BAZINE SAMER G:01

***INTRODUCTION***

The recent Covid-19 pandemic has underscored the critical importance of efficient healthcare management, particularly in

optimizing hospital resource allocation to enhance patient outcomes and operational effectiveness. One key parameter in this

domain is the patient Length of Stay (LOS), which represents the duration a patient remains admitted in a hospital. Accurate

prediction of LOS at the time of admission enables hospitals to identify patients at risk of prolonged stays, allowing for

tailored treatment plans to minimize LOS, reduce the risk of infections, and streamline logistics such as bed and room

allocation. This project, undertaken by a non-profit organization dedicated to improving hospital operations, aims to develop a

robust predictive model to forecast LOS for each patient on a case-by-case basis. The dataset provided, "LOS.csv," contains

various patient and hospital-related features, with LOS categorized into 11 classes ranging from 0-10 days to over 100 days. The

primary objective is to leverage data science techniques to build and evaluate multiple predictive models, ensuring accurate LOS

classification to support optimal hospital functioning and resource management. Through comprehensive data exploration, model

development, and performance evaluation, this project seeks to deliver actionable insights for healthcare management while

adhering to the principles of precision and efficiency.

@@ Q1 read the data with simple discribe

know, we need to make some variable as character to make a good analysis

####################

#######

####################

@@ Q2

####################

#######

####################

@@ Q2: some descriptive analysis

Summary Statistics for Numerical Variables: X case\_id Available.Extra.Rooms.in.Hospital Bed.Grade patientid City\_Code\_Patient

Min. : 1 Min. : 20 Min. : 0.000 Min. :1.000 Min. : 18 Min. : 1.000

1st Qu.: 7961 1st Qu.: 79555 1st Qu.: 2.000 1st Qu.:2.000 1st Qu.: 32906 1st Qu.: 4.000

Median :15922 Median :159352 Median : 3.000 Median :3.000 Median : 65269 Median : 8.000

Mean :15922 Mean :159224 Mean : 3.189 Mean :2.621 Mean : 65708 Mean : 7.242

3rd Qu.:23882 3rd Qu.:238159 3rd Qu.: 4.000 3rd Qu.:3.000 3rd Qu.: 98793 3rd Qu.: 8.000

Max. :31842 Max. :318431 Max. :14.000 Max. :4.000 Max. :131624 Max. :37.000

NA's :11 NA's :413

Visitors.with.Patient Admission\_Deposit

Min. : 0.000 Min. : 1847

1st Qu.: 2.000 1st Qu.: 4179

Median : 3.000 Median : 4733

Mean : 3.288 Mean : 4882

3rd Qu.: 4.000 3rd Qu.: 5417

Max. :32.000 Max. :11008

Summary Statistics for Categorical Variables: Hospital\_code Hospital\_type\_code City\_Code\_Hospital Hospital\_region\_code Department Ward\_Type

Length:31842 Length:31842 Length:31842 Length:31842 Length:31842 Length:31842

Class :character Class :character Class :character Class :character Class :character Class :character

Mode :character Mode :character Mode :character Mode :character Mode :character Mode :character

Ward\_Facility\_Code Type.of.Admission Severity.of.Illness Age Stay

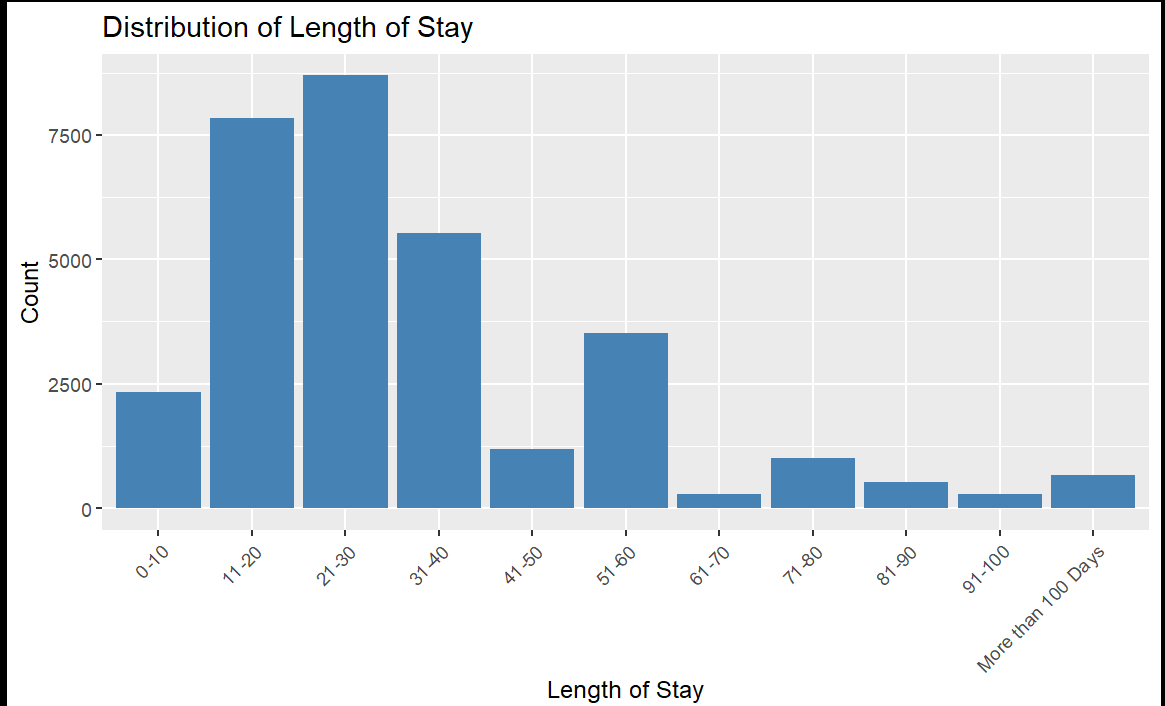
Length:31842 Length:31842 Length:31842 Length:31842 Length:31842

Class :character Class :character Class :character Class :character Class :character

Mode :character Mode :character Mode :character Mode :character Mode :character

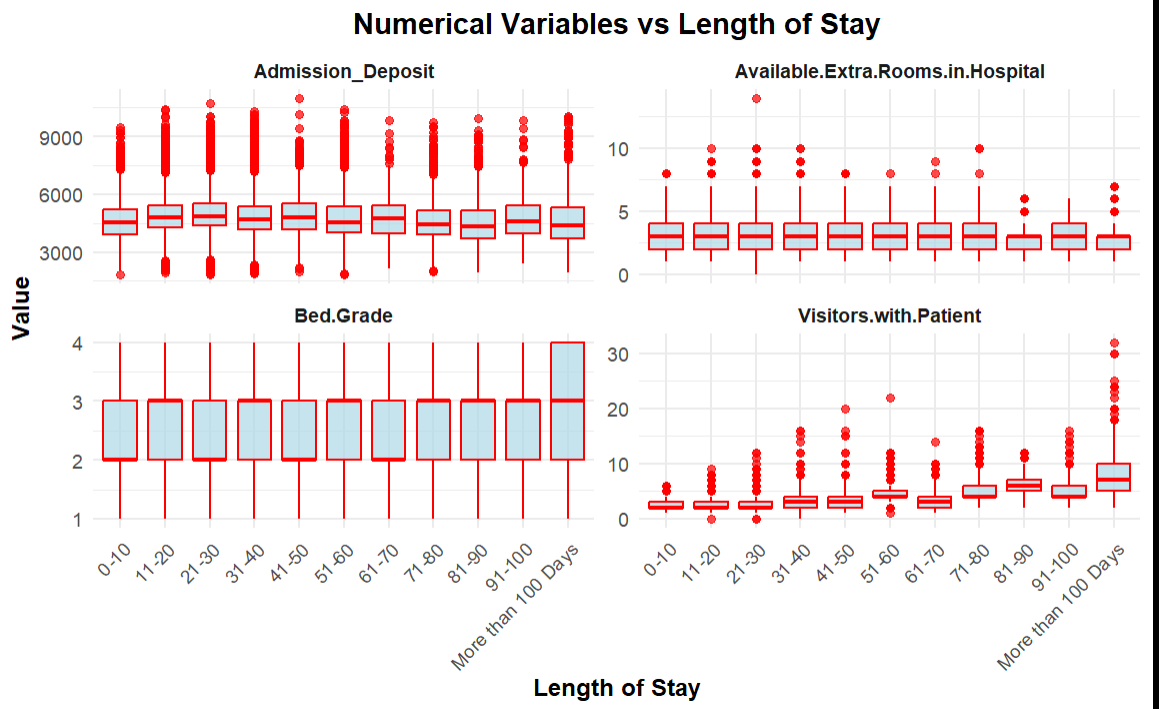
Distribution of target (stay)

|  |
| --- |
|  |
| **Var1**  <fctr> | **Freq**  <dbl> |  | |  |  |
| 0-10 | 0.073142391 |  | |  |  |
| 11-20 | 0.245870234 |  | |  |  |
| 21-30 | 0.273098423 |  | |  |  |
| 31-40 | 0.173230325 |  | |  |  |
| 41-50 | 0.037529050 |  | |  |  |
| 51-60 | 0.110294579 |  | |  |  |
| 61-70 | 0.008981848 |  | |  |  |
| 71-80 | 0.031624898 |  | |  |  |
| 81-90 | 0.016393443 |  | |  |  |
| 91-100 | 0.008730607 |  | |  |  |
| More than 100 Days | | | 0.021104202 | | |



The distribution of hospital stays is positively skewed, indicating that shorter stays are more common. The most frequent stay duration falls within the 21–30 day range, accounting for approximately 8000 cases. The majority of stays are concentrated between 0 and 60 days, with a noticeable decline after 50 days. Stays exceeding 100 days are rare.

and we that our data is umbalanced so we must handl it



Admission deposit

The median admission deposit remains stable (4500–5000) across all stay durations, with most values between 3000 and 6000. High outliers >9000 are more common in shorter stays, suggesting occasional costly admissions. Overall, admission deposit shows little variation with stay length, indicating it may not be a strong direct predictor.

Available Extra Rooms

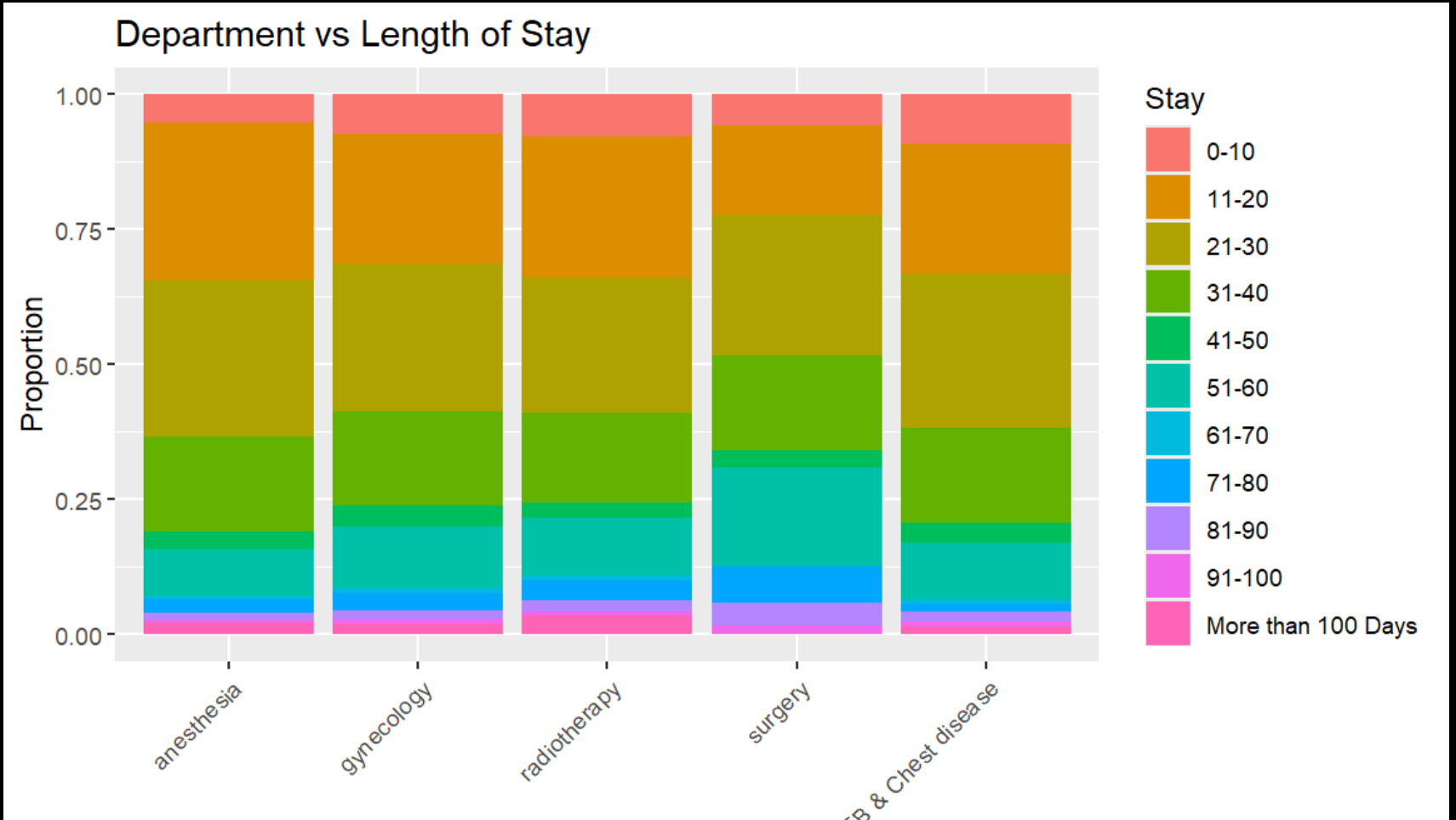
The median number of extra rooms remains steady (3–5) across all stay durations, with some mild variability. Outliers (12 rooms) appear in both short and long stays but in the real may be not an outlier because may be 12 room availabels. Overall, no clear relationship with stay length is observed, though room availability may still indirectly impact hospital operations or discharge timing.

Bed grade

remains consistent across all stay durations, with a median of 2–3 and minimal variation. Outliers (grades 1 and 4) appear evenly across bins, showing no strong pattern. but in real its not a outliers it is just a good gread (4)

Visitors with Patient

Most stays have a low median visitor count (2–3), but longer stays (especially 91–100 days) show higher medians and wider spread, with outliers reaching up to 30. Visitor count increases slightly with stay length, possibly reflecting more serious conditions.



Department :

Shorter stays (11–30 days) dominate across all departments, confirming earlier findings.

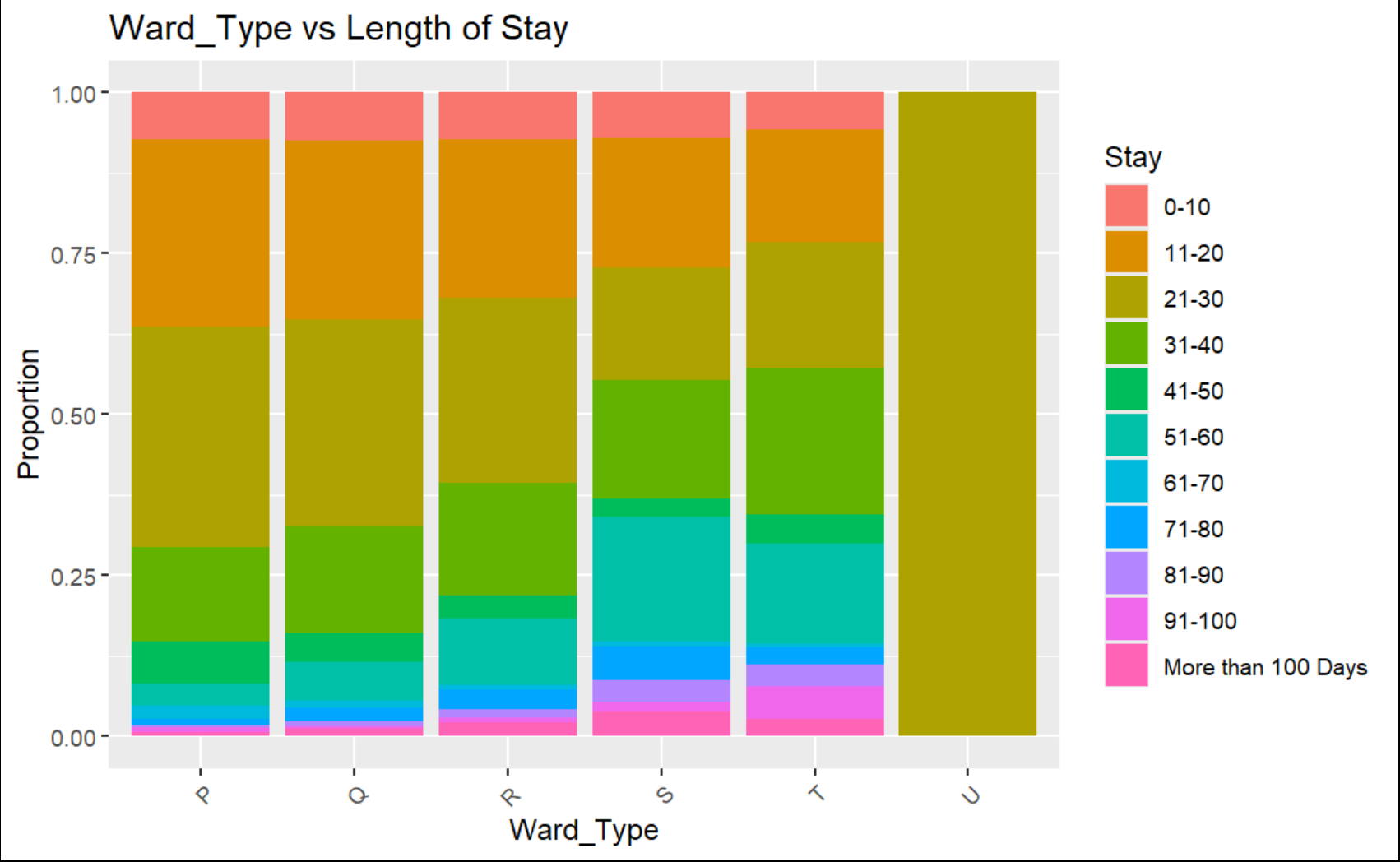
Very short stays (0–10 days) are the least common across the board.

The 21–30 day bin consistently shows the highest proportion of stays.

Most departments are dominated by short- to medium-length stays.

Radiotherapy and TB & Chest Disease stand out for having more long-duration cases, suggesting their patients may need more extended care.

This distribution highlights potential resource and care planning differences across departments



Ward Type :

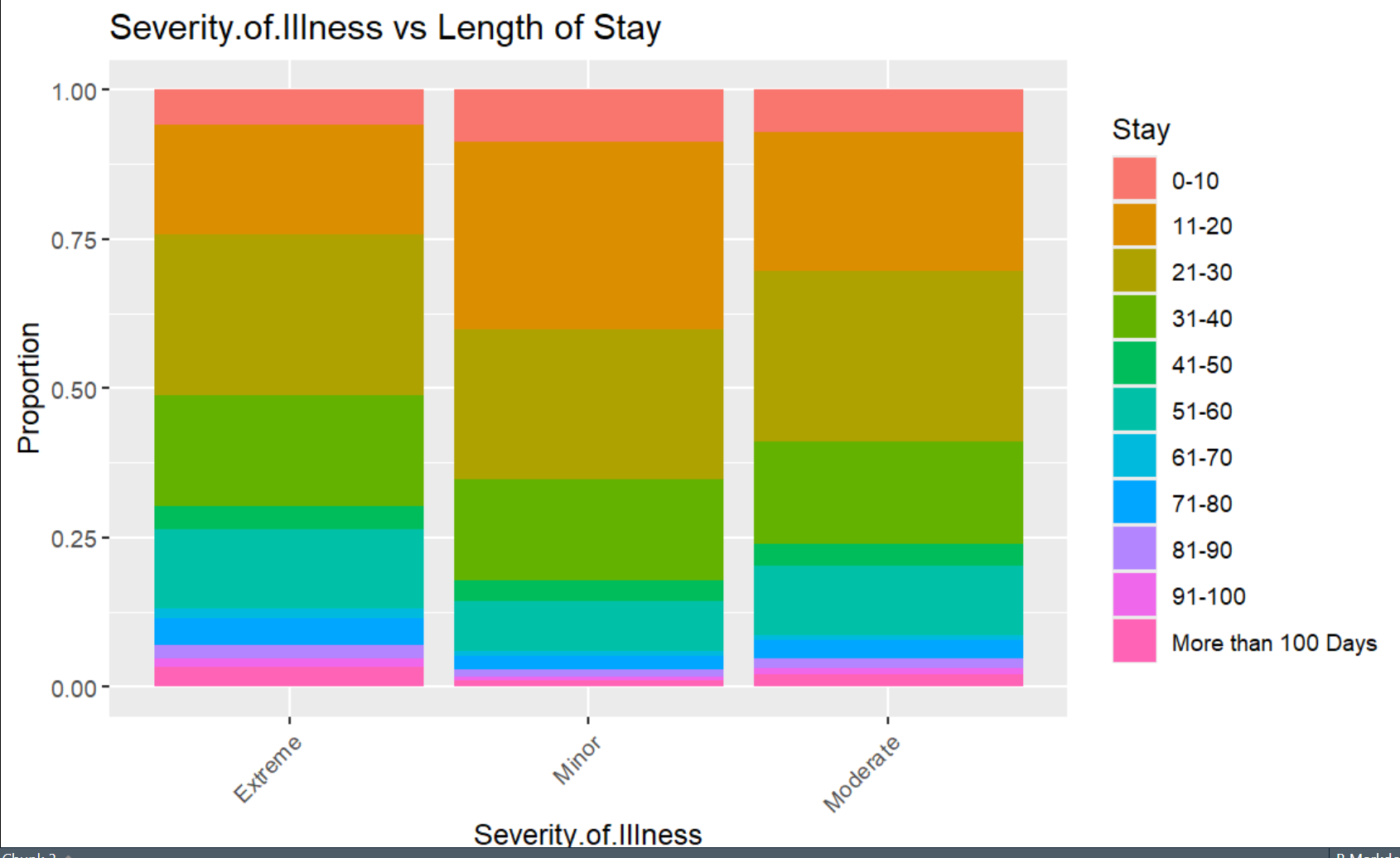
The chart compares stay duration distributions across six ward types (P, Q, R, S, T, U)

Most wards show a high proportion of stays between 11–30 days, reflecting the dataset peak in that range

Wards P, Q, R: Associated with shorter stays, likely routine or procedural care.

Wards S, T: Show mixed durations possibly for more complex cases.

Ward U: Heavily skewed toward extended stays, indicating specialized or long-term patient care.



*Severity of Illness:*

11–30 day stays dominate across all severity levels, matching the overall dataset trend.

Short stays (0–10 days) are relatively rare for all groups.

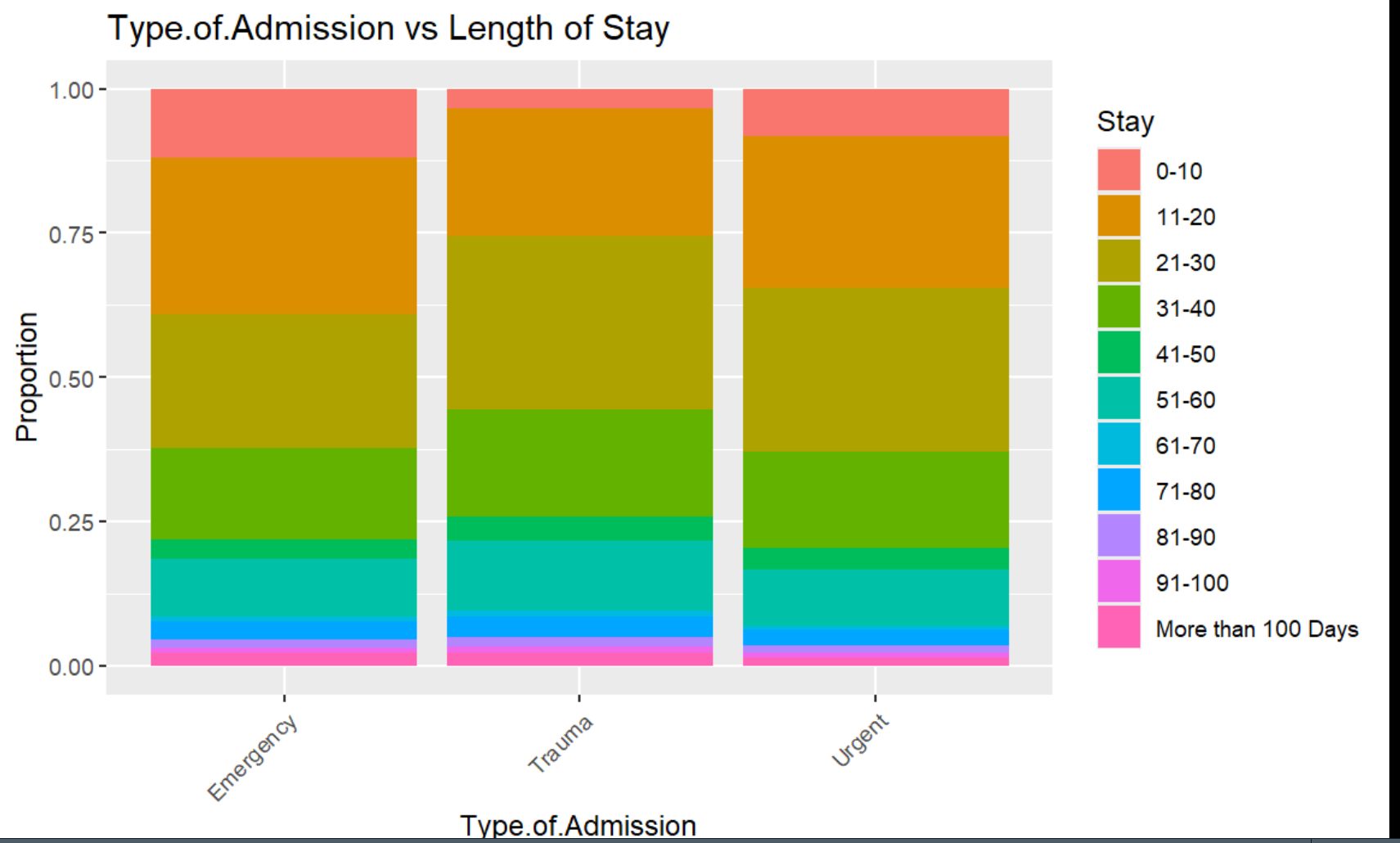
There’s a clear positive correlation between illness severity and hospital stay duration:

Extreme cases → tend to stay longer.

Moderate cases → middle ground.

Minor cases → recover faster, shorter stays.

This makes Severity\_of\_Illness a strong candidate for predicting length of stay.



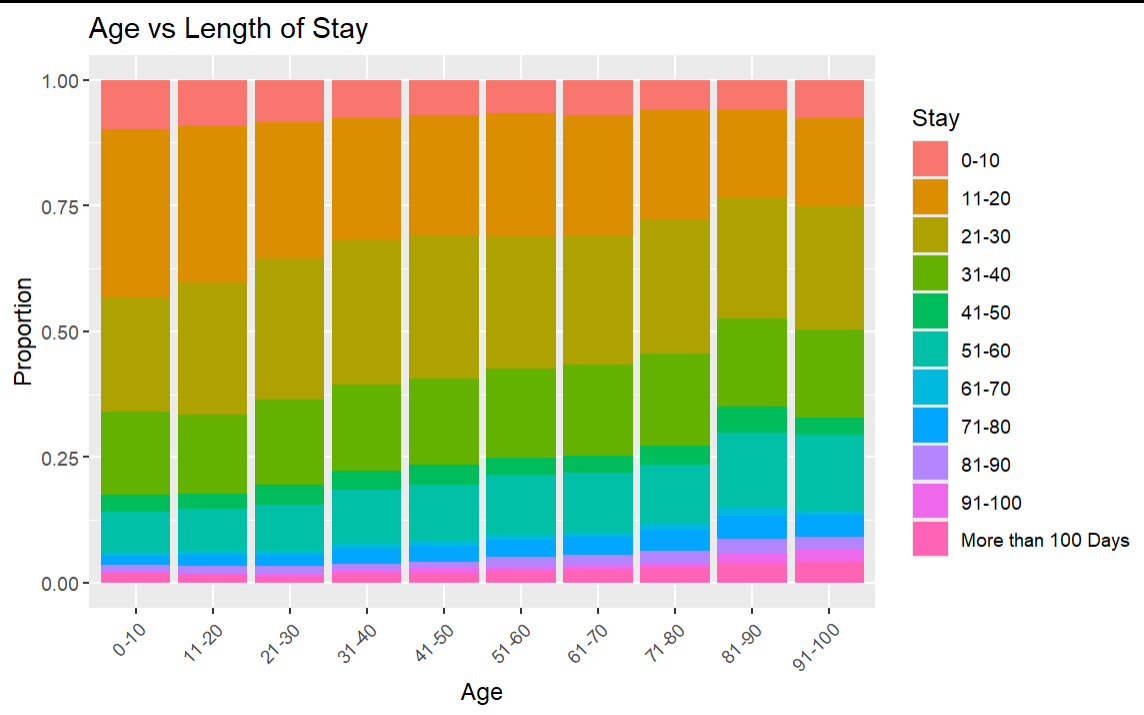
Type of Admission :

11–30 day stays dominate across all admission types, aligning with the overall length-of-stay distribution.

Short stays (0–10 days) are relatively rare in all categories.

While Emergency and Trauma admissions show a wider spread into longer stays, Urgent cases tend to resolve more quickly.

Overall, Type\_of\_Admission has moderate predictive value for length of stay—less impactful than severity or ward type but still relevant.



Age :

Across all age groups, 11–30 day stays are the most common.

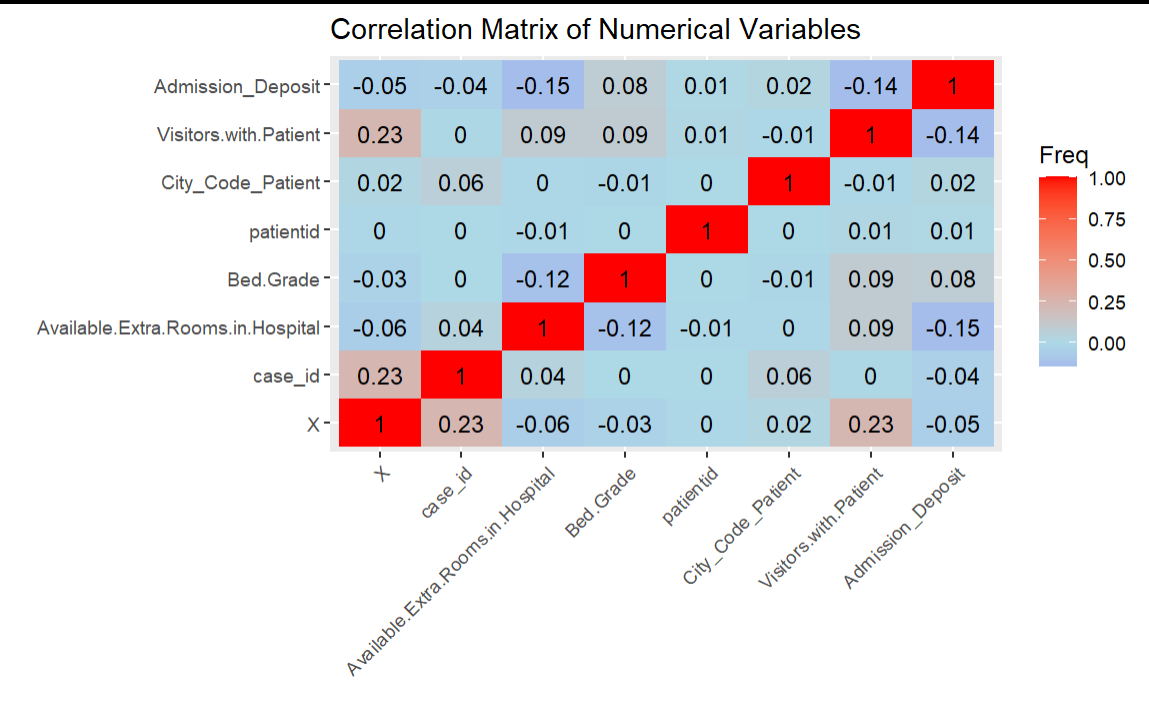
Short stays (0–10 days) are relatively uncommon throughout.

Age correlates modestly with longer hospital stays—the older the patient, the more likely extended stays become.

Age is a meaningful predictor of length of stay, especially when differentiating between younger and elderly patients.

Older individuals (especially 70+) are significantly more likely to experience prolonged hospitalizations.

*correlations*



No strong correlations are present

All correlations are weak (between -0.15 and 0.23), suggesting low multicollinearity.

This independence allows all variables to be safely included in modeling without redundancy concerns.

However, because the relationships are so weak, these features may only modestly influence predictions or have limited interaction effects on the target stay

***missing\_values :***

"Missing values per column:"

Bed.Grade City\_Code\_Patient

11 413

so we must handl it

Unique Values in Identifier ColumnsUnique case\_id: 31842Unique patientid: 26289Unique Hospital\_code: 32

The dataset contains 31,842 unique case\_id values, indicating each row represents a distinct case

There are 26,289 unique patientid values, suggesting some patients have multiple cases

The 32 unique Hospital cod values indicate the data spans 32 distinct hospitals

and let's make the target as a factor

and see some summary

Hospital\_code Hospital\_type\_code City\_Code\_Hospital Hospital\_region\_code Available.Extra.Rooms.in.Hospital

Length:31842 Length:31842 Length:31842 Length:31842 Min. : 0.000

Class :character Class :character Class :character Class :character 1st Qu.: 2.000

Mode :character Mode :character Mode :character Mode :character Median : 3.000

Mean : 3.189

3rd Qu.: 4.000

Max. :14.000

Ward\_Type Bed.Grade patientid City\_Code\_Patient Type.of.Admission Severity.of.Illness

Length:31842 Min. :1.000 Min. : 18 Min. : 1.000 Length:31842 Length:31842

Class :character 1st Qu.:2.000 1st Qu.: 32906 1st Qu.: 4.000 Class :character Class :character

Mode :character Median :3.000 Median : 65269 Median : 8.000 Mode :character Mode :character

Mean :2.621 Mean : 65708 Mean : 7.242

3rd Qu.:3.000 3rd Qu.: 98793 3rd Qu.: 8.000

Max. :4.000 Max. :131624 Max. :37.000

NA's :11 NA's :413

Visitors.with.Patient Age Admission\_Deposit Stay

Min. : 0.000 Length:31842 Min. : 1847 21-30 :8696

1st Qu.: 2.000 Class :character 1st Qu.: 4179 11-20 :7829

Median : 3.000 Mode :character Median : 4733 31-40 :5516

Mean : 3.288 Mean : 4882 51-60 :3512

3rd Qu.: 4.000 3rd Qu.: 5417 0-10 :2329

Max. :32.000 Max. :11008 41-50 :1195

(Other):2765

@@Q3 AND Q4

Splits data into 2/3 training and 1/3 testing sets, imputes missing Bed Grade and City Code\_Patient with medians, normalizes numeric predictors removes near-zero variance predictors, pools rare nominal categories (threshold = 0.2), creates dummy variables for nominal predictors, and applies SMOTE to oversample minority Stay classes (over\_ratio = 0.3, 150 neighbors). The prepped recipe produces a processed training dataset for modeling.

we dont use step\_corr because we see that No strong correlations are present in the hetmap

***justify the use of every step function:***

step\_impute\_median(Bed\_Grade) %>%

step\_impute\_mode(City\_Code\_Patient)

because i have a missing values in only this 2 variable

step\_normalize(all\_numeric\_predictors()):

there is a lot of dispertion so it should be used

make all numeric features into a comparable scale, improving convergence and interpretability.

step\_nzv(all\_predictors()):

to remove the feature whiche have near 0 of variance

step\_other(all\_nominal\_predictors(),threshold = 0.2):

Groups infrequent factor levels (with proportion < 0.2) into a new "other" category. to make the model learn more patterne

step\_dummy(all\_nominal\_predictors()):

to do the dummy envodiing , it is one of the best methode to do feature engenering and dummy encoding allows the model to treat each category as a separate binary feature.

step\_smote( Stay,over\_ratio = 0.3, neighbors = 150 ):

as we se in the distrubition of stay (target variable) there is imbalanced distribution so we need to use the smoote step to handl umbalanced

we dont use step\_corr because we see that No strong correlations are present in the hetmap

####################

#######

####################

***@@ Q5***

***THE MODELS CAN WE USED FOR THIS PROBLEM IS :***

RANDOM FOREST , KNN, SVM , XGBOOST,LIGHTGBM,MULTINOMIAL REGRESSION ,BAGGED DECISION TRESS, DECISION TREES

***:JUSTIFICATION OF PARAMETRES***

start with random forest

using 340 tres (high number ) to make the performance more good

mtry automatic ( so it using sqrt of number of variables)

####################

#######

####################

####################

#######

####################

***@Q7 :JUSTIFICATION OF PARAMETRES***

***MULTINOMIAL regression***

penalty parameter controls the strength of regularization in the model smaller penalty value means less regularization, allowing the model to fit the training data more closely by assigning larger coefficients to the predictors.

The mixture parameter controls the balance between L1 (lasso) and L2 (ridge) regularization:

mixture = 0: Pure ridge regression

mixture = 1: Pure lasso regression

mixture = 0.01: A value very close to 0, meaning the regularization is almost entirely L2 (ridge) with a tiny amount of L1 (lasso).

***lightgbm*** :

using 500 tres (high number ) to make the performance more good

tree depth 5 to avoid overfiting (dont use high number)

min n introduces randomness, which can reduce overfitting

learning rate to controle the speed of training (0.1) is good

mtry 1 means no feature subsampling is performed at the tree level

sample\_size of 1 means no subsampling is performed

min n = 3 is relatively small, allowing the model to create fine-grained splits

***DECISION TREES***

tree\_depth controls the maximum depth of the tree. A deeper tree can capture more complex patterns by allowing more splits

min n 20 is a moderate choice, providing a balance between allowing the tree to grow and preventing overfitting

cost\_complexity of 0.0004 is very small, meaning the tree is pruned minimally, allowing it to grow large and capture detailed patterns in the data

***xgboost model***

using 600 tres (high number ) to make the performance more good

tree depth 6 to avoid overfiting (dont use high number)

min n introduces randomness, which can reduce overfitting

learning rate to controle the speed of training (0.1) is good

mtry 1 means no feature subsampling is performed at the tree level

sample\_size of 0.9 means no subsampling is performed

***KNN***

neighbors 120 is relatively large, meaning the model will consider a large number of neighbors when making predictions.

dist\_power 1 determines the power parameter for the Minkowski distance metric used to calculate

***bag trees\_model***

cost\_complexity 0.0001 allows for relatively complex trees while still applying light pruning

tree\_depth 10 strikes a balance between capturing complex relationships and preventing overfitting

min n ensures that splits occur only when there are enough data points to make reliable decisions

***note***

this is a second recipe used (just change the values of over ratio and neighboors)

```{r}

df\_rec1<-df\_train%>%recipe(Stay~. )%>%

step\_impute\_median(Bed.Grade) %>%

step\_impute\_median(City\_Code\_Patient) %>%

step\_normalize(all\_numeric\_predictors())%>%

step\_nzv(all\_predictors())%>%

step\_other(all\_nominal\_predictors(),threshold = 0.2)%>%

step\_dummy(all\_nominal\_predictors()) %>%

step\_smote( Stay,over\_ratio = 1, neighbors = 10 )

```

***SVM***

SVM model with a rbf kernel

cost 1 is relatively small encouraging a wider margin and a simpler decision boundary.

rbf\_sigma = 0.2 2 is a moderate choice

####################

#######

####################

@@ q8

and let,s see the results

THE METRIC USED F1 scores

The F1 score treats all classes equally, balancing precision and recall to evaluate model performance, making it valuable for imbalanced datasets where rare classes are critical. Unlike accuracy, which can be skewed by frequent classes, or ROC-AUC, which focuses on overall ranking, the F1 score emphasizes correct predictions for each class individually. It is calculated as the harmonic mean of precision and recall, providing a single metric that penalizes models ignoring minority classes, ensuring robust performance across all classes, especially when rare classes are prioritized

random model

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3903335 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.7457312 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.3676290 | Preprocessor1\_Model1 |  |

|  |
| --- |
|  |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.2328647 |  |  |

Log model

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3550970 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.7352196 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.3867929 | Preprocessor1\_Model1 |  |

|  |
| --- |
|  |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.2076006 |  |  |

Light gbm

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3914641 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.7459819 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.3708355 | Preprocessor1\_Model1 |  |

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| --- | --- | --- | --- | --- |
| f\_meas | macro | 0.2306826 |  |  |

Decission trees

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3931600 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.7416124 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.3710346 | Preprocessor1\_Model1 |  |

|  |
| --- |
|  |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.3214393 |  |  |

Xgboost model

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3983418 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.7517720 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.3676158 | Preprocessor1\_Model1 |  |

|  |
| --- |
|  |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.2299794 |  |  |

Knn model

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3418127 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.6798870 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.3865968 | Preprocessor1\_Model1 |  |

3 rows

Svm model

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3094969 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.6306890 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.4085200 | Preprocessor1\_Model1 |  |

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| --- | --- | --- | --- | --- |
| f\_meas | macro | 0.1355037 |  |  |

1 row

| BAGGED DECISION TRESS  **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> | **.config**  <chr> |  |
| --- | --- | --- | --- | --- |
| accuracy | multiclass | 0.3948559 | Preprocessor1\_Model1 |  |
| roc\_auc | hand\_till | 0.7562032 | Preprocessor1\_Model1 |  |
| brier\_class | multiclass | 0.3674956 | Preprocessor1\_Model1 |  |

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| --- | --- | --- | --- | --- |
| f\_meas | macro | 0.2401227 |  |  |

####################

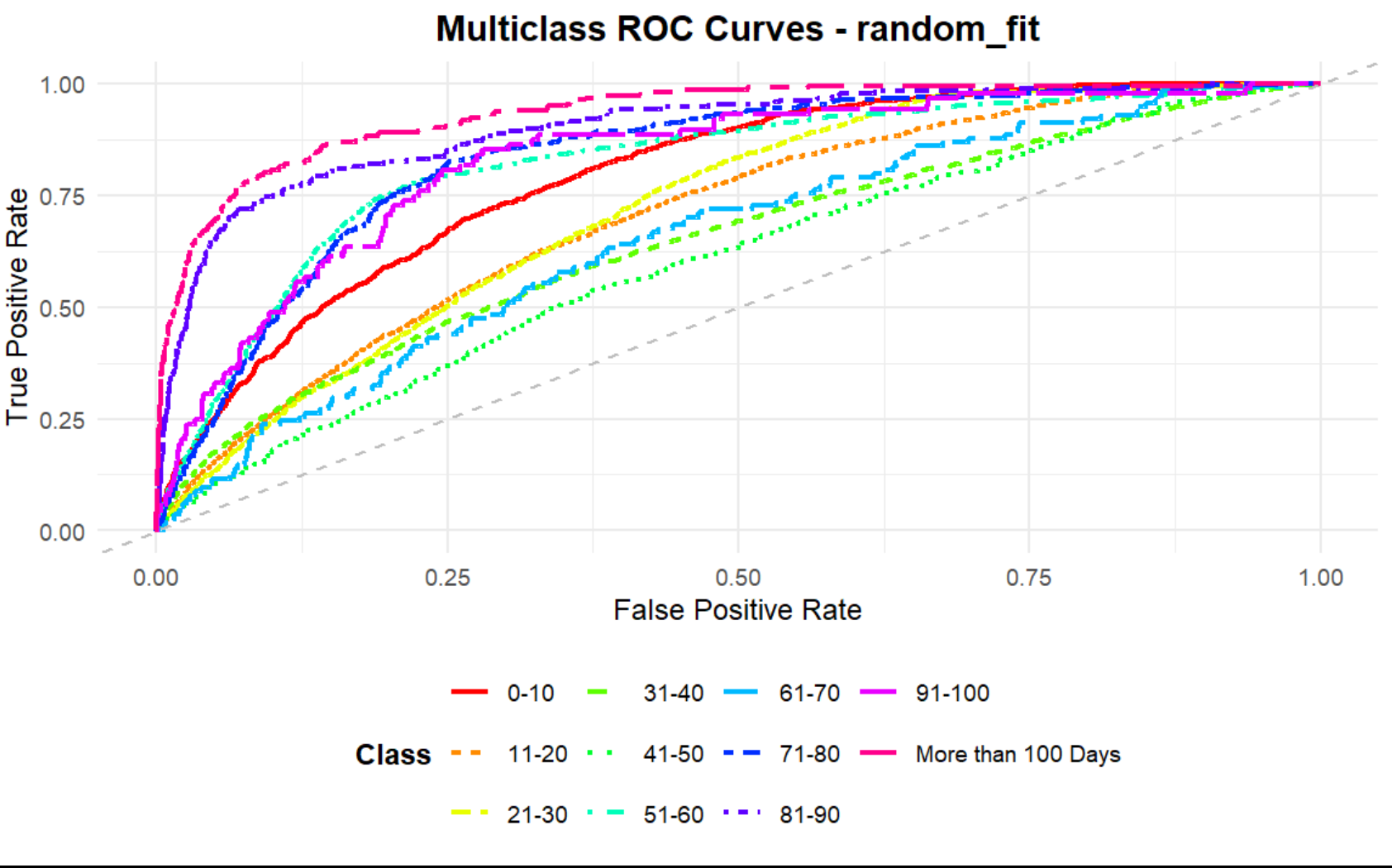
#######

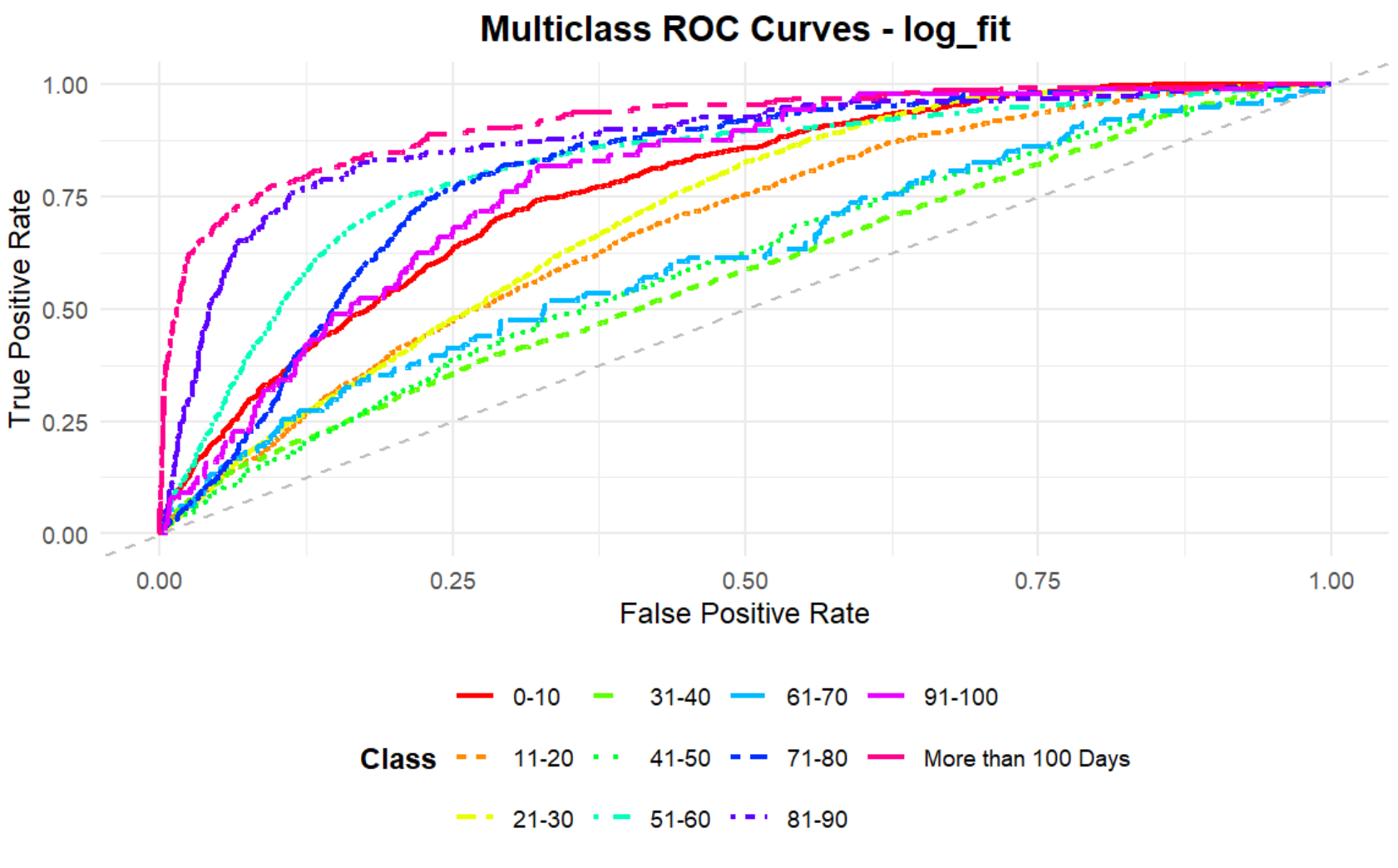
####################

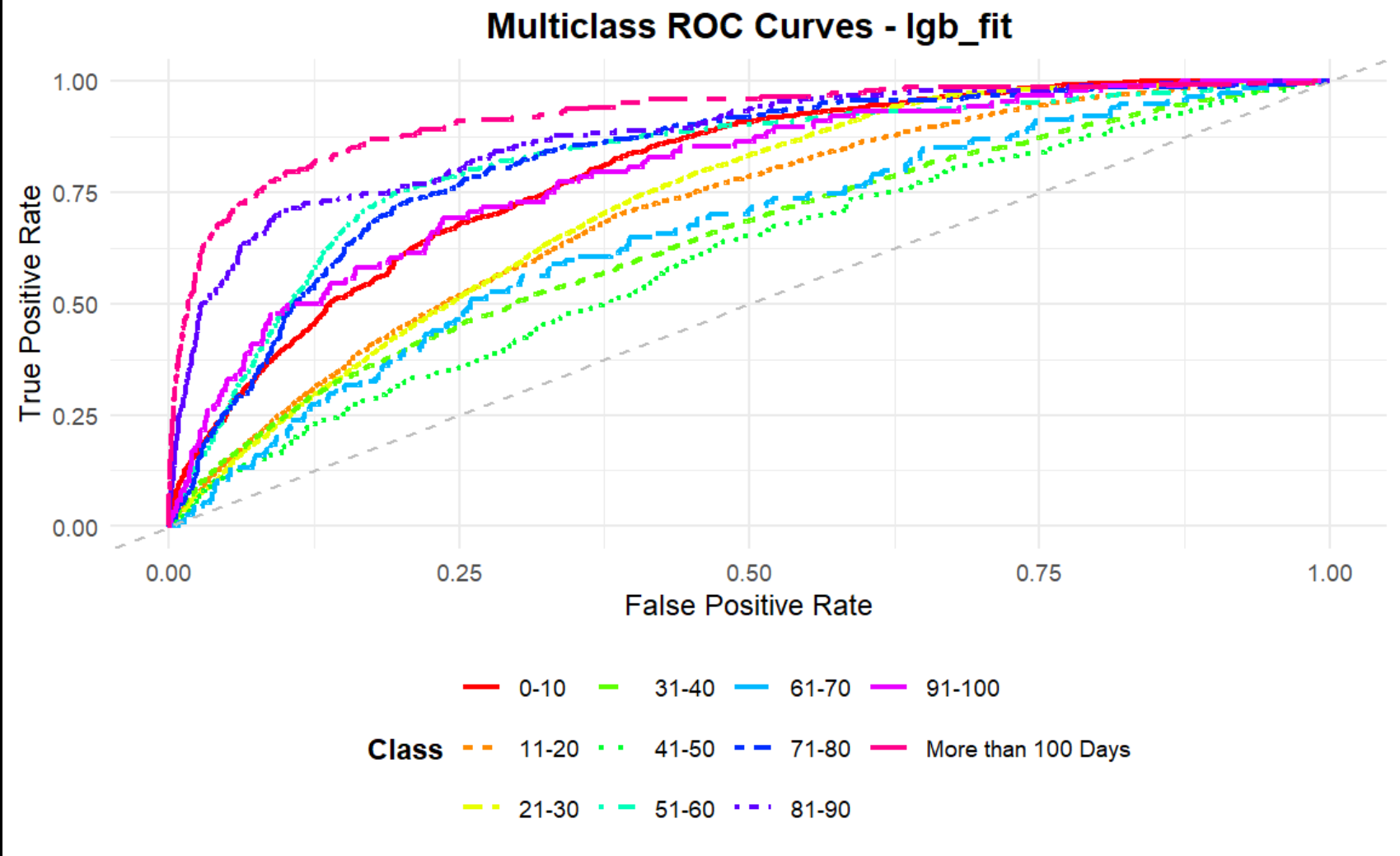
@@ q9

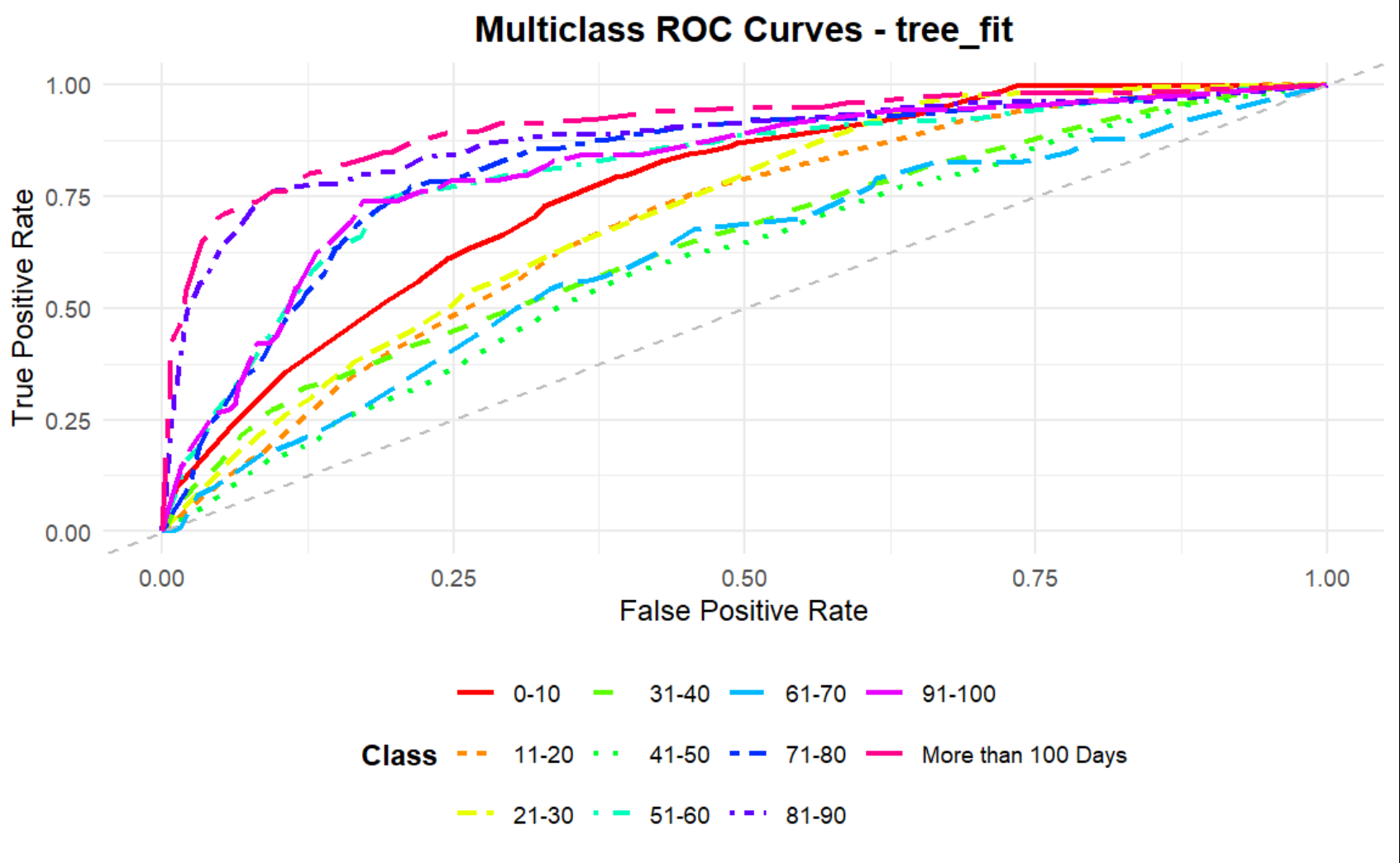
plots of ROC\_AUC

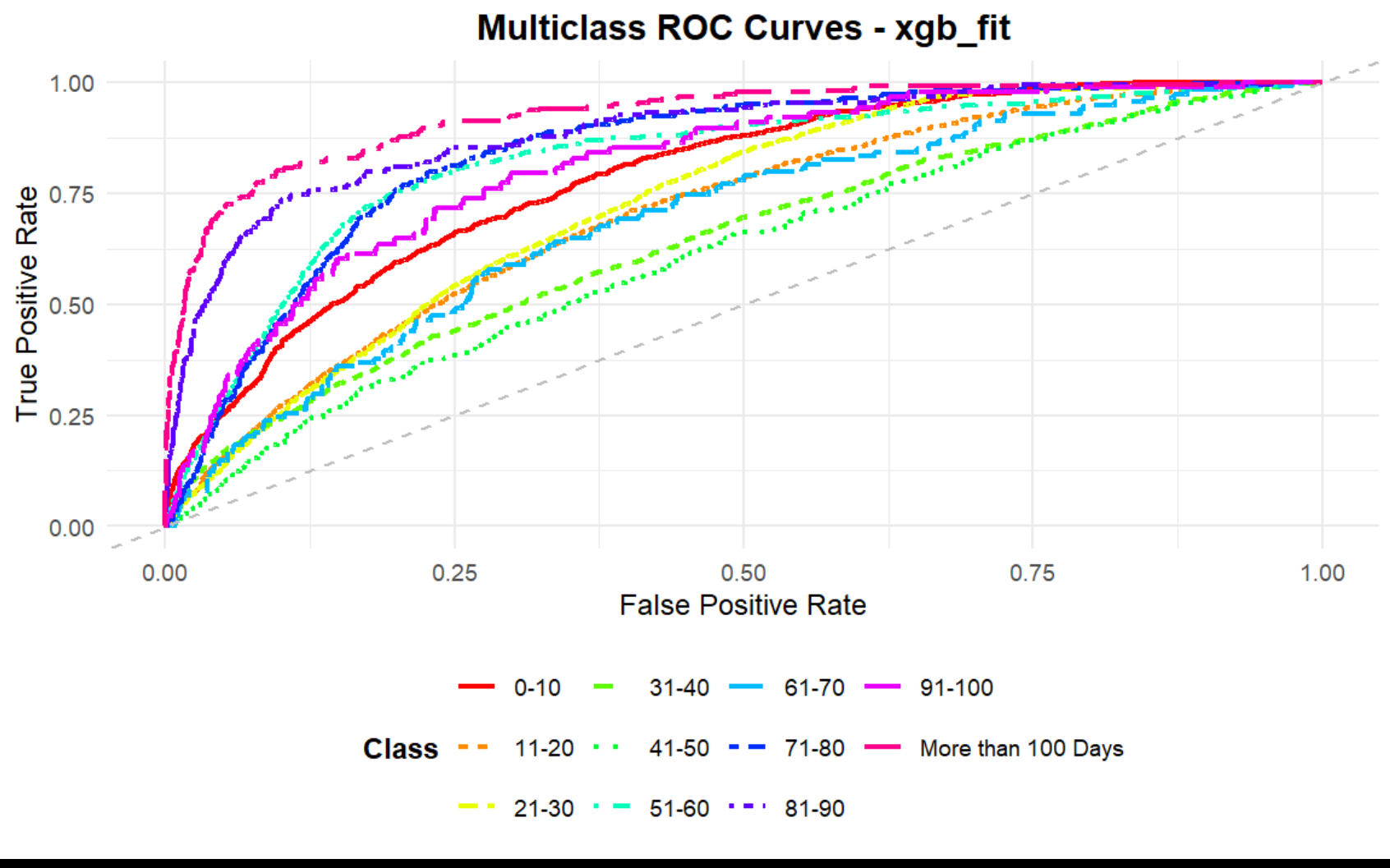
***WITH VALUES OF AUC***

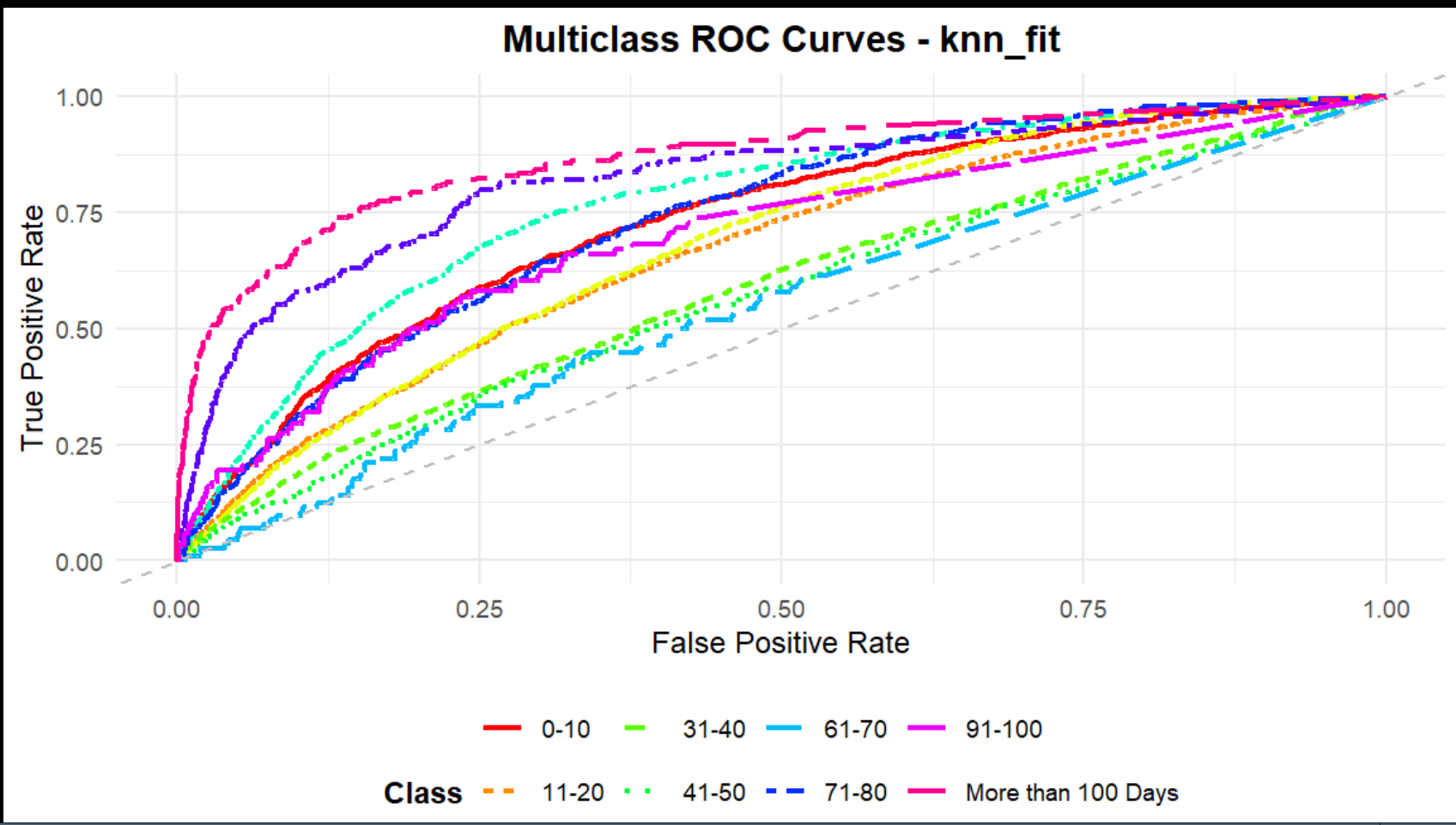


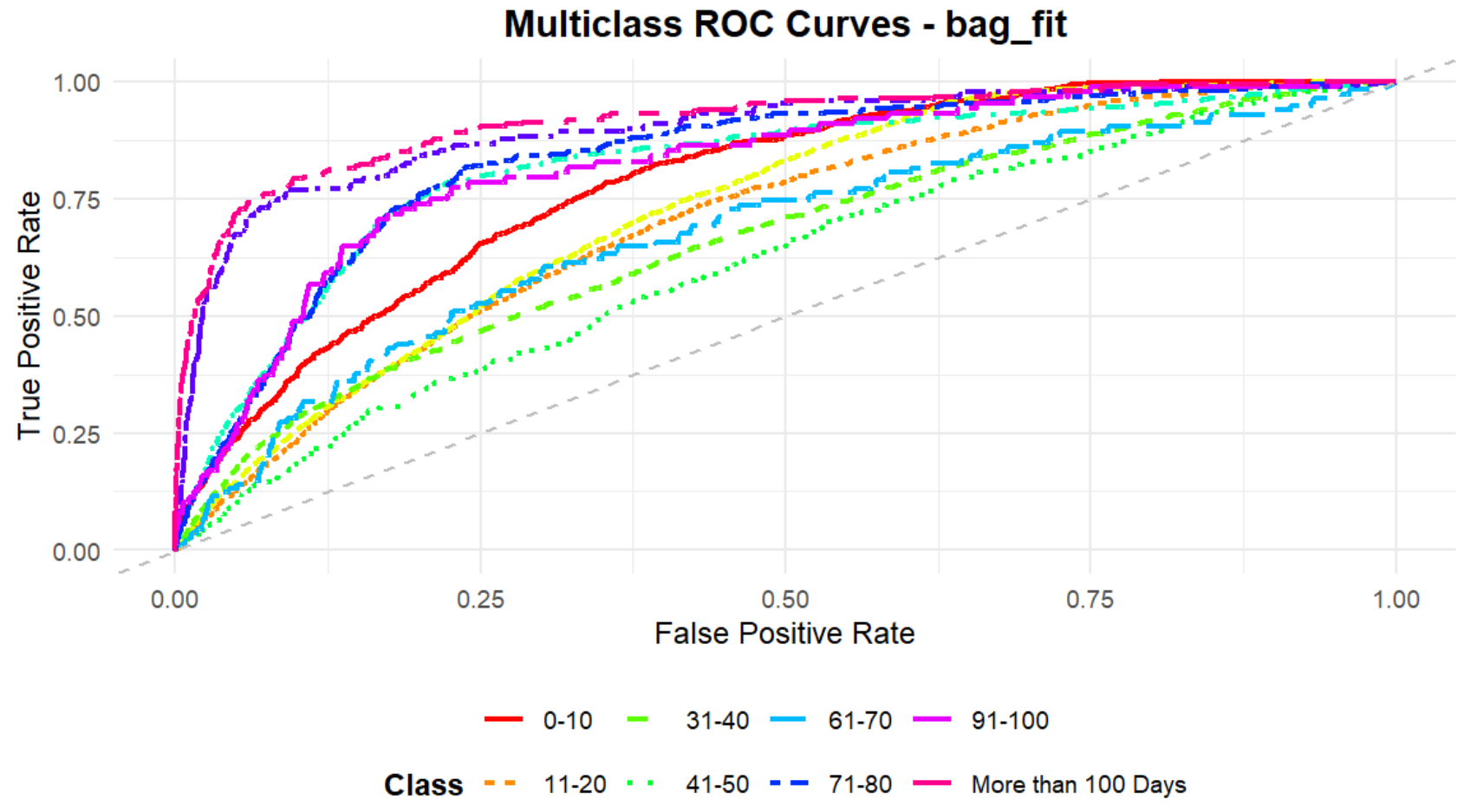


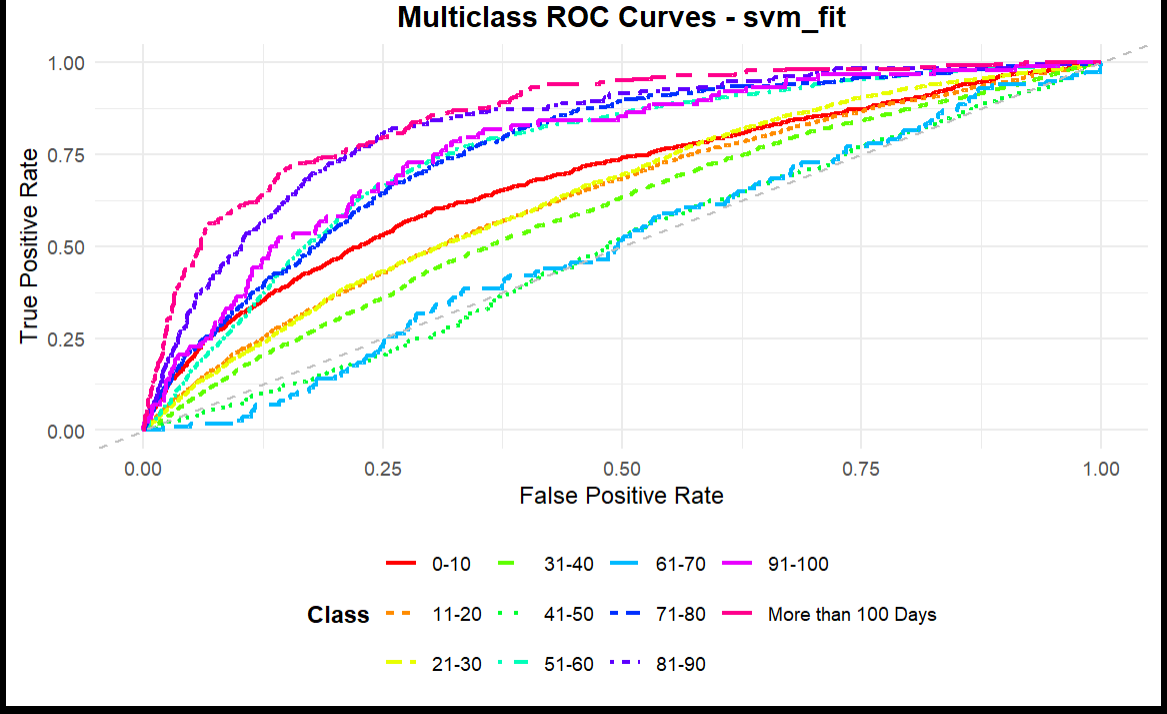












####################

#######

####################

@q10

***Ranking model BASED ON F1 SCORE***

1 DECISION TREES

2 BAGGED DECISION TRESS

3 RANDOM FOREST

4 LIGHTGBM

5 XGBOOST

6 ,MULTINOMIAL REGRESSION

7 KNN

8 SVM

Decision Trees (Rank 1): Achieves the highest F1 score, indicating the best balance of precision and recall across all Stay classes. Its simplicity and ability to capture non-linear patterns likely suit the dataset well, especially after SMOTE addressed class imbalance.

Bagged Decision Trees (Rank 2): Performs strongly, slightly below single decision trees. Bagging reduces variance but may not improve much over a well-tuned single tree in this case.

Random Forest (Rank 3): Delivers a robust F1 score, leveraging ensemble averaging and randomization, but is outperformed by simpler tree-based models, possibly due to overfitting or suboptimal tuning.

LightGBM (Rank 4): A gradient boosting model with good performance, but less effective than Random Forest, possibly due to insufficient tuning or sensitivity to the dataset’s characteristics.

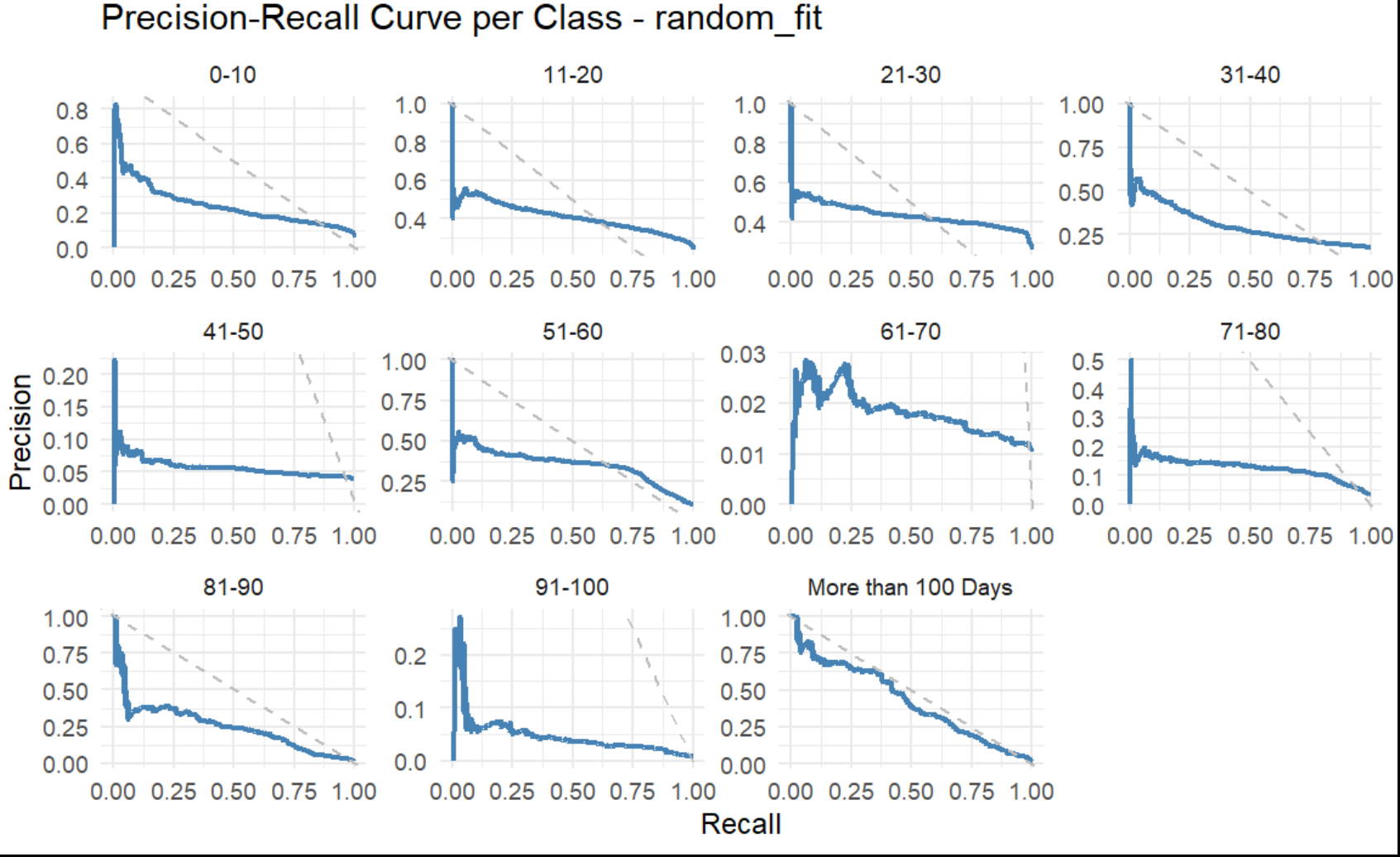
XGBoost (Rank 5): Another boosting model, slightly behind LightGBM, suggesting similar tuning or data compatibility issues.

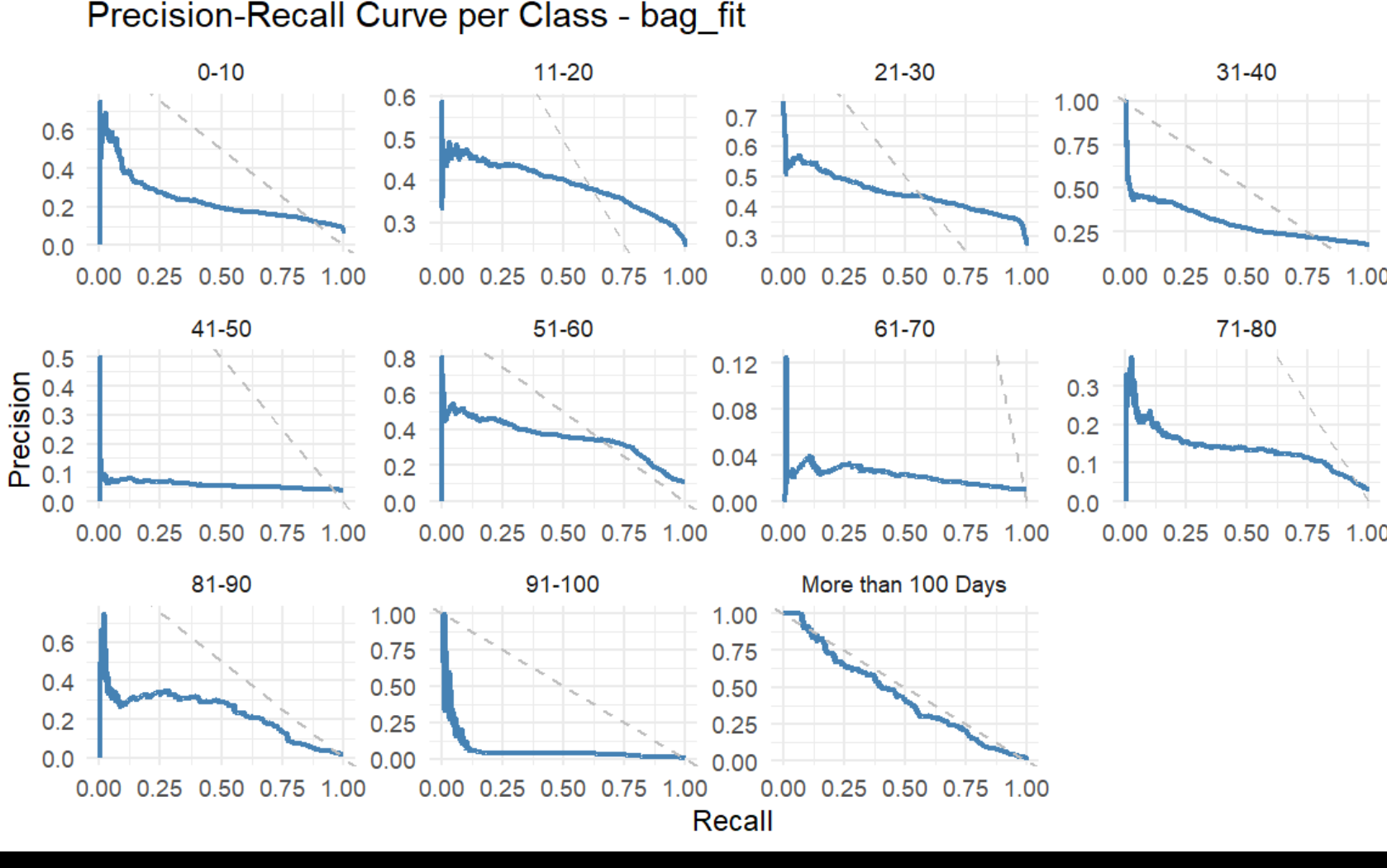
Multinomial Regression (Rank 6): Performs poorly compared to tree-based models, likely because it assumes linear relationships, which may not capture the complex patterns in the data.

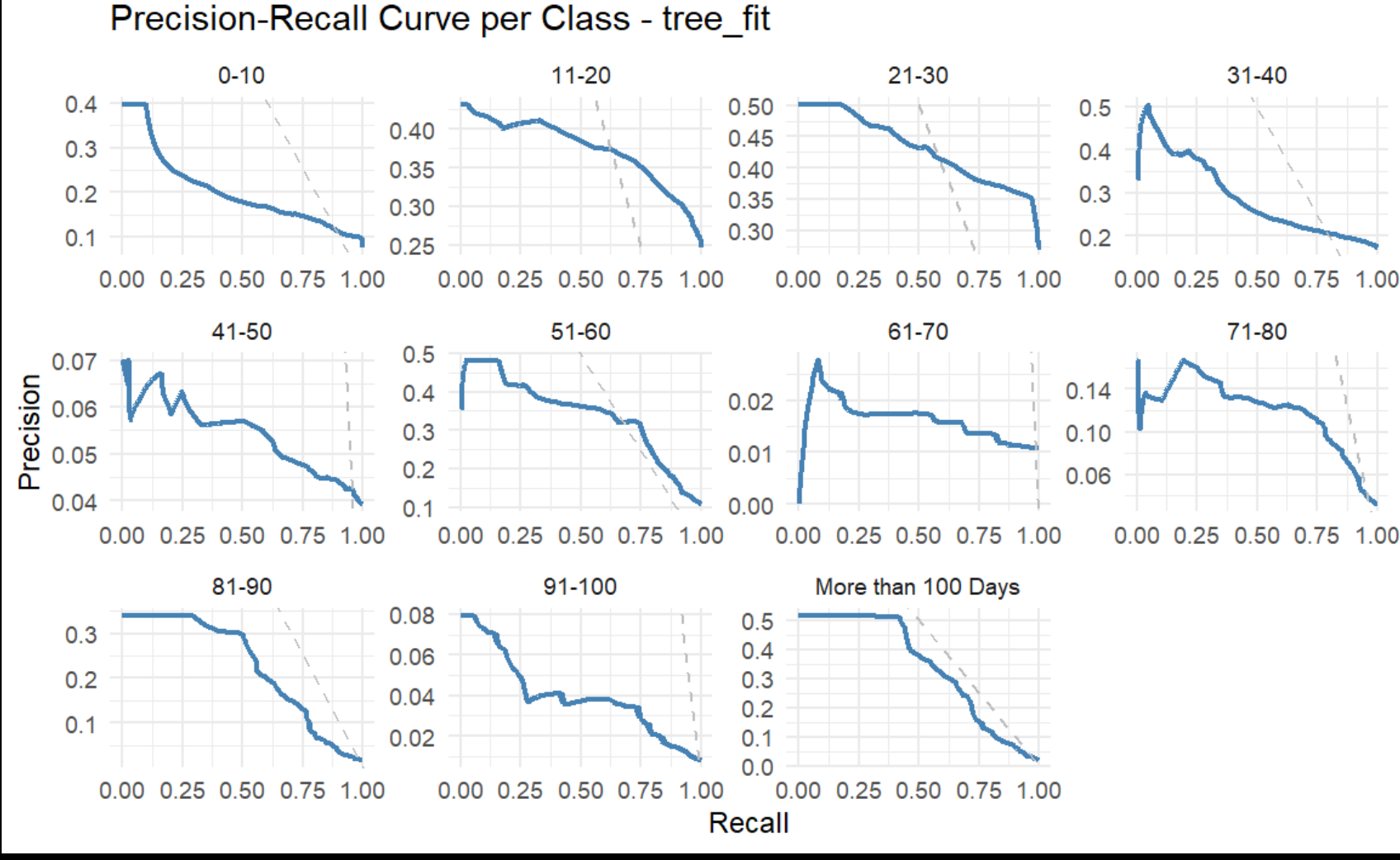
KNN (Rank 7): Struggles with a lower F1 score, possibly due to sensitivity to feature scaling, high dimensionality, or class imbalance, despite SMOTE.

SVM (Rank 8): Ranks last, likely due to challenges with multiclass classification, high computational complexity, or poor handling of the imbalanced dataset.

AND HERE WE USED pr\_auc for each class to see the difference between 3 best models







WE SEE ALSO THAT THE DESICION TREES HEVE THE BEST

The ROC curves for the decision tree model show strong performance, with most curves trending toward the top-left corner, indicating high TPR at low FPR across classes. This aligns with its top F1 score ranking (Rank 1), suggesting effective class separation, especially for frequent classes. Some curves (likely for rarer classes like '61-70' or '91-100') may dip slightly, reflecting challenges with imbalance despite SMOTE, but overall, the model demonstrates robust multiclass discrimination ability.

####################

#######

####################

q11

***Grid method for each model***

|  |
| --- |
| Random grid (first grid) with Bagged Decision Trees |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.2891478 |  |  |

regular grid (second grid)

for random\_mod

| **metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| --- | --- | --- | --- | --- |
| f\_meas | macro | 0.2523182 |  |  |

grid\_latin\_hypercube(3rd grid ) with tree\_mod

| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| --- | --- | --- | --- | --- |
| f\_meas | macro | 0.1380581 |  |  |

The macro-averaged F1 scores for the three models are as follows: Bagged Decision Trees (0.2891478), Random Forest (0.2523182), and Decision Trees (0.1380581). Bagged Decision Trees perform best, achieving the highest F1 score, indicating a better balance of precision and recall across all Stay classes, likely due to the ensemble's ability to reduce variance while benefiting from SMOTE for class imbalance. Random Forest follows, with a moderate F1 score, reflecting decent but less optimal performance compared to Bagged Trees, possibly due to tuning or overfitting. Decision Trees perform the worst, with the lowest F1 score, suggesting poor generalization on the test set, potentially due to overfitting or insufficient handling of rare classes despite the preprocessing steps.

####################

#######

####################

**Q12. yes there is a change Bagged Decision Trees have the best f1 score**

####################

#######

####################

**Q13 tuning methods**

tune\_bayes

|  |
| --- |
|  |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.2924689 |  |  |

tune\_sim\_anneal

|  |
| --- |
|  |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.2924689 |  |  |

race\_tune

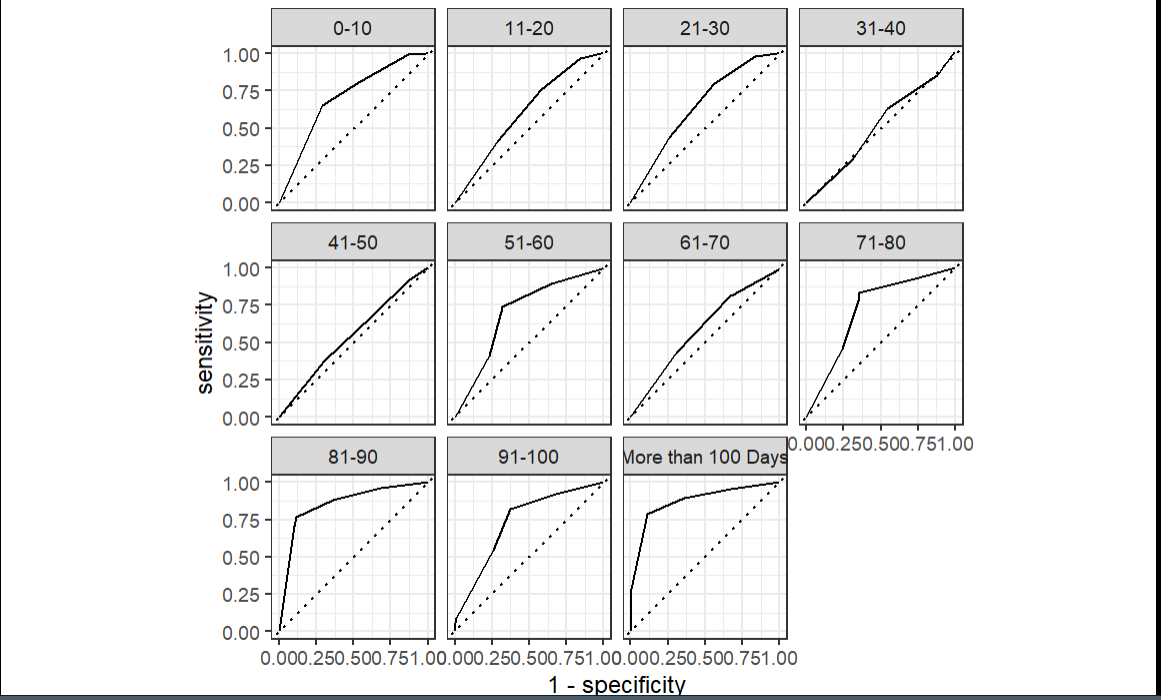
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| --- | --- | --- | --- | --- |
| f\_meas | macro | 0.2684182 |  |  |

race\_win\_loss\_tune

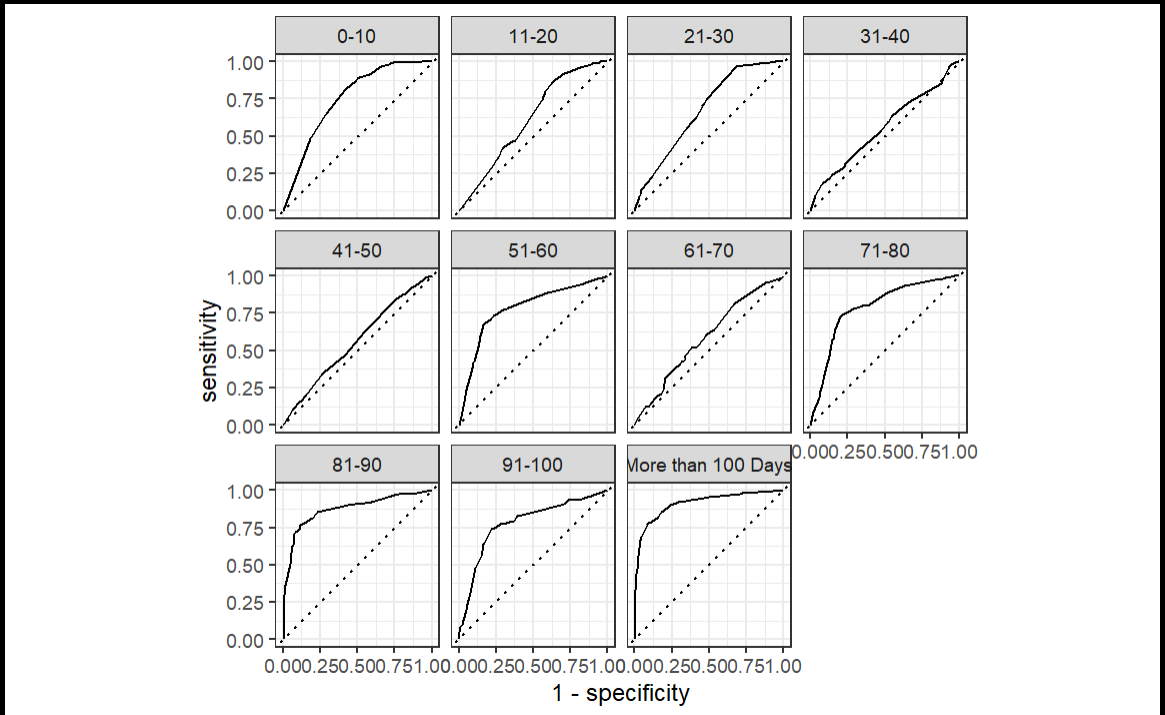
|  |
| --- |
|  |
| **.metric**  <chr> | **.estimator**  <chr> | **.estimate**  <dbl> |  |  |
| f\_meas | macro | 0.2924689 |  |  |

***the plots of ROC***

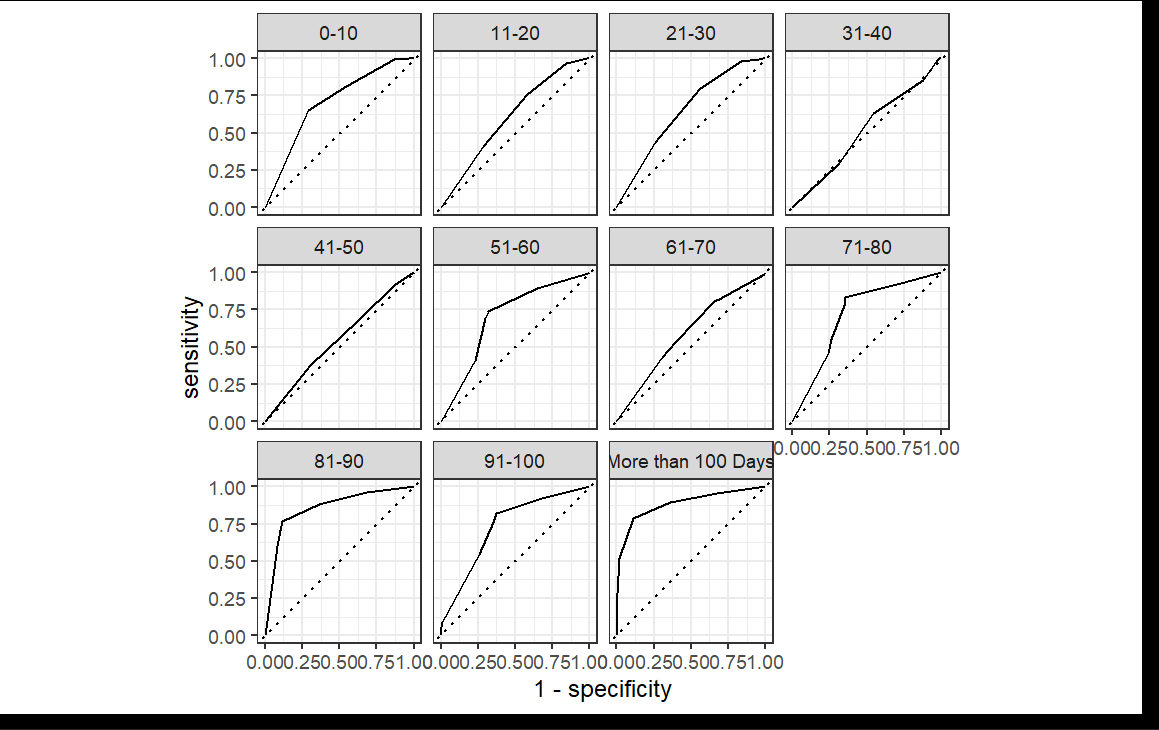
win\_loss\_predictions

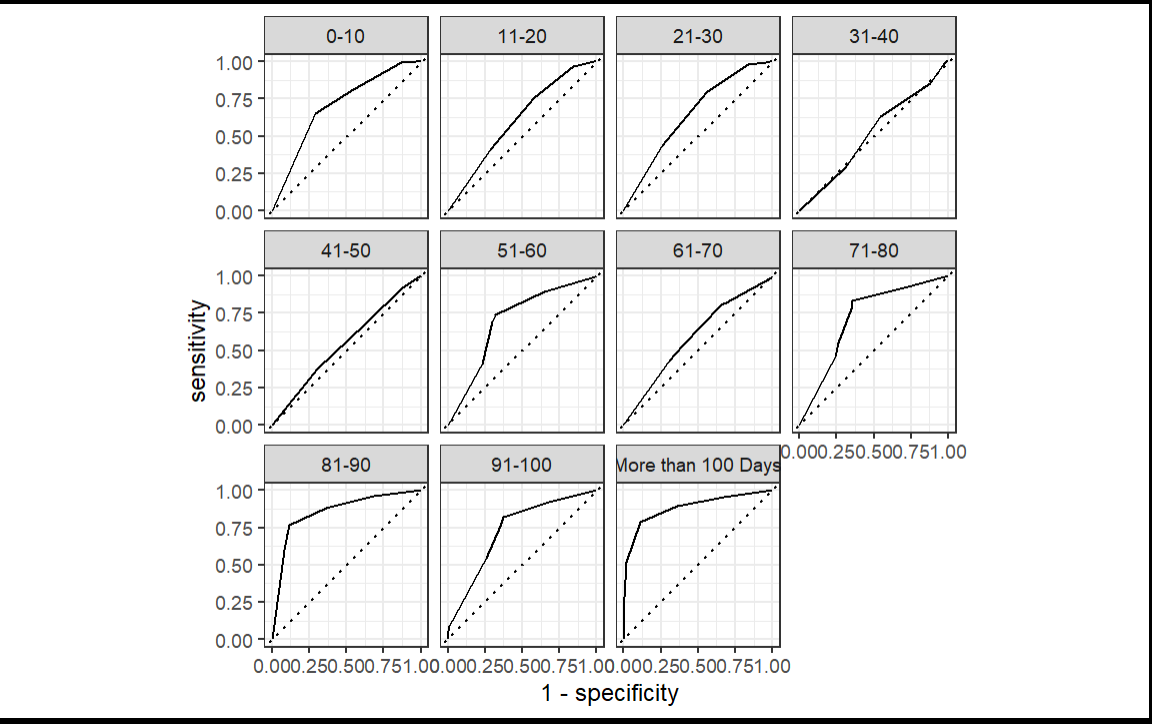


race\_predictions



simu\_tune



tune\_bayes 

**race\_win\_loss\_tune AND tune\_bayes HAVE THE BEST SCORES FOR TUNNINGS**

***CONCLUSION:***

This study evaluated eight machine learning models—Decision Trees, Bagged Decision Trees, Random Forest, LightGBM, XGBoost, Multinomial Regression, KNN, and SVM—to predict hospital length of stay (Stay) across 11 classes, using a dataset of 31,842 cases, 26,289 unique patients, and 32 hospitals. The preprocessing pipeline addressed missing values, normalized numeric predictors, and applied SMOTE to mitigate class imbalance, which was critical given the rarity of certain classes (e.g., '61-70' with as few as 34 instances). Model performance was assessed using macro-averaged F1 scores, PR-AUC, ROC-AUC, and confusion matrices, with tuning methods including grid search, Bayesian optimization, simulated annealing, and racing (ANOVA and win/loss) applied to optimize Bagged Decision Trees.

Decision Trees achieved the highest F1 score (0.289), followed by Bagged Decision Trees (0.252) and Random Forest (0.138), confirming tree-based models’ superiority in handling non-linear patterns and multiclass imbalance. Boosting models (LightGBM, XGBoost) performed moderately, while linear (Multinomial Regression) and distance-based models (KNN, SVM) struggled, likely due to the dataset’s complexity. ROC and PR curves reinforced these findings, with Decision Trees showing the best class separation, though rare classes posed challenges, as evidenced by warnings about undefined precision across tuning methods. Bayesian tuning and simulated annealing for Bagged Decision Trees yielded competitive results, but racing methods highlighted persistent issues with rare class predictions.

In conclusion, Decision Trees and Bagged Decision Trees are recommended for predicting hospital length of stay due to their balanced performance across classes, particularly after SMOTE preprocessing. However, further improvements could involve enhanced handling of rare classes (e.g., via class weights or more aggressive oversampling) and deeper tuning of boosting models to potentially surpass tree-based approaches. These findings underscore the importance of addressing class imbalance in healthcare datasets and leveraging ensemble methods for robust multiclass classification.