Group C: Tina Chen and Xiaoli Lin
Competition: Sberbank Russian Housing Market Prediction
Link to project: https://www.kaggle.com/c/sberbank-russian-housing-market
----

Aim of Competition:

To predict the sale price of each property. The target variable is called price_doc in train.csv. The training data is from August 2011 to June 2015, and the test set is from July 2015 to May 2016. The dataset also includes information about overall conditions in Russia's economy and finance sector, so that we can focus on generating accurate price forecasts for individual properties, without needing to second-guess what the business cycle will do.

Timeline:

5/5/2017: Our approach was to start on the project early, so we met up on the first day the project was assigned and each chose three competitions from Kaggle.
We looked at all the competitions to see which one was the most interesting and fitting for us. At the end, we chose Sberbank Russian Housing Market, because we wanted to understand housing costs from the perspectives of both consumers and developers. Additionally, Xiaoli is studying Russian at Middlebury and Tina's family is from Vladivostok.

5/7/2017-5/9/2017: We started planning out what to do and when to meet up in the next coming days. We were able to understand our data (though we did do a lot of outside research to understand some of the predictor variables—for e.g., we used Google Maps to view the districts). We also narrowed down the 393 variables to 75, but we thought that was still too many.

5/10/2017 – 5/13/2017: We spent a large chunk of our time trying to understand the data, so performed exploratory data analysis on the different predictors. We were also able to measure the population density of the districts, which can be found in Extra as well.
From manipulating data to visualizing the data, we found this step to be extremely valuable in helping us visualize what are the important predictor variables.
We also decided what models to build. Since we had both continuous and categorical variables, we believe that Ridge and LASSO are suitable methods.

5/14/2017 – 5/19/2017: After we analyzed our data, we also tried to make sense of our scoring mechanism, RMSLE. We started by using the built-in cross-validation (CV) function and built-in RMSLE function from the Metrics Package, because we wanted to make a submission just to see how we would do. However, our computed score was completely different than our Kaggle score (it estimated it to be ~1.30). We presume that maybe the built-in RMSLE calculates the scores differently.We went on to write our own CV for the models and later found a way to calculate our RMSLE score. Our resulting scores were between 0.31-0.38, which is in the same ballpark as our Kaggle scores.

5/21/2017: We finished up editing our final submission, and revised some of our EDA's, which can be found at the bottom of the Extra Section. Additionally, we were thinking of running this Naïve XGBoost script that everyone was talking about in the Discussions Board that would have given us an automatic ~0.31. However, we decided not to because it seemed wrong, even if we were just curious.
We also decided to deal with the three negative values in our predictions. Initially, we had changed the values to 0, but would houses be sold at $0? We thought about making the values

into absolutes and just taking the mean. In the end, we explored the areas the houses were in and decided to take the mean price of the houses in the said area.

Methods:

Data: We were given five data files: train, test, data dictionary, macro, and sample submission. The training dataset had 30,471 observations, with 292 variables pertaining to the individual transactions of sale prices of each property between August 2011 to June 2015. The macro data (~100 predictor variables) gives information about Russia's macro-economy and financial sector, and we were able to join it with train and test. The data dictionary was unhelpful; a majority of the variable descriptors were missing. For example, we weren't sure what 'brent' or 'rent_price_4.room_bus' meant. Some data were unclear and needed a lot of cleaning. For example, for 'floor' and 'max floor', there were times when 'floor' >= 'max floor.' We also had many questions about the data: are floors only pertaining to apartments or are they for houses as well? What's the difference between 'raion' and 'sub_area'? Understanding the data to accurately predict prices for properties was challenging, but at the same time, very eye-opening because we were able to appreciate Russia's culture and economy in a whole new perspective.

Choosing Variables: Our dataset had both continuous and categorical variables. We first narrowed it down to 75 from the 392 total variables and performed EDA on the different predictors to see which may be the most useful for our model. At the end, we were able to narrow it down to ~25.

EDA: We first performed data analysis on all the predictors to figure out which variables had more than 50% of its data missing. From there, we chose 50 that we thought may be the most important when it comes to buying/ selling houses in the Russian market (full disclosure: we both have never purchased a house before so what we ended up choosing were mainly based off of intuition and EDA). Then from the 50, we narrowed it down to 25 by performing LASSO to see which were converging to zero, so it would help alleviate multicollinearity amongst regression predictor variables. We also performed additional outside research to see what the districts in Moscow looked like (link: https://www.kaggle.com/jtremoureux/map-visualizations-with-external-shapefile) and what types of houses are sold in the Russian market (link: http://russian-federation.realigro.com/for-sale/property/).

Root Mean Squared Logarithmic Error (RMSLE): Our submissions were evaluated based on RMSLE, and this scoring mechanism is completely different from what we've done or seen in the past. However, we finally made sense of this mechanism. From our understanding, one of the reasons to score with RMSLE is because you don't want to penalize huge differences in the predicted and true values when both predicted and true values are huge values (house prices are pretty huge). We also found this .pdf to be quite helpful to see RMSLE in action for Kaggle: (http://cs229.stanford.edu/proj2013/Lee-311%20Predictions.pdf).

Models and Cross-Validation: LASSO, Ridge, and CART (Classification and Regression Treee) are the three models we tried. Our final model is LASSO, and both Ridge and CART can be found in the Extra Section Part A and Part B.

- Best score: 0.37 (LASSO submission).
- Mean score: ~0.39 (All three models).
- Lowest score: 0.43 (from our first submission, where we did not calculate RMSLE, but instead, calculated MSE. We also used the built-in CV function).

LASSO model: We initially chose LASSO because it worked for both continuous and categorical variables. We are also thankful because LASSO is able to penalize the absolute size of the regression coefficients. We used CV to identify the best lambda value (the value was around ~1x10^5) and the CV scores are under the cv_results dataframe. We then extracted the optimal lambda value from cv_results and applied the model on test data and made predictions using the best lambda value. For Ridge, we just changed the Alpha to 0.

CART model: We tried CART since Regression Tree allows us to predict continuous values based on both categorical and continuous variables. We first set the maximum depth of the CART model at 3, making sure the CART model would give us viable results, and then we used cross validation to find the best maximum depth (the knob). It turned out that the best maximum depth is 6, which yields a Kaggle score of approximately ~0.41. Ultimately, we decided not to use CART as our final model, because LASSO model has a higher score. Also, CART model creates so many duplicate prediction results since it categorized our observations only into a limited number of categories.

Cross-Validation: We divided the training data into ten folds. For each lambda value, we computed its RMSLE score by taking the mean of the RMSLE score of each fold (scores can be found in the cv_results data frame). It turned out that our simulated scores are pretty close to our final scores on Kaggle (ranging from 0.31 – 0.38).

Final Thoughts:

Initially, we both had this idealistic vision of wanting to make it to the top of the leaderboard as a team. We wanted to somehow prove to ourselves that we're capable of applying what we have learned in the classroom into the "real world." That's what students are supposed to do, right? Whether this mentality of needing to do well is to impress others or ourselves, we're not too sure, but we completely acknowledge this fact and we are completely humbled by this entire experience.

The truth is, making predictions is really, really hard. Like, REALLY hard. We learned that it's not just about formulating the perfect model or finding the correlations between predictor variables. It's more than that. In a sense, we're trying to make sense of things we don't know and to do so, we're trying to our knowledge to seek for new knowledge and manipulate the knowledge to make forecasts that may or may not be accurate (but then again, how do we ever know anything?).

For us, we don't think there could have been a better way of ending this class. We learned a lot, and this final project (or machine learning in general) has given us access into a whole new world, and we're never going to look at data the same way again. We are thankful for the experience of entering into a Kaggle competition and we both have grown a lot, as computer scientists and as people. We are, therefore, very content with our approach and results (and plus, we worked well as a team!!). If there's one thing statistical learning has taught us, it's to always embrace the unknown, because unknown is more than what meets the eye.