

Kobe Bryant shot selection

Artur Samigullin & Antony Bykov

Supervisor: Damir Mirzanurov

Agenda

Business Understanding

Data Understanding

Data Preparation

Modeling

Evaluation

Summary

Business Understanding

Description: To find which of the Kobe Bryant's shot was successful

Task: for each missing shot result predict a probability that Kobe made goal

Evaluation metric: log loss

Data Understanding

Name	Type	# of unique values	Comment
opponent	categorical	33	Opposite team name
matchup	categorical	74	Opponent + game type
game_event_id	numerical	620	Unique NBA id for shot in game
game_id	numerical	1559	Unique NBA id for game
lat	numerical	457	Latitude
loc_x	numerical	489	X coordinate on the court
loc_y	numerical	457	Y coordinate on the court
lon	numerical	489	Longitude

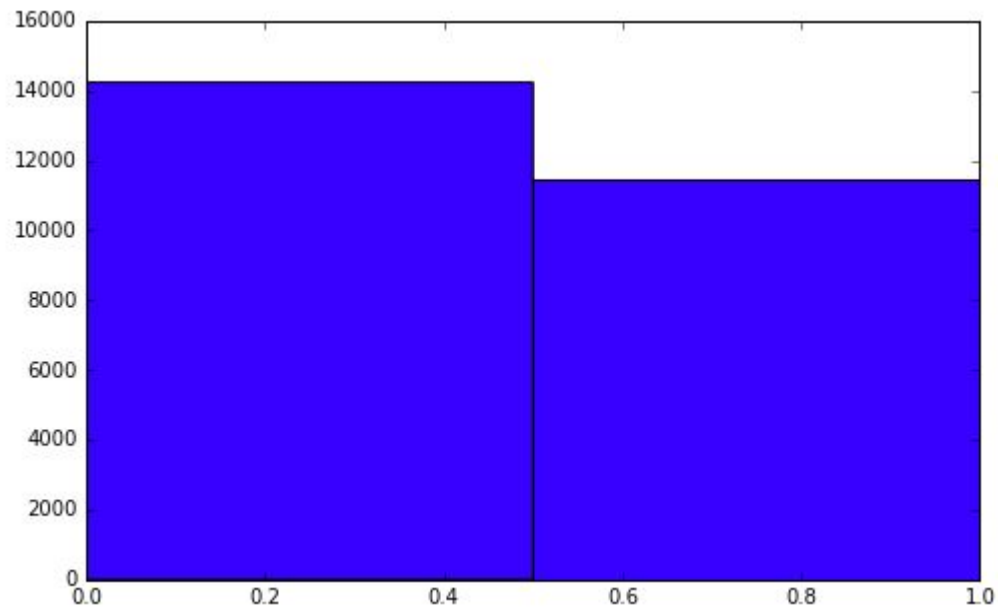
Data Understanding

Name	Type	# of unique values	Comment
action_type	categorical	57	Type of shot
combined_shot_type	categorical	6	Summary type of shot
shot_distance	numerical	74	Distance from basket
shot_made_flag	binary+nan	3	Field to predict
shot_type	binary	2	2 point or 3 point zone
shot_zone_area	categorical	6	Zones of players
shot_zone_basic	categorical	7	Court zones where shot done
shot_zone_range	ordinal	5	Distance ranges

Data Understanding

Name	Type	# of unique values	
minutes_remaining	numerical	12	Each period is 12 minutes (NBA)
seconds_remaining	numerical	60	Each minute is 60 seconds
period	numerical	7	4 main periods, 3 overtimes
season	ordinal	20	Year and ID of season
playoffs	binary	2	Type of match
team_id	numerical	1	Kobe played for 1 team
team_name	categorical	1	Kobe played for LA Lakers
shot_id	numerical	30697	Id for record

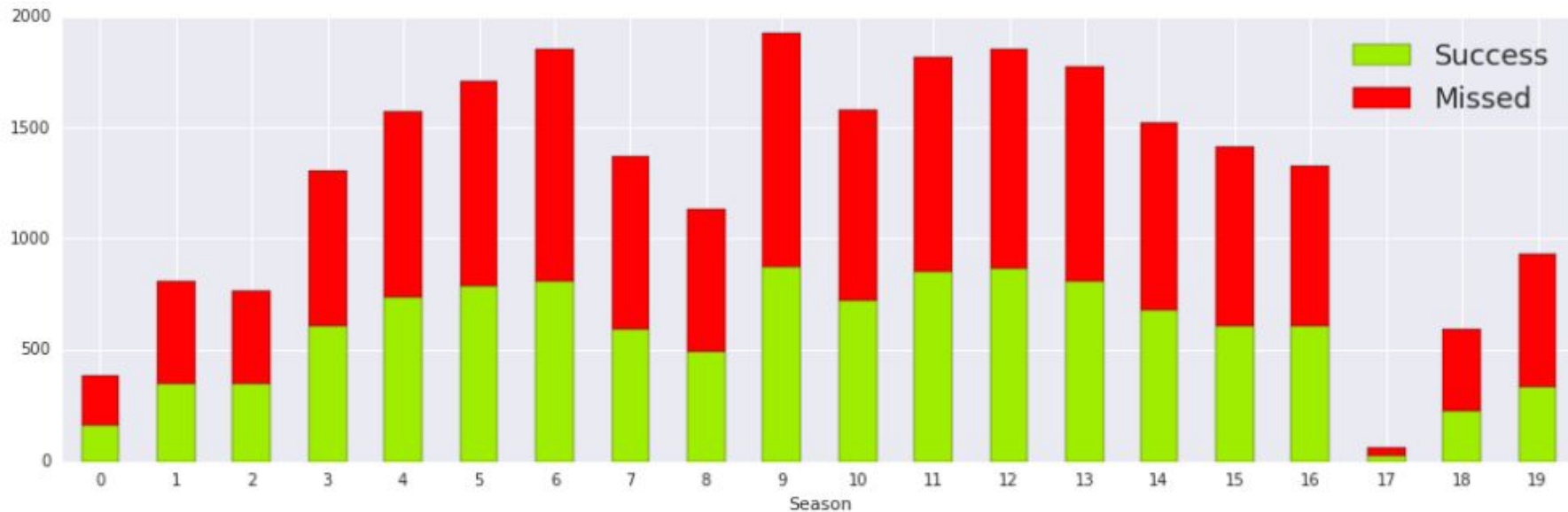
Classes Distribution



From this histogram we see that zero class has most fraction

Actually, it is 55.4% for missing and 44.6% for success shot

Success/Missed ratio by seasons

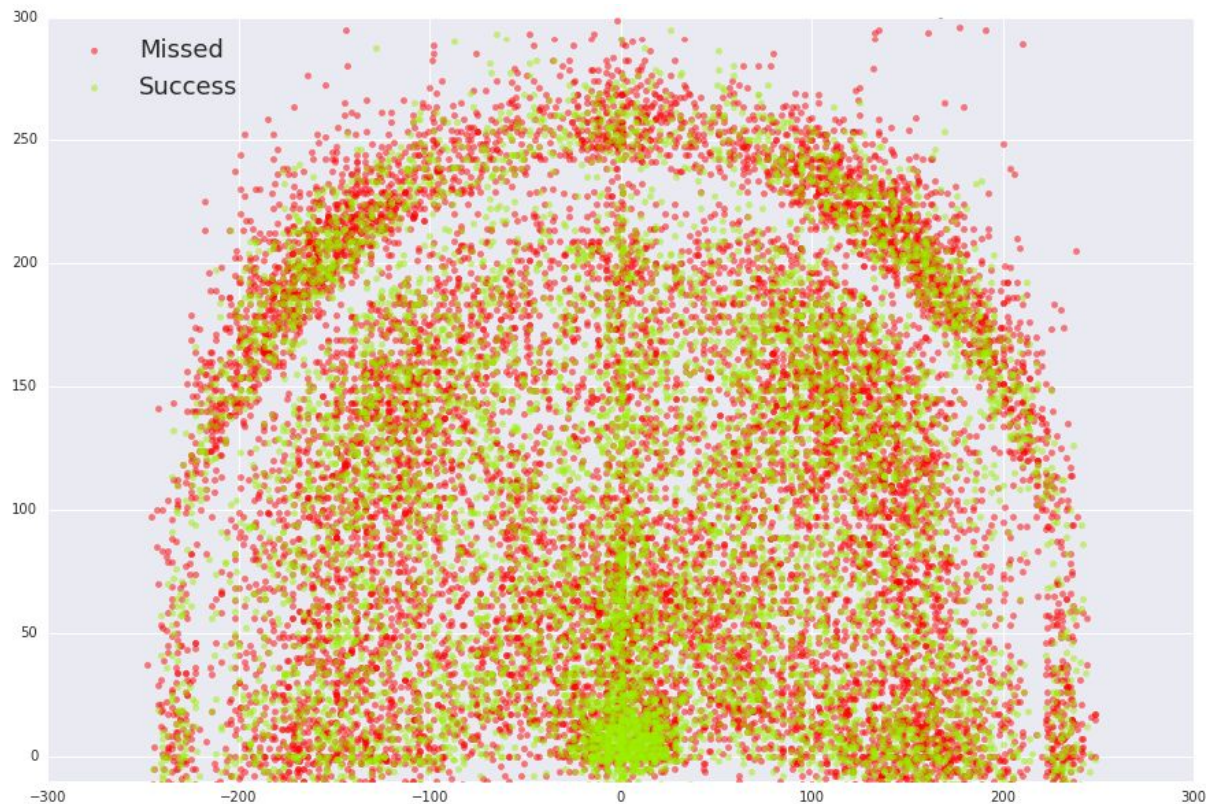


Here we can see that:

Seasons not numbered in right historical order

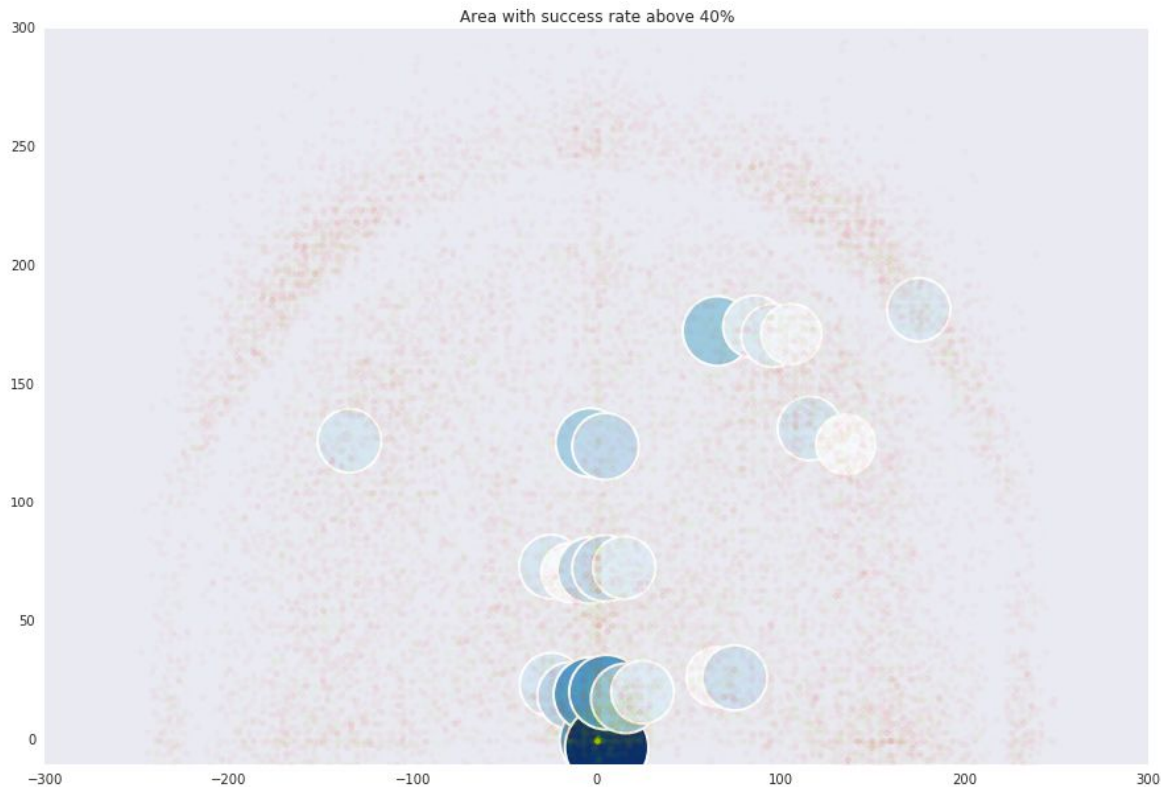
Ratio of success/missed is likely to be the same

Missed/Success by coordinates



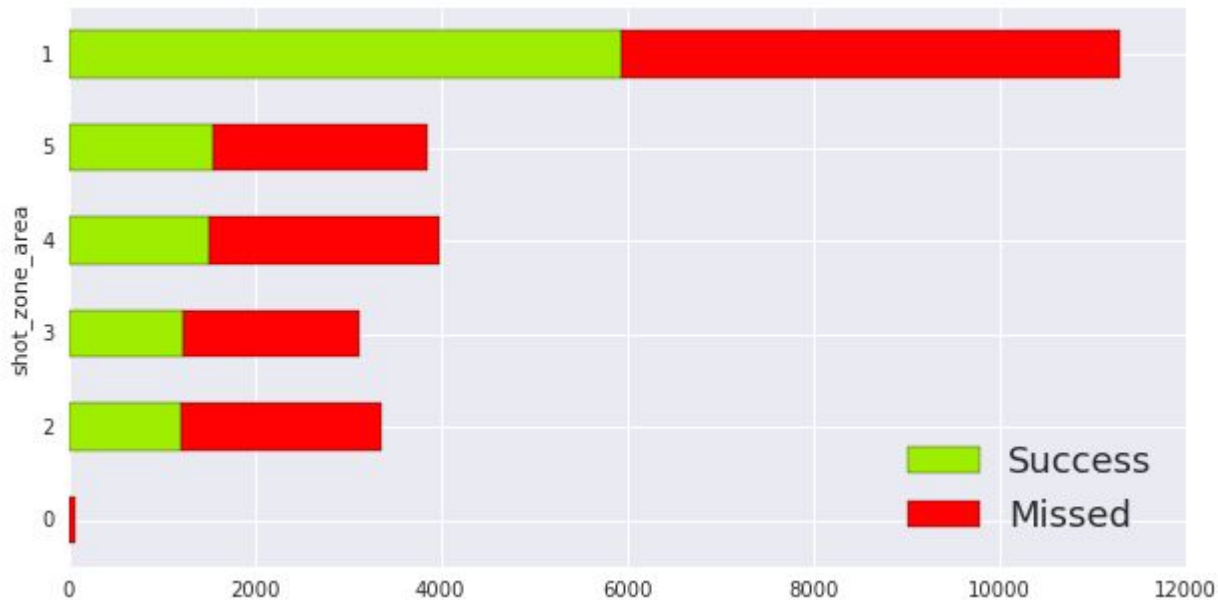
Here we can see concentration of success shots in the zone near to basketball ring

Above 40% success rate areas



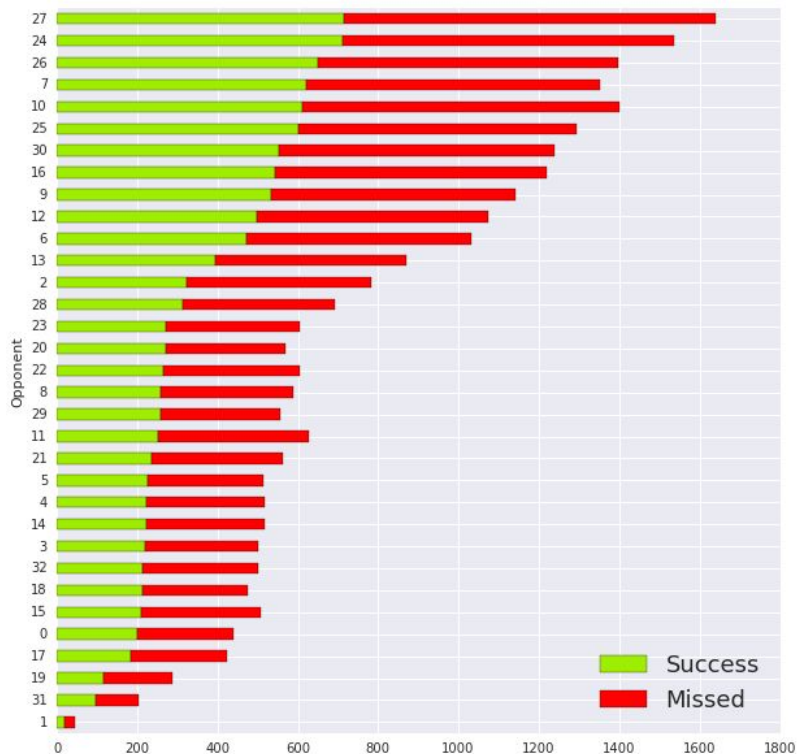
From this graph we can think that distance and coordinate feature give us a lot of information about success probability

Success/missed by shot_zone_area



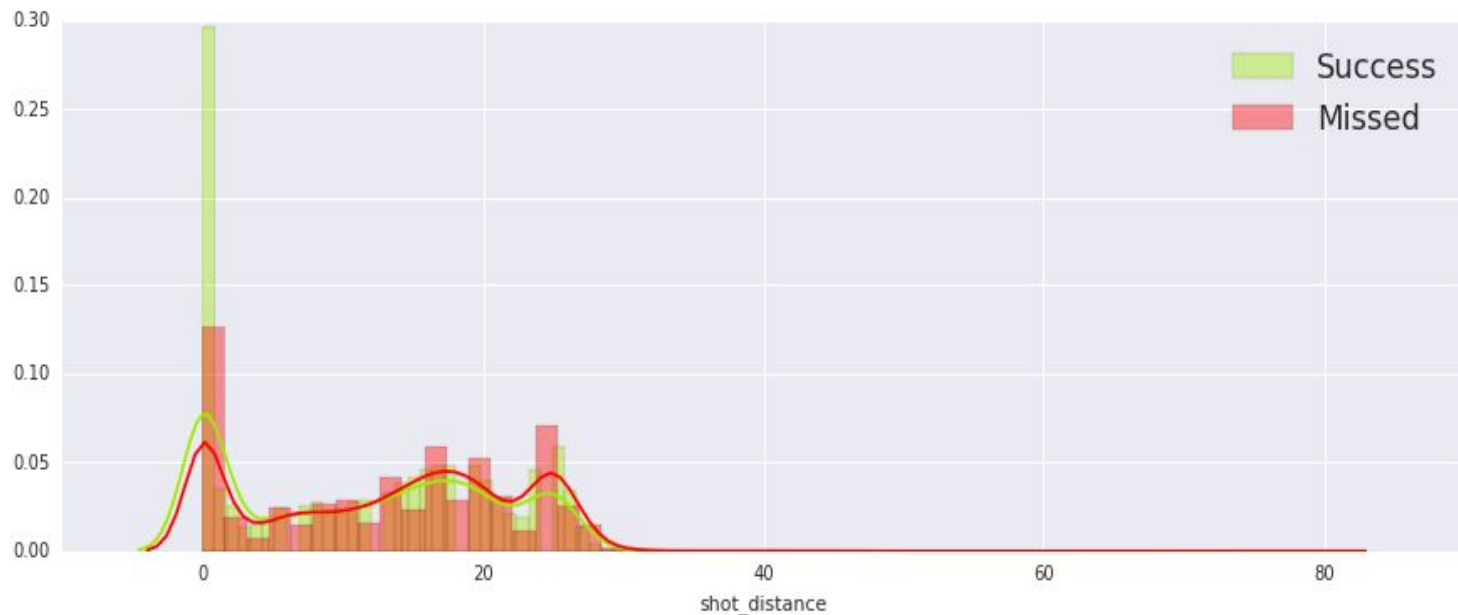
Shots from backcourt zone have a very little chance to be successful

Opponents



NBA League has a very good basketball teams -
Ratio of success shots likely to be the same on different
Playings with different teams

Distance



Distance can be split into two classes - where success probability higher and where missing probability higher

Data Understanding Summary

We have 23 features that describe time, type, position of shot, opponents and etc.

Features look like having **duplicate information** and some of features need to be **combined together**

Distance and **coordinates** looks like very important features

Pair plot of features brought no insights, just obvious dependency between distance and coordinates

Data Preparation

Main steps:

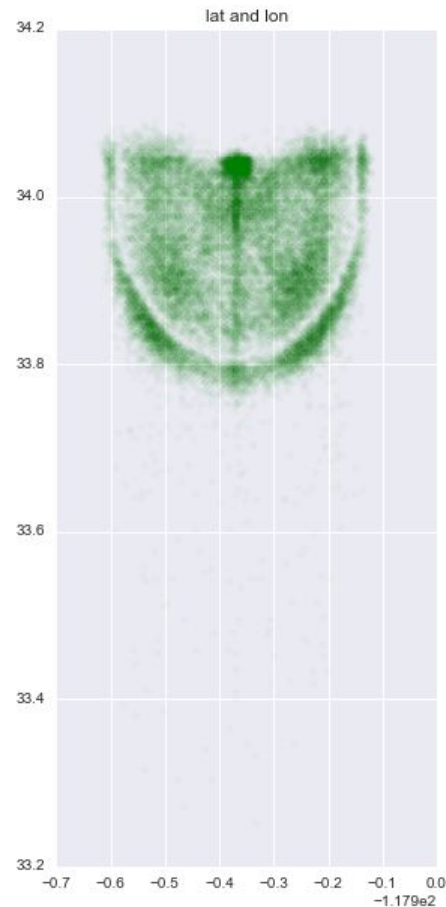
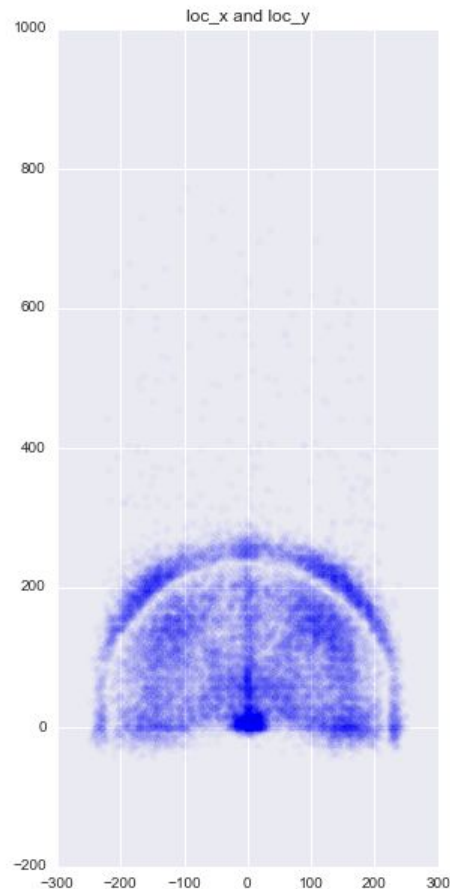
1. Data Selection
2. Hidden meaning
3. Useful Data
4. Encoding, Normalization and Insights

Data Selection

We found that:

team_id and team_name is useless,
since we have constant value on whole
dataset

pairs (loc_x, loc_y) and (lon, lat) shows
same information



Data Preparation. Hidden Meaning

We extracted and encoded home/guest from matchup and made new feature `is_home`, that shows was it home match or not

Now matchup can be decomposed with **`is_home`** and **`opponent`** features

Data Preparation. Useful Data

We used the rows in which there is the count of unique features is less than some number (supposing that they are discrete), we've tried to play with this number and stopped on the value of 80.

But that wasn't enough because there are some features which are not discrete and that have to be included: **game_date**, **loc_x**, **loc_y**.

Encoding and Insights

We tried out two encoding of categorical data - label-encoding and one-hot-encoding. One-hot-encoding worked better

Minutes and seconds remaining is one feature - time remaining in seconds

Seasons is ordinal data, but better to encode them in historical order

Modeling

We tried plenty of classifiers:

KNN

RandomForest

LogisticRegression

XGBoost

Modeling. KNN

KNN is distance based method and we have a coordinates and distance features

But without optimization it worked awful and was unpredictable - on out log loss scoring it got score about 9, and in score about 3

(It is arised from number of predictions)

Modeling. Random Forest

Random Forest showed not very good result out of the box, but optimized with Grid Search Cross Validation with plenty of parameters brought a nice result

Parameters for GridSearchCV:

```
{'n_estimators':[100,200,300,500,800,1300],  
'max_features':[1,2,3,"auto","log2",None],  
'min_samples_leaf':[1,2,3,5,8],  
'max_depth': [None, 10, 20, 30, 50],  
'min_samples_split':[1,2,3]}
```

And best set is:

```
{'max_depth': 10,  
'max_features': 'auto',  
'min_samples_leaf': 2,  
'min_samples_split': 2,  
'n_estimators': 200}
```

Modeling. Logistic Regression

Logistic regression showed a good results out of the box

Result improved by 0.1 on a dataset, where we dropped all features whose importance was less than 0.1

Modeling. AdaBoost

The more estimators AdaBoost have, the more careful predictions it does.

We tried a 1000 - 20000 estimators and playing just with estimators limited with log-loss of all 0.5 submissions

Since that, we tried to play with features. Looking to feature importances we dropped features with importance less than 0.1.

It gave improving for result, but very small.

Modeling. XGBoost - Best Model

The first model worked on standard loss-function, on different amount of features - starting from the tuple (loc_x,loc_y) through tuple of all the features we had in the data.

First big step to optimise the way XGBoost works - was adding logloss loss function.

The next step was adding loc_x and loc_y features manually, all other features were filtered by the amount of their amount of unique values.

The last step was using one-hot-encoding for all the category-features.


Evaluation

Algorithm	FeatureSet	Train	Score
XGBoost	Discreteness 80	log_loss	0.60669
AdaBoost	Feature_importances ≥ 0.1	grid_search	0.69325
LogisticRegression	Cleaned ds	-	0.7
Random Forest	Cleaned ds	GridSearchCV + log_loss	0.89272
Random Forest	Cleaned ds	GridSearch	0.94057
Gradient Boosting	Feature_importances ≥ 0.1	-	1.02574

Summary

The best Kaggle evaluation score is 0.607 and place is 78 out of 382

The best model was the XGBoost with number of discreteness = [10..1000] and trained on log-loss metric

78	↑149	innodan 	0.60669	9*
----	------	---	---------	----

*Don't get confused about small amount of commits - we had committed from ourselves most of the time and joined the team just at the end

Review from supervisor

In this competition we faced with classical classification task, in which we could play with different algorithms and feature engineering. Anton and Artur step by step went through all phases of CRISP-DM for solving tasks:

- 1) data understanding and visualization;
- 2) initial hypothesis about feature set;
- 3) initial algorithm selecting.

After initial attempt, they made a new sprint.

Each of the guys showed good knowledge about classification task, data visualization, and model testing.

Each of them plotted graphs, which helped them understand data.

Each of them tried different algorithms (RandomForest, XGBoost, AdaBoost), and played with parameters and evaluation function.

The main step was working with features.

Good visualization showed that Distance and field coordinate is very important features. Another problem was how to encode categorical features - one-hot encoding showed better results with XGBoost

For this work we gained a 78th position across 382 teams on leaderboard.

This is good result, and it's show that guys can apply their knowledge to real competition and they know main steps of solving practical Machine Learning task.

Guys did great job by going through whole process individually and gained a lot from my supervision.