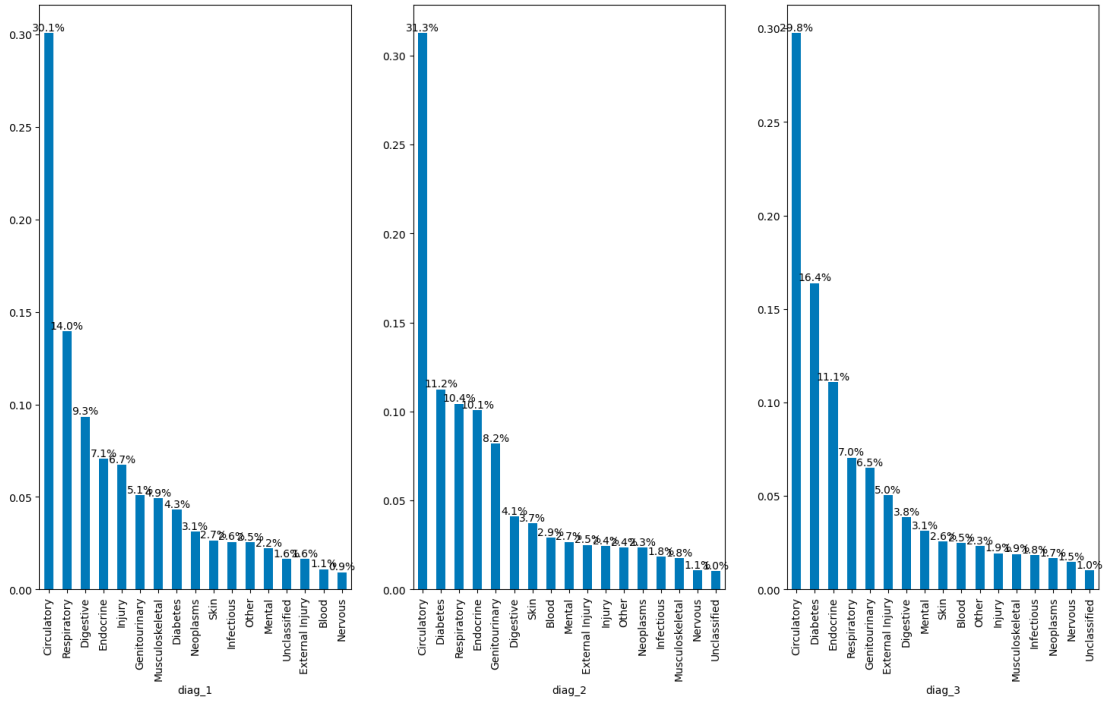


Figure 3: proportion per category for each of the diagnosis feature encoded following the ICD-9 coding scheme. The coding happens in code cell 27.

	time_in_hospital	num_lab_procedures	num_procedures	num_medications	number_outpatient	number_emergency	number_inpatient	number_diagnoses	readmitted
min	1.00	1.00	0.00	1.00	0.00	0.00	0.00	1.00	0.00
max	14.00	129.00	6.00	81.00	40.00	76.00	19.00	16.00	1.00
mean	4.38	42.92	1.34	15.98	0.37	0.20	0.63	7.40	0.47
std	2.98	19.65	1.71	8.09	1.28	0.94	1.27	1.94	0.50

Table 3: Summary stats of numerical features of training set. We observe no negative values or physically impossible values. There are some unlikely numbers like 129 lab procedures, however, close examination of that patient show that she has plausible values for the other features (e.g. 45 medications, 1 outpatient and 1 inpatient visit, 8 number of diagnoses, etc.), thus it's unreasonable to drop her because one feature's value is extreme. The same applies to other patients with an extreme value like 70 emergency visits despite a mean of less than 1.

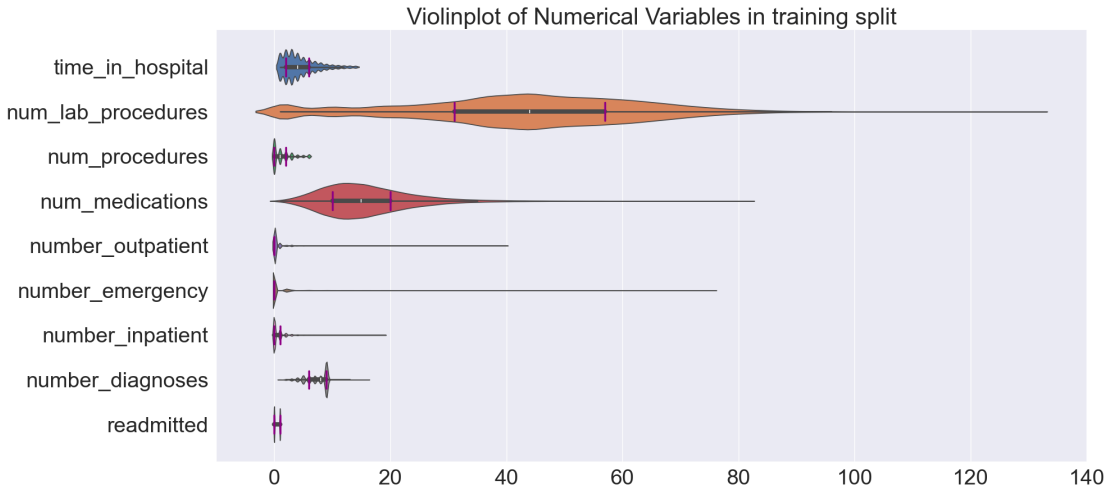


Figure 4: Violin plots per numerical feature in the train set showing the distribution of values per feature. In purple vertical lines we annotate the 25th and 75th percentile per feature.

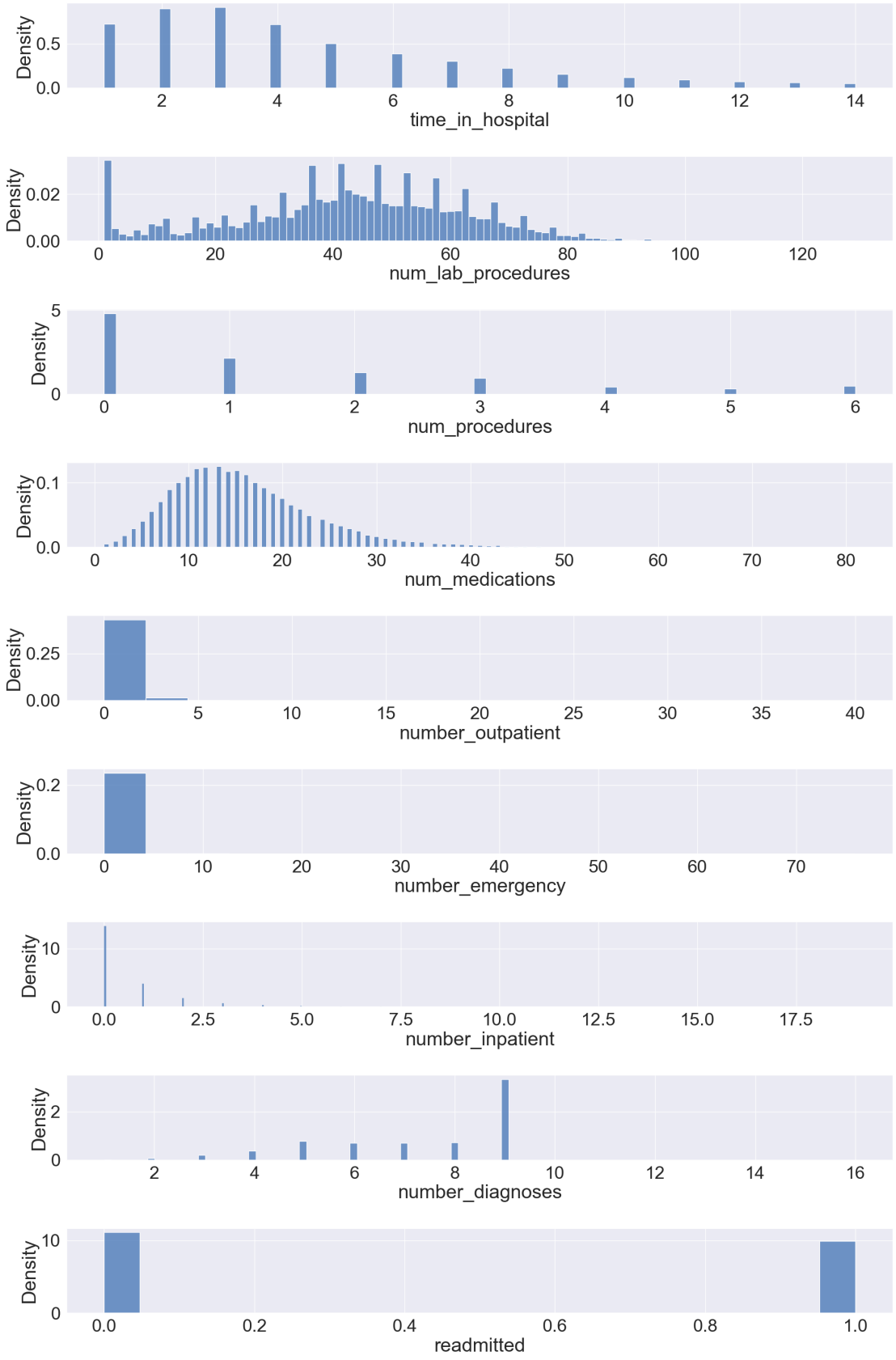


Figure 5: Histogram of the relative density of the values per numerical feature in the train set.

skimpy summary

Data Summary

dataframe	Values
Number of rows	71771
Number of columns	31

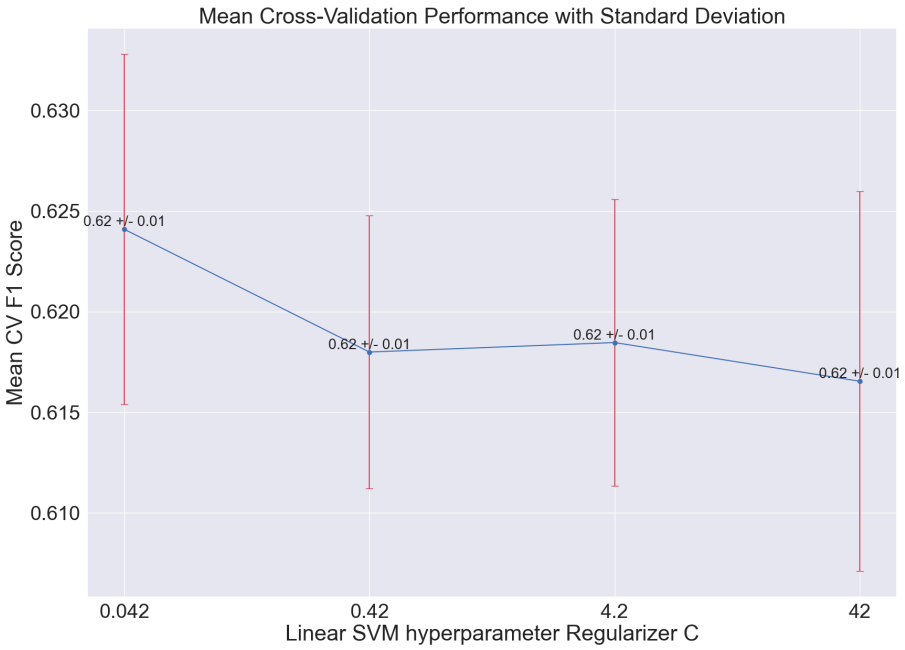
Data Types

Column Type	Count
float64	23
int64	8

number

column_name	NA	NA %	mean	sd	p0	p25	p50	p75	p100	hist
time_in_hospital	0	0	4.4	3	1	2	4	6	14	
num_lab_procedures	0	0	43	20	1	31	44	57	130	
num_procedures	0	0	1.3	1.7	0	0	1	2	6	
num_medications	0	0	16	8.1	1	10	15	20	81	
number_outpatient	0	0	0.37	1.3	0	0	0	0	40	
number_emergency	0	0	0.2	0.94	0	0	0	0	76	
number_inpatient	0	0	0.63	1.3	0	0	0	1	19	
number_diagnoses	0	0	7.4	1.9	1	6	8	9	16	
age_[70-80)	0	0	0.25	0.44	0	0	0	1	1	
age_[80-90)	0	0	0.16	0.37	0	0	0	0	1	
admission_type_id_1	0	0	0.53	0.5	0	0	1	1	1	
admission_type_id_3	0	0	0.19	0.39	0	0	0	0	1	
admission_type_id_6	0	0	0.052	0.22	0	0	0	0	1	
discharge_disposition_id_1	0	0	0.61	0.49	0	0	1	1	1	
discharge_disposition_id_6	0	0	0.13	0.34	0	0	0	0	1	
admission_source_id_1	0	0	0.29	0.46	0	0	0	1	1	
admission_source_id_4	0	0	0.032	0.17	0	0	0	0	1	
medical_specialty_Emergency/Trauma	0	0	0.074	0.26	0	0	0	0	1	
medical_specialty_ObstetricsandGynecology	0	0	0.0067	0.081	0	0	0	0	1	
diag_1_Endocrine	0	0	0.071	0.26	0	0	0	0	1	
diag_1_Musculoskeletal	0	0	0.049	0.22	0	0	0	0	1	
diag_1_Neoplasms	0	0	0.031	0.17	0	0	0	0	1	
diag_2_Diabetes	0	0	0.11	0.32	0	0	0	0	1	
diag_3_Diabetes	0	0	0.16	0.37	0	0	0	0	1	
diag_3_Genitourinary	0	0	0.065	0.25	0	0	0	0	1	
metformin_No	0	0	0.8	0.4	0	1	1	1	1	
insulin_Down	0	0	0.12	0.32	0	0	0	0	1	
insulin_No	0	0	0.47	0.5	0	0	0	1	1	
insulin_Up	0	0	0.11	0.31	0	0	0	0	1	
change_Ch	0	0	0.46	0.5	0	0	0	1	1	
diabetesMed_No	0	0	0.23	0.42	0	0	0	0	1	

Table 4: Descriptive statistics of predictor variables $\boldsymbol{X}$ after preprocessing, one-hot encoding, and variable selection. Statistics of the readmission variable y are not shown as they're the same as in Figures 4 and 5.



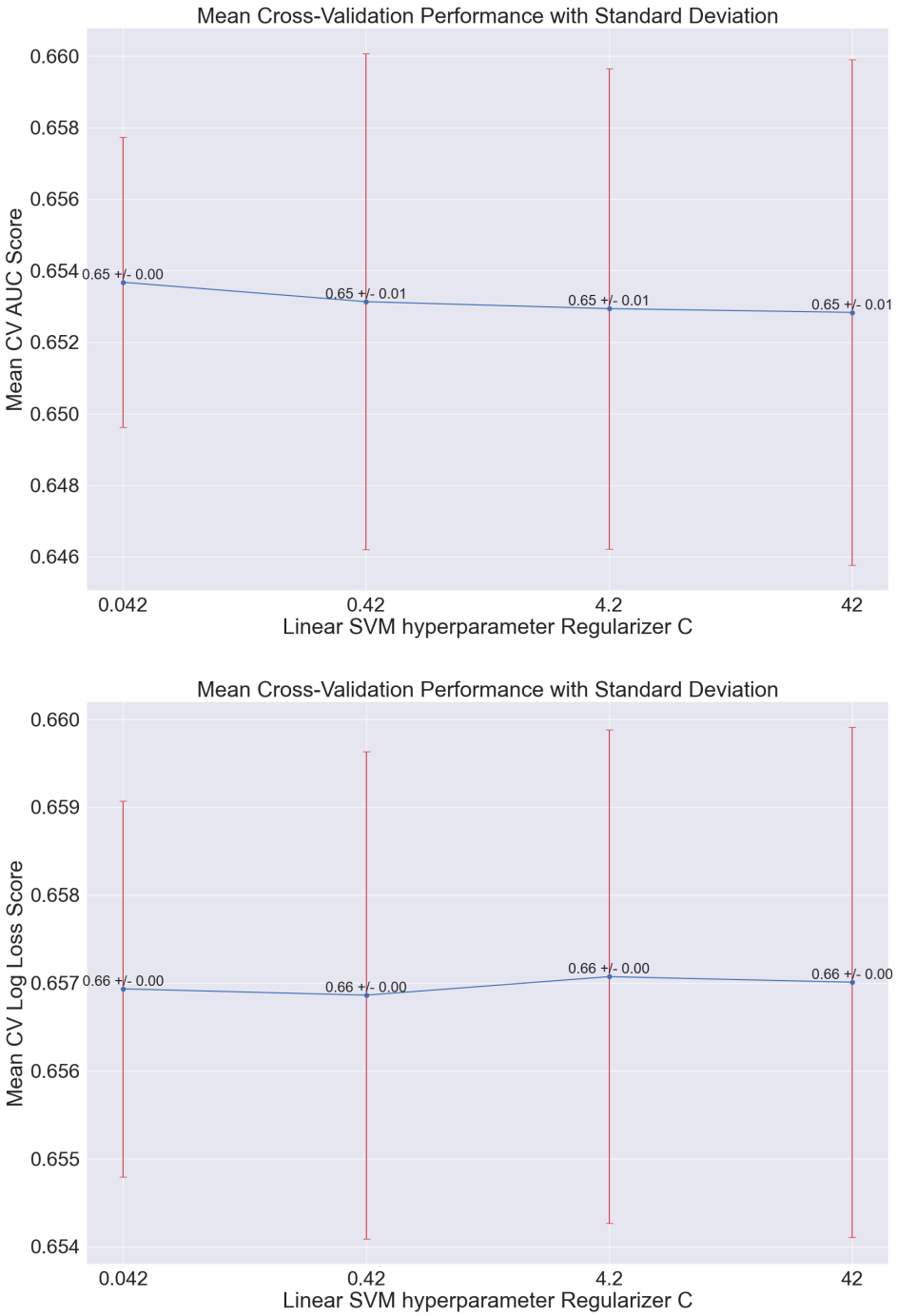
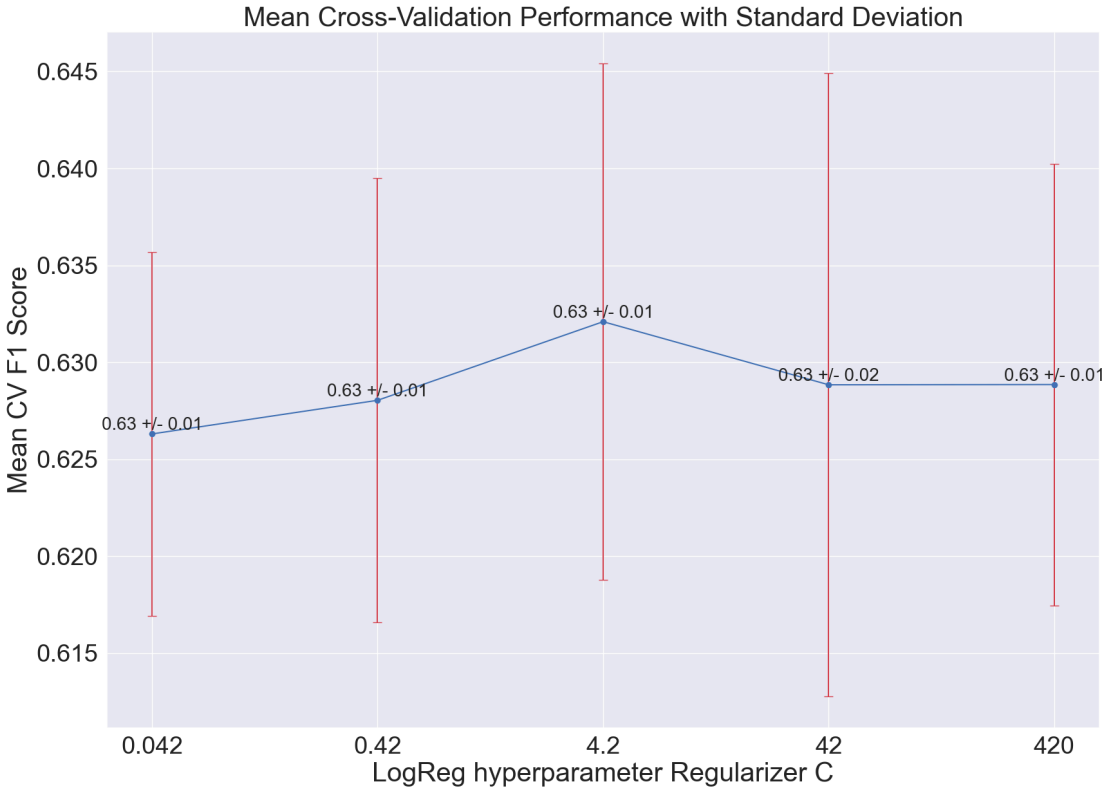
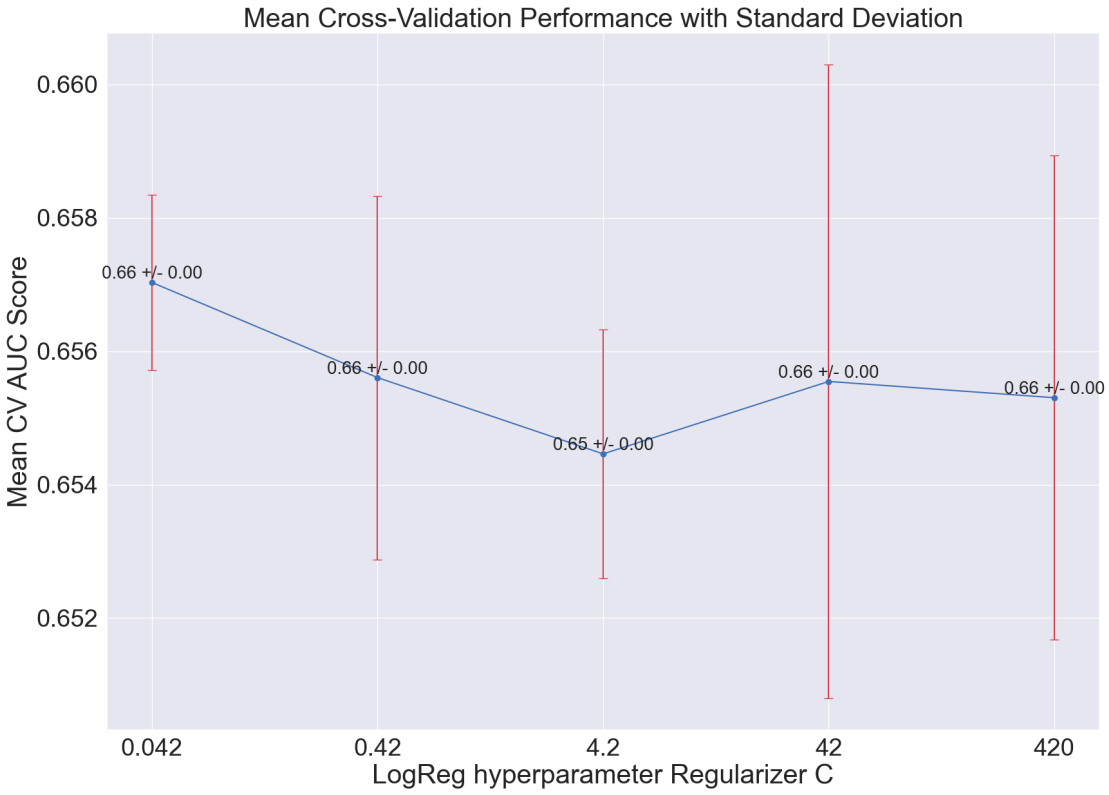


Figure 6: mean CV metrics with standard deviation (red lines) as a function for 4 hyperparameters {0.042, 0.42, 4.2, 42} for the linear SVM. The regularizer controls the SVM's complexity, and influences whether it overfits, its tolerance to outliers and the margin width of the decision boundary. From top to bottom we report the mean CV F1 score, CV AUC score and log loss. We consistently observe a very high standard deviation, indicating unstable predictions which could result from the linear kernel being unable to capture the complicated dependencies amongst input features for predicting readmission rate. Preliminary experiements (not shown) show that a more expressive RBF kernel yields lower standard deviation in CV scores. Both F1 and AUC

hover around 0.6, which is a reasonable good performance, at least much better than random guessing the need for hospital readmission.





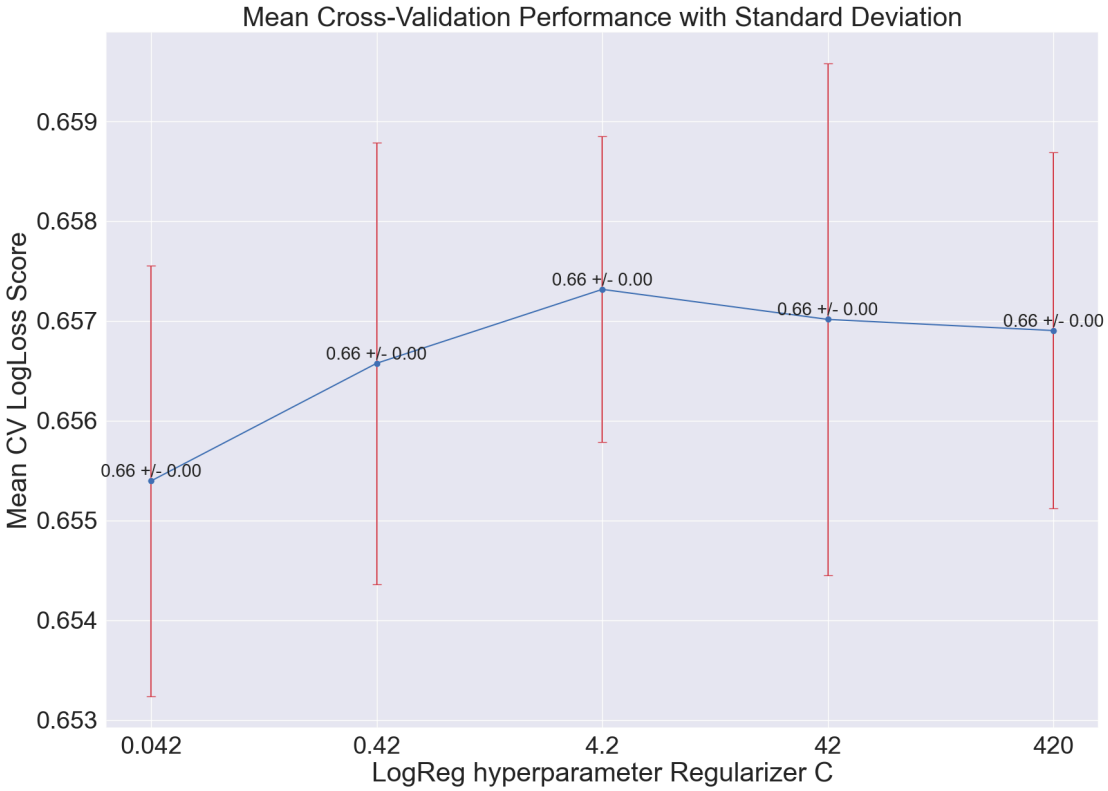
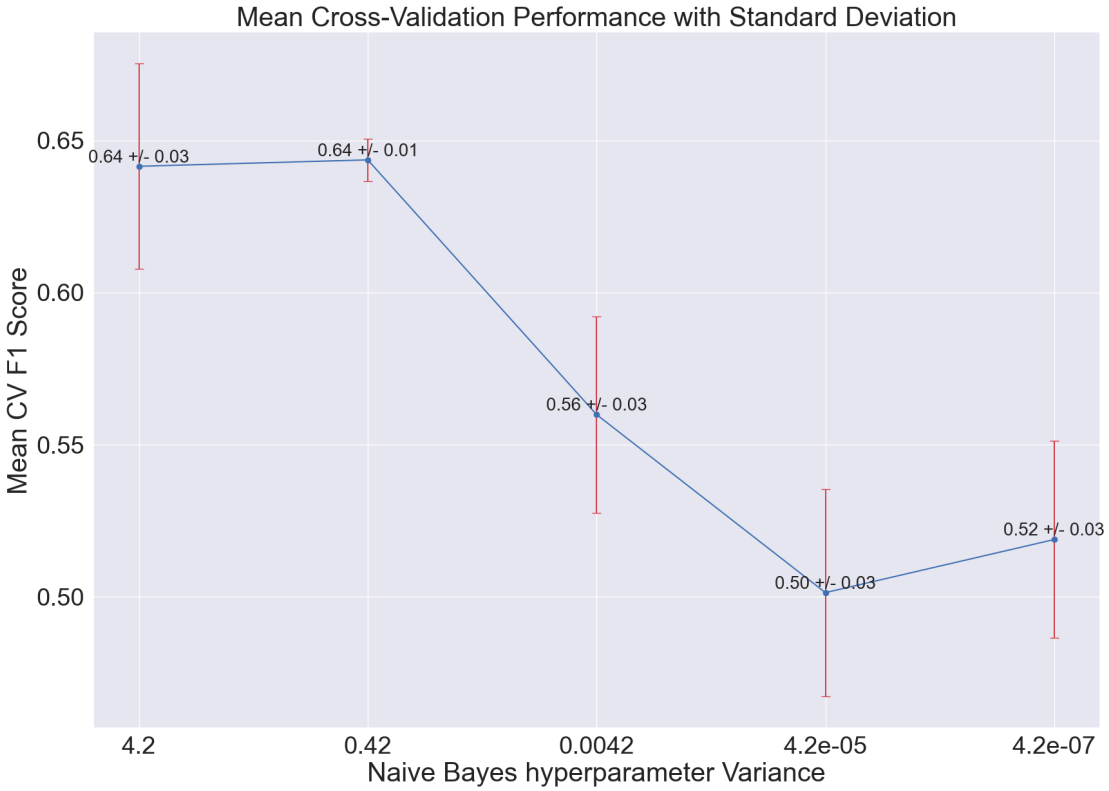
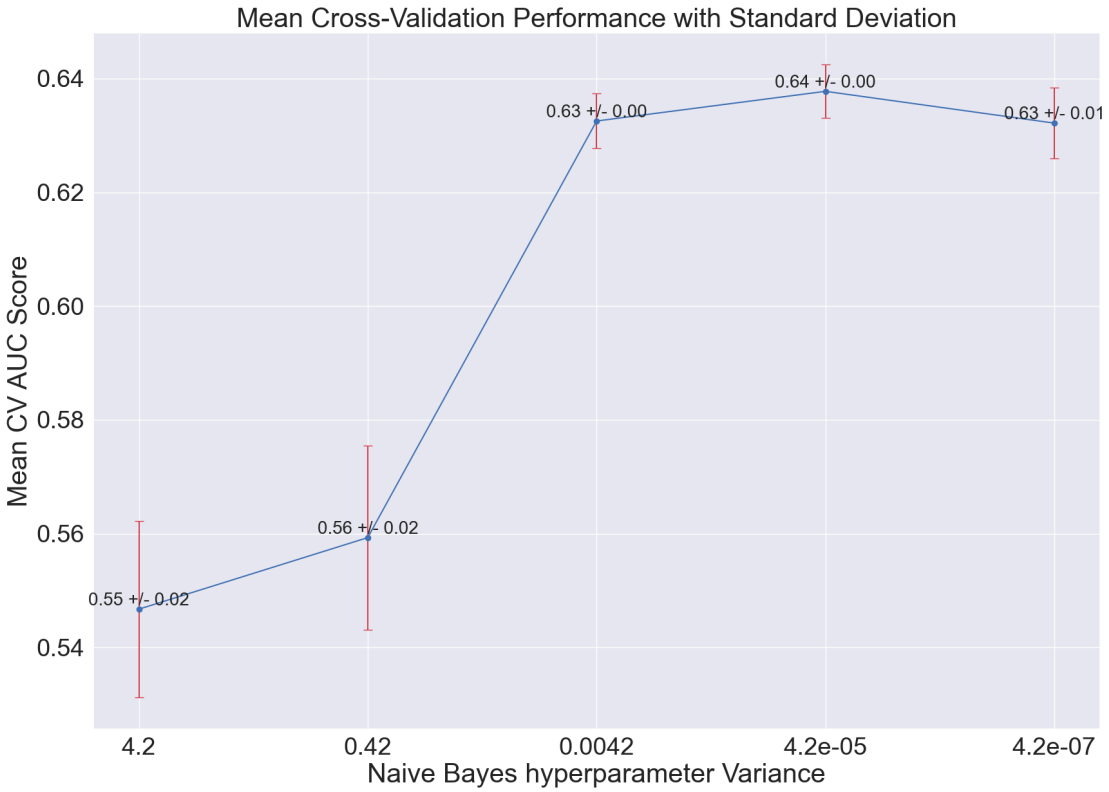


Figure 7: mean CV metrics with standard deviation (red lines) as a function for 5 hyperparameters {0.042, 0.42, 4.2, 42, 420} for the logistic regression. The regularizer controls the LogReg's coefficient values, and influences whether it overfits, and its tolerance to outliers. From top to bottom we report the mean CV F1 score, CV AUC score and log loss. We observe a classical inverted cup shape where there is some "optimal" hyperparameter 4.2 that is neither too low (encourages overfitting), nor too high (encouraging underfitting). Again, we consistently observe a very high standard deviation, similar to the linear SVM, which again indicates unstable predictions which could result from the linear model being unable to capture the complicated dependencies amongst input features for predicting readmission rate. Despite this, both F1 and AUC hover around 0.6, which is a reasonably good performance, at least much better than random guessing the need for hospital readmission.


Mean Cross-Validation Performance with Standard Deviation
Mean CV F1 Score
0.65
0.60
0.55
0.50
0.64 +/- 0.03
0.64 +/- 0.01
0.56 +/- 0.03
0.50 +/- 0.03
0.52 +/- 0.03
4.2
0.42
0.0042
4.2e-05
4.2e-07
Naive Bayes hyperparameter Variance


Mean Cross-Validation Performance with Standard Deviation
Mean CV AUC Score
0.64
0.62
0.60
0.58
0.56
0.54
0.55 +/- 0.02
0.56 +/- 0.02
0.63 +/- 0.00
0.64 +/- 0.00
0.63 +/- 0.01
4.2
0.42
0.0042
4.2e-05
4.2e-07
Naive Bayes hyperparameter Variance

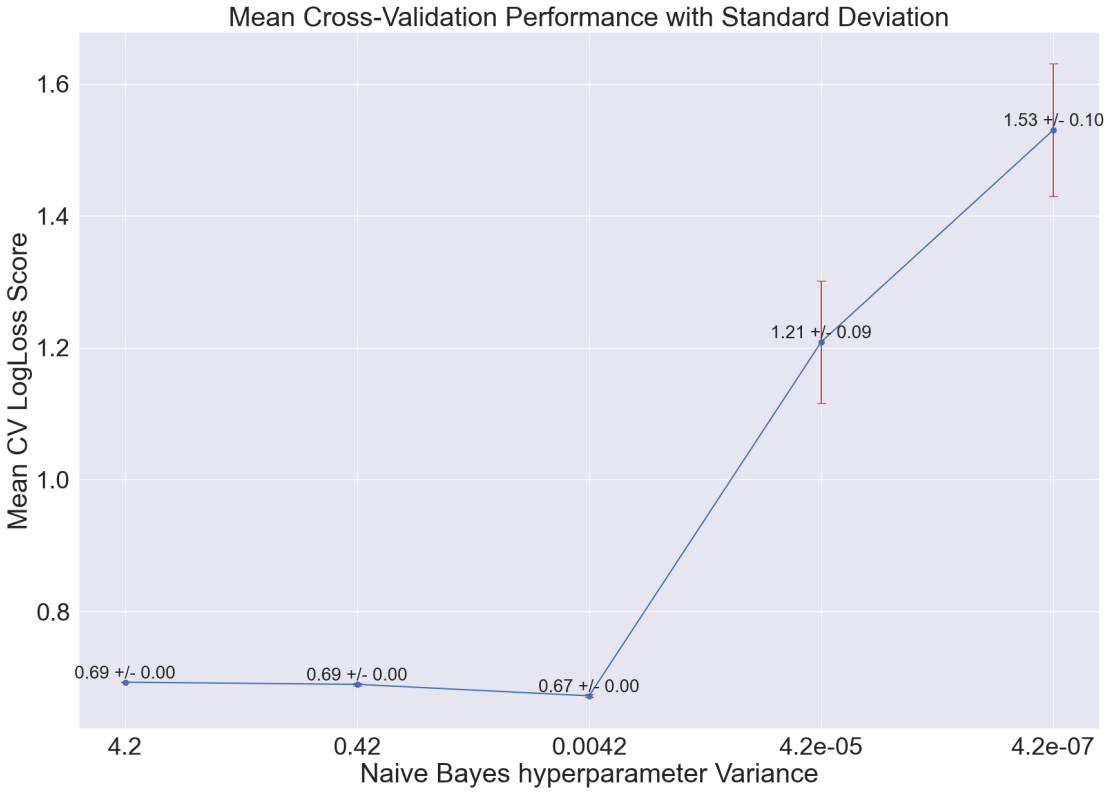
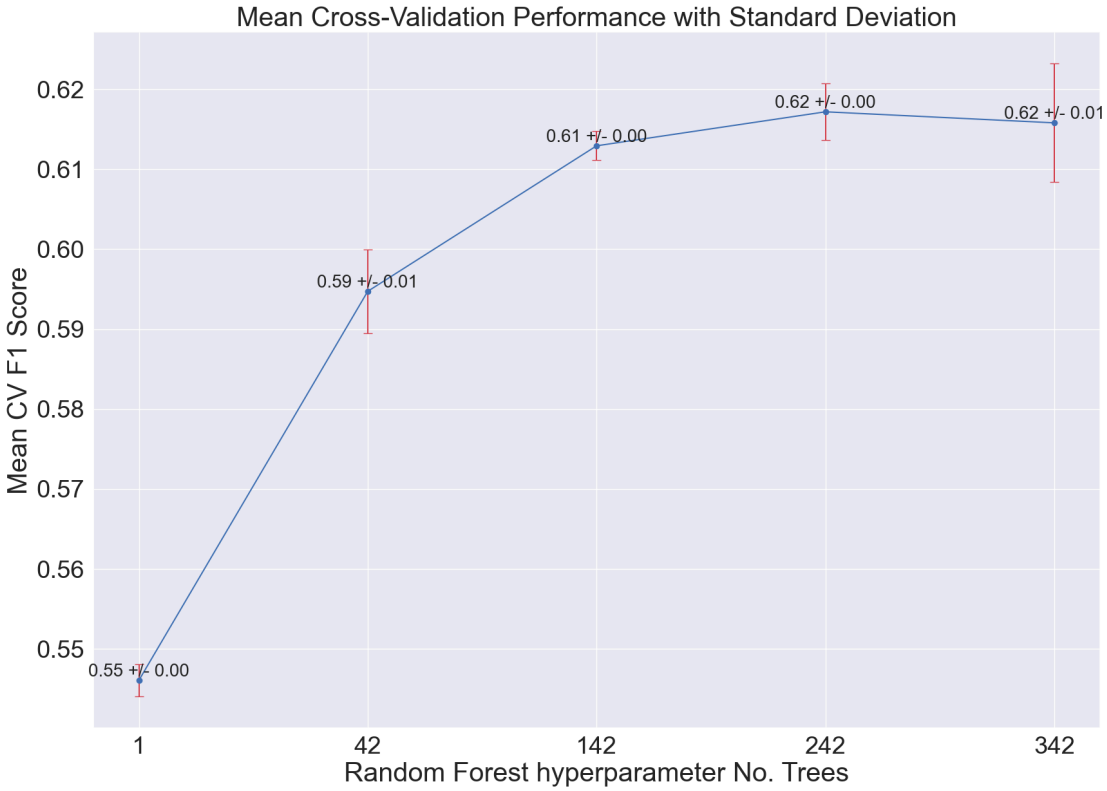
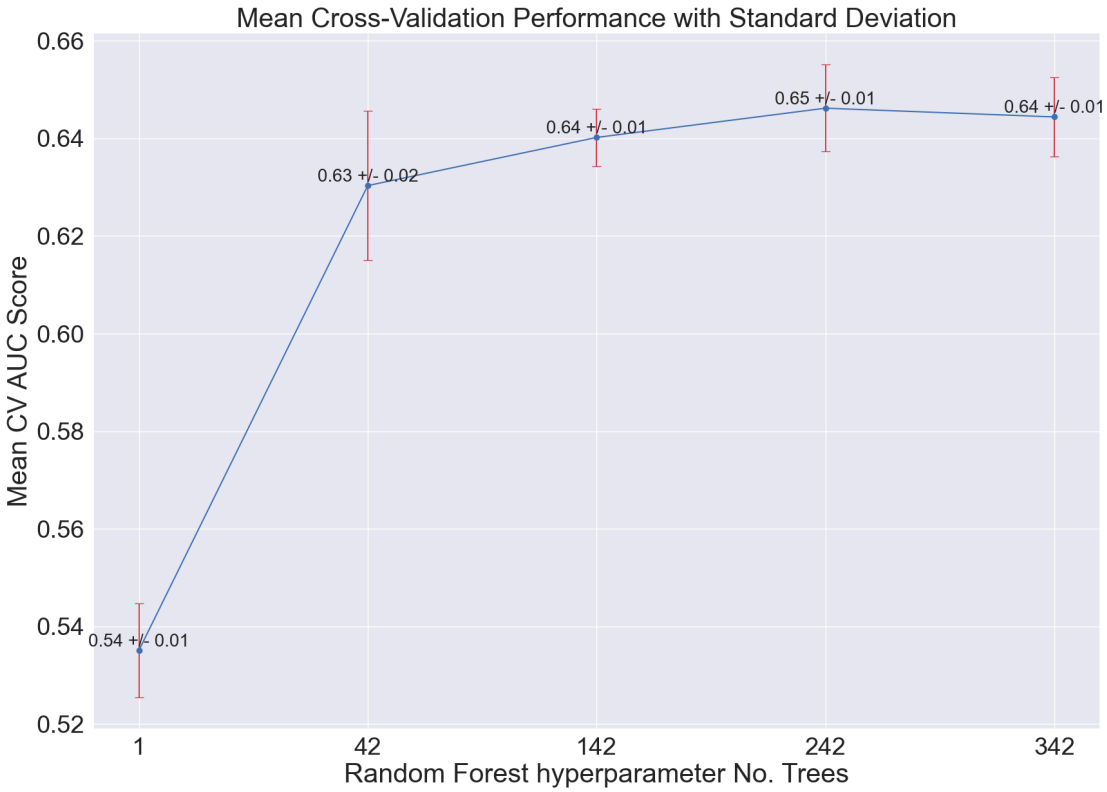


Figure 8: mean CV metrics with standard deviation (red lines) as a function for 5 hyperparameters {4.2, 0.42, 4.2e-3, 4.2e-5, 4.2e-7} controlling the Laplace smoothing of the variance for the Naïve Bayes classifier. A low smoothing value pushes the likelihood predictions toward a value of 0.5, while a high Laplace smoothing valueadds some noise to categories with extremely low appearance rate, addressing class imbalance to some extent. It's clear from the graph that smoothing values less than 0.42 destabilizes training (increase log loss), as well as yields low F1-score.


Mean Cross-Validation Performance with Standard Deviation
Mean CV F1 Score
0.62
0.61
0.60
0.59
0.58
0.57
0.56
0.55
0.55 +/- 0.00
0.59 +/- 0.01
0.61 +/- 0.00
0.62 +/- 0.00
0.62 +/- 0.01
1
42
142
242
342
Random Forest hyperparameter No. Trees


Mean Cross-Validation Performance with Standard Deviation
Mean CV AUC Score
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0.54 +/- 0.01
0.63 +/- 0.02
0.64 +/- 0.01
0.65 +/- 0.01
0.64 +/- 0.01
1
42
142
242
342
Random Forest hyperparameter No. Trees

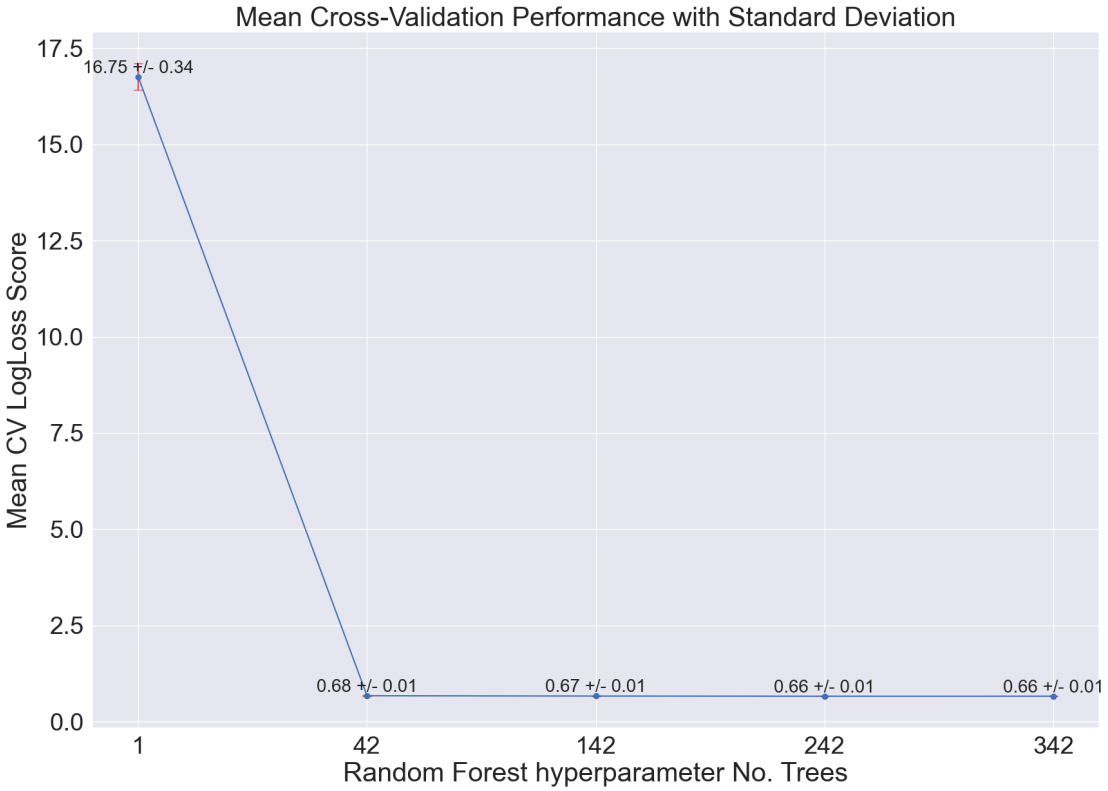


Figure 9: mean CV metrics with standard deviation (red lines) as a function for 5 hyperparameters {1, 42, 142, 242, 342} controlling the number of tree estimators in the random forest, whilst keeping other default hyperparameters such as tree width and depth the same. Compared to the other models, the RFC shows the least SD across mean CVs, indicating stable predictions. Like Figure 7, we observe a (less noticeable) inverted cup shape where there is an optimal number of trees sitting between 342 (which may encourage overfitting to raining) and 42 (which might be too low for capturing the complexities of the data). At 1 tree estimator, we retrieve a decision tree, which is the worst performing and most unstable according to log loss. The plateau indicate that performance has converged with a maximum amount of trees.

Model	Optimal hyperparameter as per F1 CV-score	F1 score on test set with optimal hyperparameter	AUC on test set with optimal hyperparameter	Log loss on test set with optimal hyperparameter
Linear SVM	C: 0.042	0.65	0.64	0.66
Logistic regression	C: 4.2	0.65	0.65	0.66
Naïve Bayes	Laplace smoothing: 0.42	0.66	0.54	0.69
Random forest classifier	No. of Trees: 242	0.64	0.63	0.67

Table 5: Test performance of different metrics for optimal hyperparameter for each model. The highest ranking, as per test F1 score is Naïve Bayes, but it also shows the lowest AUC, indicating that at different classification thresholds, the NB may classify poorly.

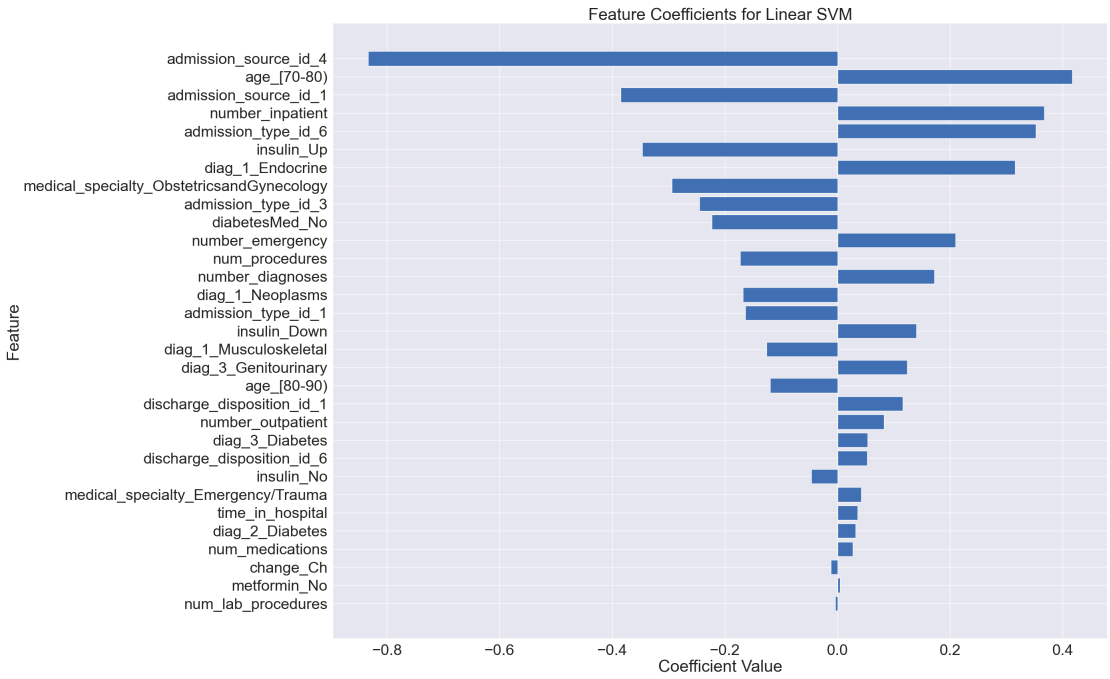


Figure 10: Linear SVM's coefficients with C: 0.042

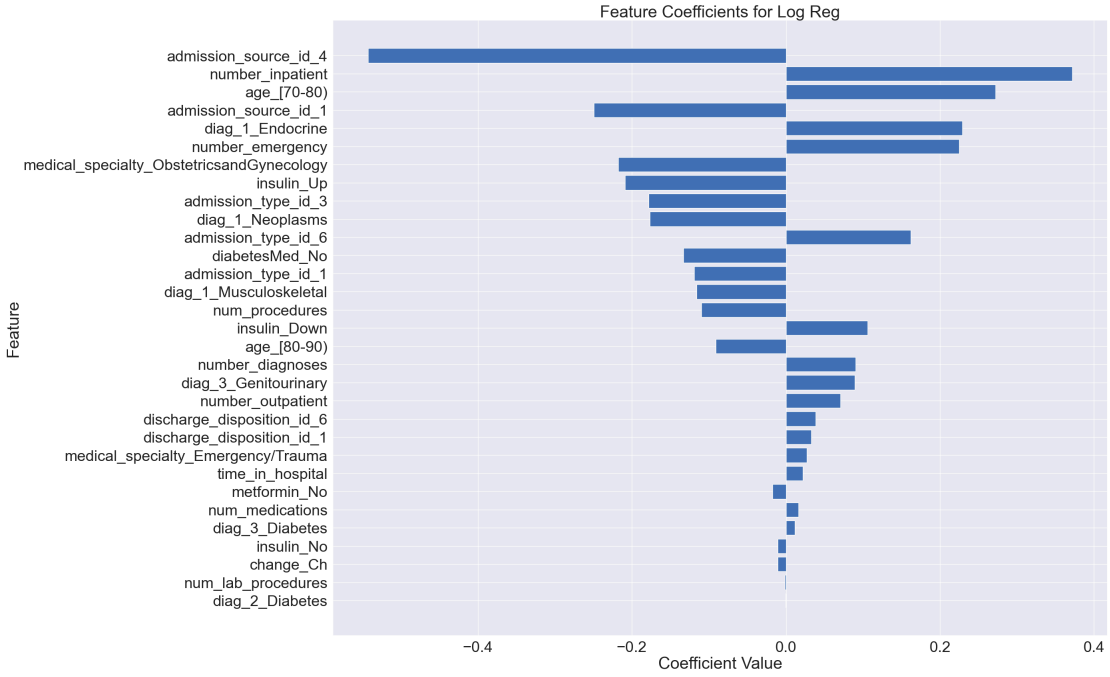


Figure 11: Model coefficients for logistic regression with C: 4.2. We notice some overlap both in terms of high ranking features (e.g. age, admission source 4) and low ranking features (e.g. number of lab procedures) most likely due to regularization gearing linear models to ignore variables like number of lab procedures which takes larger values compared to other features (as per Figure 4)

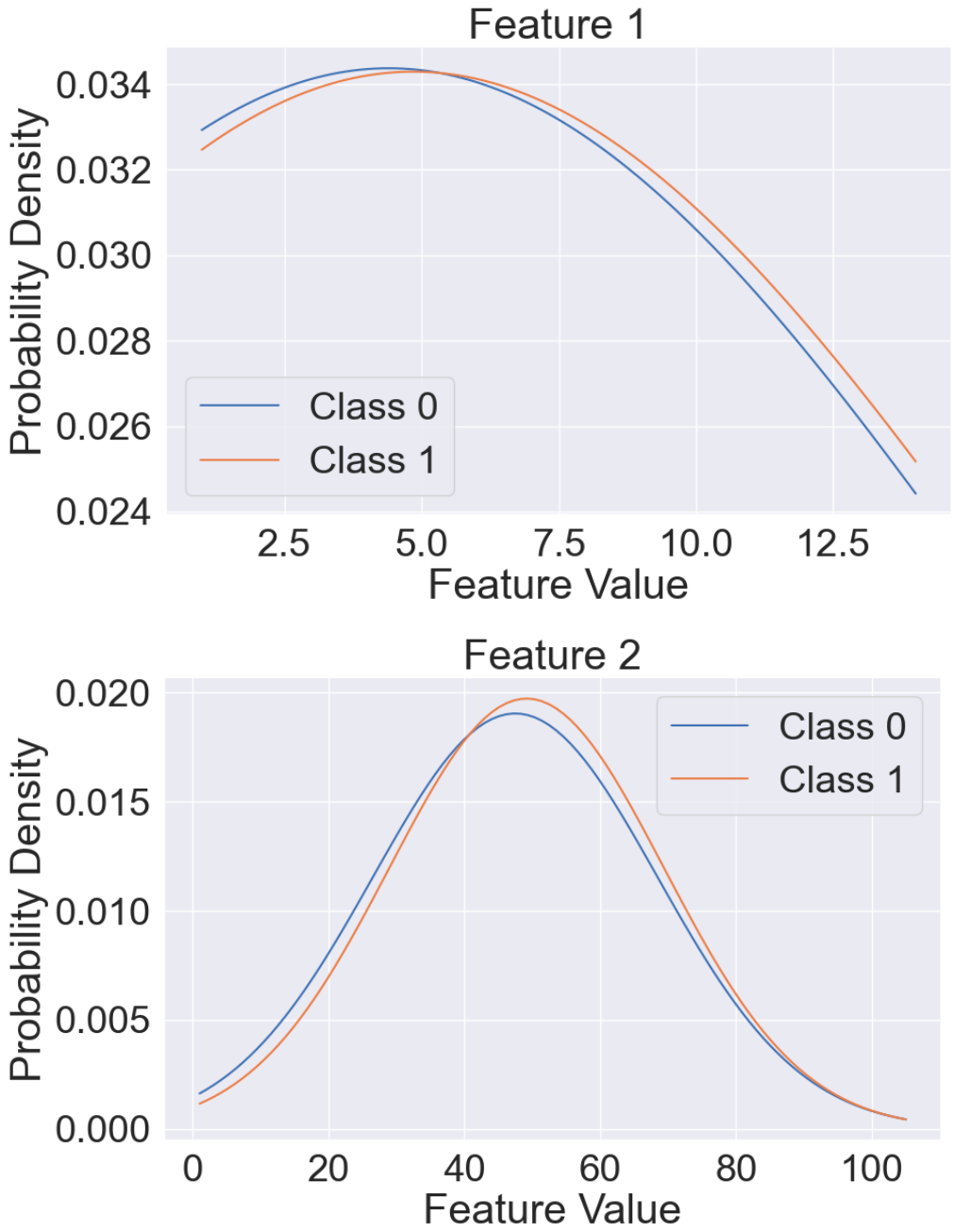


Figure 12: Gaussian distribution of the first 2 features estimated by NB corresponding to time in the hospital and number of lab procedures. We only plot 2 since 31 images would be too much.

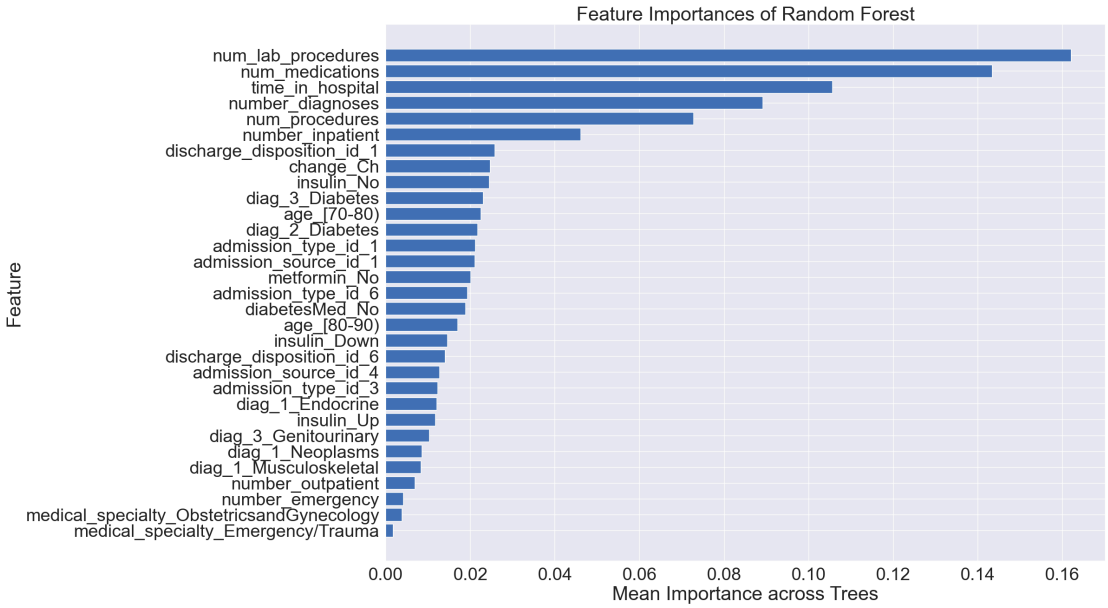


Figure 13: Mean feature importance of 242 trees in RF. We observe the most important feature belonging to number of lab procedures, and this could be due to the tree estimators being sensitive to the scale of this feature.

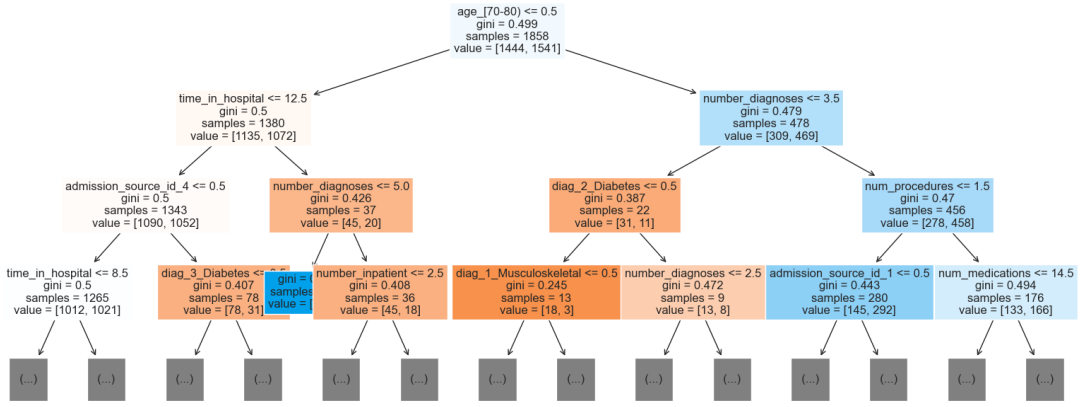


Figure 14: Decision path of tree 42 with a max depth of 3.

Model	**Optimal hyperparameter as per inner F1 CV-score**	**Mean F1 score with standard deviation on outer fold test set with optimal hyperparameter**
Linear SVM	C: 4.2	0.64 +/- 0.009
	C: 0.42	0.649 +/- 0.0102
Logistic regression	C: 4.2	0.641 +/− 0.007
	C: 0.042	0.647 +/− 0.0063
Naïve Bayes	Laplace smoothing: 4.2	0.681 +/− 0.002
	Laplace smoothing: 4.2e-5	0.604 +/− 0.002
	Laplace smoothing: 0.0042	0.6189 +/− 0.034
Random forest classifier	No. of Trees: 242	0.624 +/− 0.016
	No. of Trees: 342	0.630 +/− 0.014

Table 6: Mean test F1 score computed over outer-fold's test set with models trained with their optimal hyperparameter according to mean inner CV F1 scores.